Laporan Praktikum Pemrograman II

Modul III – VISIBILITY, SETTERS dan GETTERS, STATIC KEYWORD, DAN CONSTANT

Rizki Saputra/22103001007

Dosen : Achmad Arif Munaji, ST., M.Kom

Tanggal praktikum: senin, 5 April 2023

[ikytopup21@gamil.com](mailto:ikytopup21@gamil.com)

Teknik Komputer

Institut Teknologi dan Sains Nahdlatul Ulama Kalimantan

***Abstrak***—Pemrograman OOP atau pemrograman orientasi object adalah jenis pemrograman yang menggunakan object dan class. Konsep OOP saat ini semakin berkembang. Hampir semua programmer dan pengembang perangkat lunak menggunakan konsep OOP. OOP bukan hanya cara menulis sintaks dari berbagai program berbeda, tetapi selain itu, OOP adalah cara untuk melihat masalah sistem dan pemrograman. Dalam OOP, setiap bagian dari program adalah object. Object mewakili bagian dari program yang dapat dieksekusi(1).

***Kata kunci : Bahasa pemrograman PHP, OOP***

1. Pendahuluan

Bahasa pemrograman memainkan peran penting dalam membuat aplikasi komputer, tanpanya bahasa pemrograman tidak dapat disebut sebagai aplikasi komputer yang diinginkan. Dari semua bahasa pemrograman komputer yang ada, bahasa pemrograman berorientasi objek merupakan bahasa yang paling populer untuk membuat program komputer. Sebuah bahasa pemrograman yang sangat fleksibel dan mudah diubah menjadi tampilan berbasis Windows yang menarik. Bahasa pemrograman berorientasi objek atau object-oriented programming (oop) adalah jenis pemrograman yang menggunakan object dan class. Php awalnya hanya kumpulan skrip sederhana. Berbagai fitur pemrograman berorientasi objek ditambahkan seiring perkembangannya (2).

Pada praktikum ini mahasiswa membuat program atau coding pada visual studio code dengan Bahasa pemrograman php menggunakan object oriented programing (OOP) Visibility, Setters, Getters, static keyword dan Constant. Hasil codingan tersebut dapat di lihat menggunakan local host XAMPP dan dapat di tampilkan pada browser firefox.

1. Tinjauan Pustaka
   1. Visibility

Visibility adalah Teknik yang di gunakan untuk mengatur property maupun method pada sebuah object. Dengan menggunakan access modifier nantinya akan memberi Batasan akses yang di inginkan terhadap suatu object tertentu.

* 1. Setters dan Getters

Object type adalah tipe data yang tidak sekedar menyimpan data tetapi berisikan juga informasi bagaimana untuk mengolah data tersebut. Maksudnya, pada tipe data data lain seperti : integer, float, string hanya berisikan tipe data dia sendiri, sedangkan pada tipe data object dapat berisikan tipe-tipe data lain.

* 1. Static Keyword

Keyword Static adalah suatu teknik memanggil Method maupun property di dalam suatu class. Karena pada dasarnya class hanya sebuah template / blueprint saja dan Static keyword merupakan sebuah kata kunci yang bisa kita gunakan untuk membuat property dan juga membuat method/function didalam sebuah class yang bisa kita akses secara langsung tanpa harus membuat object nya terlebih dahulu atau tanpa harus menginstansiasi class nya terlebih dahulu.

* 1. Constan

constan adalah pengidentifikasi (nama) untuk nilai sederhana. Nilai tidak dapat diubah selama skrip. Nama konstanta yang valid dimulai dengan huruf atau garis bawah (tanpa tanda $ sebelum nama konstanta). Tidak seperti variabel, konstanta secara otomatis bersifat global di seluruh script.

1. METODE PRAKTIKUM

Praktikum ini di mulai dengan memahami modul yang sudah di berikan, mengerjakan free test setelah memahami modul dan menjawab free test lalu membuka aplikasi XAMPP control panel (local host)> di mulai dengan klik tombol start apache hingga indicator pada aplikasi XAMPP berwarna hijau> kemudian masuk pada file XAMPP>htdock dan Buat folder bernama **praktikum02** pada *path* "C:\xampp\htdocs\", lalu open with (Visual Studio Code) > lalu menambahkan file dengan nama prak.php dan mulai membuat codingan sesuai dengan modul yang di bagikan sebagai berikut :

1. Tambahkan class Orange yang merupakan child class dari class Fruit, kemudian tambahkan property $taste dan method itsTaste() yang menampilkan output dari method get\_name() dan get\_color() dengan menerapkan konsep overriding.
2. Buat agar property $taste dan method itsTaste() hanya bisa diakses oleh class Orange saja dan buat agar property dan method pada class parent hanya bisa diakses oleh child dan parent itu sendiri.
3. Buat settters dan getters untuk property $taste dan method itsTaste().
4. Buat property $height yang berisikan nilai integer pada class Orange yang akan selalu tetap nilainya walaupun dilakukan instance berkali-kali.
5. Tugas 2:
   1. Tentukan kelas yang disebut Person.
   2. Karena semua Persons adalah dari species "Homo Sapiens", jadikan itu sebagai konstanta kelas
   3. Deklarasikan (tetapi belum ditentukan ) 3 properti kelas , dan \*Mereka semua harus \* . $name $age $occupationpublic
   4. Tentukan konstruktor kelas yang menerima tepat tiga argumen dengan urutan sebagai berikut: $name, $age, $occupationdan simpan di properti masing-masing.
   5. Tentukan metode yang dipanggil introduce yang tidak menerima argumen dan mengembalikan string format "Hello, my name is NAME\_HERE"
   6. Tetapkan metode lain yang disebut describe\_job yang tidak menerima argumen dan mengembalikan format string "I am currently working as a(n) OCCUPATION\_HERE"( CATATAN: "a(n)"Bagian dari string adalah literal yang berarti Anda tidak perlu membuat persyaratan untuk memeriksa apakah "a"atau "an"harus digunakan. )
   7. Ketika makhluk luar angkasa tiba di Bumi, semua Persons diharapkan untuk menyambut mereka dengan cara yang persis sama. Oleh karena itu, tentukan metode kelas statis yang disebut greet\_extraterrestrials yang menerima argumen $species dan mengembalikan string format "Welcome to Planet Earth SPECIES\_NAME\_HERE!"
6. hasil dan analisis

Praktikum ini melakukan coding di VSC menggunakan bahasa pemrograman php masih pada materi OOP yaitu Visibility, Setters and Getters, Static Keyword dan Constans.

Praktikum dilakukan setelah memahami modul dan mengerjakan free test, lalu melakukan coding dengan perintah sebagai berikut :

* + - 1. Tambahkan class Orange yang merupakan child class dari class Fruit, kemudian tambahkan property $taste dan method itsTaste() yang menampilkan output dari method get\_name() dan get\_color() dengan menerapkan konsep overriding.
      2. Buat agar property $taste dan method itsTaste() hanya bisa diakses oleh class Orange saja dan buat agar property dan method pada class parent hanya bisa diakses oleh child dan parent itu sendiri.
      3. Buat settters dan getters untuk property $taste dan method itsTaste().

Buat property $height yang berisikan nilai integer pada class Orange yang akan selalu tetap nilainya walaupun dilakukan instance berkali-kali.

<?php

class Fruit {

protected $name;

protected $color;

function \_\_construct($name, $color)

{

$this->name = $name;

$this->color = $color;

}

function get\_name() {

return $this->name;

}

function get\_color() {

return $this->color;

}

function getColor()

{

return "$this->color, $this->name";

}

function setColor($color, $name)

{

$this->color = $color;

$this->name = $name;

}

}

class Orange extends Fruit {

private $teste;

public static $height = 4.12;

public function nameFruit()

{

return "Orange" . self::$height;

}

function \_\_construct($name, $color, $teste)

{

parent:: \_\_construct($name, $color);

$this->teste = $teste;

}

function itsTeste()

{

return "$this->name colored $this->color have teste $this->teste";

}

function getTaste()

{

return "$this->teste";

}

function setTaste()

{

$this->teste = $teste;

}

}

$fruit = new Fruit("Orange", "Oren");

$orange = new Orange("Orange", "Oren", "Sweet");

$orange->setColor("Oren", "Orange");

//var\_dump($fruit);

echo $orange->itsTeste();

echo "<br>";

echo $orange->getColor();

echo "<br>";

echo Orange::$height;

//echo $orange->setTeste("Sweet");

Codingan di atas menunjukan penerapan Visibility pada property class masing-masing yang berfungsi sebagai ketentuan pengaksesan pada class yang di inginkan sesuai dengan visibility masing-masing seperti pada class fruit property nya menerapkan visibility protected sehingga nilai dari property fruit dapat di akses class fruit dan turunan nya atau child class nya dan pada child class nya yaitu class orange menerapkan visibility private sehingga nilai yang ada pada properti pada class orange ini hanya dapat di inisiasi oleh class itu sendiri. Kemudian codingan di atas di terapkan setters and getters dan juga static keyword pada properti child class orange.

Setelah selesai melengkapi codingan pada modul dengan perintah 1,2,3, dan 4 di atas di lanjutkan dengan perintah ke 5 yaitu.

1. Tugas 2:
   1. Tentukan kelas yang disebut Person.
   2. Karena semua Persons adalah dari species "Homo Sapiens", jadikan itu sebagai konstanta kelas
   3. Deklarasikan (tetapi belum ditentukan ) 3 properti kelas , dan \*Mereka semua harus \* . $name $age $occupationpublic
   4. Tentukan konstruktor kelas yang menerima tepat tiga argumen dengan urutan sebagai berikut: $name, $age, $occupationdan simpan di properti masing-masing.
   5. Tentukan metode yang dipanggil introduce yang tidak menerima argumen dan mengembalikan string format "Hello, my name is NAME\_HERE"
   6. Tetapkan metode lain yang disebut describe\_job yang tidak menerima argumen dan mengembalikan format string "I am currently working as a(n) OCCUPATION\_HERE"( CATATAN: "a(n)"Bagian dari string adalah literal yang berarti Anda tidak perlu membuat persyaratan untuk memeriksa apakah "a"atau "an"harus digunakan. )
   7. Ketika makhluk luar angkasa tiba di Bumi, semua Persons diharapkan untuk menyambut mereka dengan cara yang persis sama. Oleh karena itu, tentukan metode kelas statis yang disebut greet\_extraterrestrials yang menerima argumen $species dan mengembalikan string format "Welcome to Planet Earth SPECIES\_NAME\_HERE!"

Berikut hasil tugas no 5 pada modul :

class Person {

const SPECIES = "Homo Sapiens";

public $name;

public $age;

public $occupation;

function \_\_construct($name = 0, $age = 0, $occupation = 0)

{

$this->name = $name;

$this->age = $age;

$this->occupation = $occupation;

}

function introduce(){

return "Hello, my name is {$this->name}";

}

function describe\_job(){

return "I am currently working as a {$this->occupation}";

}

function greet\_extraterrestrials(){

return "Welcome to Planet Earth ".self::SPECIES;

}

}

$Persons = new Person( "Rizky Saputra",21,"programmer");

echo $Persons->introduce();

echo "<br/>";

echo $Persons->describe\_job();

echo "<br/>";

echo $Persons->greet\_extraterrestrials();

Codingan di atas menunjukan Hasil dari perintah no 5 dengan class Person yang menerapkan cons pada property species yang berfungsi sebagai nilai tetap pada property nya, lalu menambahkan method.

1. Kesimpulan

Pada praktikum ini dapat di simpulkan :

OOP adalah paradigma pemrograman yang memecahkan masalah pemrograman dengan menyediakan objek (terdiri dari beberapa atribut dan metode) yang terkait dan diatur dalam satu grup atau class. Nantinya, objek-objek tersebut berinteraksi satu sama lain untuk memecahkan masalah pemrograman yang kompleks. Dari praktikum sejauh ini praktikum pertama,kedua dan ketiga ini mempelajari OOP dari mulai class, object, property, method, constructur, object type, inheritance dan overriding, visibility, setters and getters, static keyword dan constans.

VI. Daftar Pustaka

1. <https://www.w3schools.com/php/php_constants.asp>
2. <https://www.warungbelajar.com/tutorial-belajar-php>
3. [https://ilmu-detil.blogspot.com/2016/06/memahami](https://ilmu-detil.blogspot.com/2016/06/memahami-tipe-data-object-pada-php) php
4. Aziz, M. Farid. 2005. Object Oriented Programing dengan PHP5, Jakarta: PT Elex Media Komputindo
5. <https://kodingin.com/overriding-pada-php-oop/>
6. [https://www.malasngoding.com/php](https://www.malasngoding.com/php-oop-pewarisan-sifat-inheritance)