Laporan Praktikum Pemrograman II

Modul II – Constructor, Object Type, Inheritance, dan Overriding

Rizki Saputra/22103001007

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

Tanggal praktikum: senin, 27 Maret 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) Constructor, Object Type, Inheritance dan Overiding. Hasil codingan tersebut dapat di lihat menggunakan local host XAMPP dan dapat di tampilkan pada browser firefox.

1. Tinjauan Pustaka
   1. Constructor

Constructor adalah method atau function yang otomatis akan dijalankan saat class diinstansiasi (dibuat sebuah object), pada bagian constructor kita dapat melakukan apapun yang bisa dilakukan dalam method / function kecuali mengembalikan nilai / return value(3). Konstruktor memungkinkan Anda menginisialisasi properti objek setelah objek dibuat, Jika Anda membuat sebuah \_\_construct()fungsi, PHP akan secara otomatis memanggil fungsi ini saat Anda membuat objek dari sebuah kelas(4).

* 1. Object Type

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(5).

* 1. Inheritance

Inheritance merupakan class turunan dari class yang lainnya, Inheritance di buat dengan membuat class baru dengan tujuan di hubungkan dengan class yang lain yang di inginkan dengan tujuan untuk mendapatkan informasi atau data dari class yang di hubungkan tersebut, class baru ini di sebut dengan child class(6).

* 1. Overring

Overriding adalah sebuah teknik PHP dengan konsep OOP yang mengimplementasikan pembuatan method yang sama pada class child dengan method parentnya. Kenapa harus menggunakan overriding ? Terkadang ada kebutuhan untuk membuat fungsi yang sama namun keadaan yang berbeda. Syarat untuk menggunakan teknik overriding yaitu harus mempunyai class parent dan class child(7).

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. Terapkan constructor untuk class fruit pada modul.
2. Buat class baru bernama DetailFruit yang menerapkan konsep *Object Type class Fruit* pada *method* detail() dan kemudian *method* tersebut akan menampilkan output “I love $name and its $color.
3. Buat *Class Apple* dan *Class Banana* yang merupakan *Child* dari *Class Fruit*. Masing-masing *Child Class* mempunyai *property* $shape dan *method* detail() untuk menampilkan output: “I Love $name, its $color, and its $shape”. Terapkan juga konsep *Overriding* untuk membuat baris program menjadi lebih rapi dan tidak diulang-ulang
4. Buat *Class* Kalkulator yang memiliki tiga *Child Class*, yaitu Tambah, Kurang, dan Bagi. *Parent Class* mempunyai *property* $angka1 dan $angka2. Semua *Class* mempunyai *method* hasil(). *Method* hasil() pada *Parent Class* hanya menampilkan angka saja tanpa dilakukan proses aritmatika, sedangkan *Child Class* melakukan proses aritmatika sesuai dengan nama *Class*-nya.
5. hasil dan analisis

Praktikum ini melakukan coding di VSC menggunakan bahasa pemrograman php pada materi OOP yaitu Constructor, Object Type, Inheritance dan Overiding.

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

* + - 1. Terapkan constructor untuk class fruit pada modul.
      2. Buat class baru bernama DetailFruit yang menerapkan konsep *Object Type class Fruit* pada *method* detail() dan kemudian *method* tersebut akan menampilkan output “I love $name and its $color.
      3. Buat *Class Apple* dan *Class Banana* yang merupakan *Child* dari *Class Fruit*. Masing-masing *Child Class* mempunyai *property* $shape dan *method* detail() untuk menampilkan output: “I Love $name, its $color, and its $shape”. Terapkan juga konsep *Overriding* untuk membuat baris program menjadi lebih rapi dan tidak diulang-ulang.

Berikut hasil perintah di atas pada VSC di bawah ini.

<?php

class Fruit {

    public $name;

    public $color;

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

    {

        $this->name = $name;

        $this->color = $color;

    }

    function get\_name()

    {

        return $this->name;

    }

    function get\_color()

    {

        return $this->color;

    }

}

class DetailFruit {

    function detail(Fruit $fruit)

    {

        return "I love {$fruit->name} and ist {$fruit->color}";

    }

}

class Apple extends Fruit {

    public $shape;

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

    {

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

        $this->shape = $shape;

    }

    function detail()

    {

        return "I Love $this->name and its $this->color and ist $this->shape";

    }

}

class Banana extends Fruit {

    public $shape;

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

    {

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

        $this->shape = $shape;

    }

    function detail()

    {

        return "I Love $this->name and its $this->color and its $this->shape";

    }

}

$apple = new Apple("Apple", "Green", "Oval");

$banana = new Banana("Banana", "Yellow", "Long");

$detail = new Fruit("Apple & Banana", "Green & Yellow", "Oval & Long");

echo $apple->detail();

echo "<br>";

echo $banana->detail();

echo "<br>";

echo $detail->get\_name();

echo "<br>";

echo $detail->get\_color();

Codingan di atas menunjukan penerapan constructor berfungsi sebagai inisialisasi nilai awal dari objek dan menetapkan nilai awal untuk properti objek. Kemudian terdapat penerapan object type pada detailFruit( ). Lalu menambahkan child class di luar class fruit yaitu class Apple dan Banana memiliki property $shapes dan method detail pada keduanya. Codingan pada vsc di atas menghasilkan pada browser “I Love Apple and its Green and its Oval” untuk child class Apple, dan “I Love Banana and its Yellow and its Long” untuk child class Banana.

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

* + - 1. Buat *Class* Kalkulator yang memiliki tiga *Child Class*, yaitu Tambah, Kurang, dan Bagi. *Parent Class* mempunyai *property* $angka1 dan $angka2. Semua *Class* mempunyai *method* hasil(). *Method* hasil() pada *Parent Class* hanya menampilkan angka saja tanpa dilakukan proses aritmatika, sedangkan *Child Class* melakukan proses aritmatika sesuai dengan nama *Class*-nya.

Berikut hasil perintah no 4 pada modul :

<?php

class Kalkulator {

    public $angka1;

    public $angka2;

    function \_\_construct($angka1 = 0, $angka2 = 0)

    {

        $this->angka1 = $angka1;

        $this->angka2 = $angka2;

    }

    function getData()

    {

        return "$this->angka1, $this->angka2";

    }

}

class Tambah extends Kalkulator {

    function hasilTambah()

    {

        return $this->angka1 + $this->angka2;

    }

}

class Kurang extends Kalkulator {

    function hasilKurang()

    {

        return $this->angka1 - $this->angka2;

    }

}

class Bagi extends Kalkulator {

    function hasilBagi()

    {

        return $this->angka1 / $this->angka2;

    }

}

$kalkulator = new Kalkulator();

$tambah = new Tambah(1,1);

$kurang = new Kurang(10,5);

$bagi = new Bagi(10,2);

//echo $tambah->hasilTambah();

//echo $kurang->hasilKurang();

echo $bagi->hasilBagi();

Codingan di atas menunjukan Hasil dari perintah no 4 dengan class Kalkulator yang mempunyai child class Tambah, Kurang dan Bagi dengan menerapkan aritmatika pada setiap method nya sesuai dngan nama class yang telah di tentukan. Codingan di atas dapat melakukan proses perhitungsn sesuai dengan class nya yaitu , penjumlahan, pengurangan, pembagian.

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 praktikum pertama dan kedua ini mempelajari OOP dari mulai class, object, property, method, constructur, object type, inheritance dan overriding.

VI. Daftar Pustaka

1. <https://ejournal.bsi.ac.id/ejurnal/index.php>
2. Aziz, M. Farid. 2005. Object Oriented Programing dengan PHP5, Jakarta: PT Elex Media Komputindo
3. <https://www.warungbelajar.com/tutorial-belajar-php-part-36-mengenal-fungsi-constructor>
4. <https://www.w3schools.com/php/php_oop_constructor.asp>
5. <https://ilmu-detil.blogspot.com/2016/06/memahami-tipe-data-object-pada-php>
6. <https://www.malasngoding.com/php-oop-pewarisan-sifat-inheritance>
7. <https://kodingin.com/overriding-pada-php-oop/>