Laporan Praktikum Pemrograman II

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

Muhamad Parhan Susilo/22103001002

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

Tanggal praktikum: senin, 27 Maret 2023

Fsusilo128@gmail.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

Pemrograman berorientasi objek (Pemrograman berorientasi objek atau OOP) adalah paradigma pemrograman berorientasi objek. Objek adalah struktur data yang terdiri dari bidang data dan metode serta interaksinya dalam desain aplikasi dan program komputer. Semua data dan fungsi dalam paradigma tersebut dikemas dalam kelas atau objek, sebanding dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lain. Banyak bahasa pemrograman saat ini mendukung OOP.

OOP adalah paradigma pemrograman yang cukup dominan saat ini karena kemampuannya memberikan solusi aturan pemrograman modern. Namun, ini tidak berarti bahwa pemrograman prosedural tidak lagi memungkinkan. OOP dibuat karena diyakini bahwa bahasa pemrograman tradisional masih memiliki keterbatasan. Konsep OOP sendiri adalah semua pemecahan masalah dipecah menjadi objek. Dalam OOP, data dan fungsi yang menggunakannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai objek. Proses editing atau proses desain dalam pemrograman merupakan proses yang tidak terpisahkan dari proses sebelumnya yaitu proses analisis dan proses selanjutnya. Pembahasan orientasi objek tidak lepas dari konsep objek, seperti,warisan atau mengurangi enkapsulasi atau kemasan dan polimorfisme atau semacam itu. Konsep-konsep ini sangat penting untuk orientasi objek dan diperlukan untuk memahami.

1. Tinjauan Pustaka
   1. Constructor

Constructor adalah fungsi anggota khusus yang dijalankan secara otomatis saat objek dibuat. selama proses instantiasi. Pembangun umumnya digunakan untuk membuat proses awal dalam persiapan objek, menugaskan nilai ke anggota data, memanggil fungsi anggota internal dan proses lain yang dianggap perlu. constructor dibuat dengan cara menulis sebuah member function yang namanya sama dengan nama class. Sebagai contoh jika kita memiliki class Laptop, maka function constructor juga harus bernama Laptop(). Sebuah constructor tidak mengembalikan nilai sehingga tidak perlu menulis tipe data sebelum nama function. Constuctor juga harus di set sebagai public, jika tidak kita tidak bisa men-instansiasi class tersebut.

* 1. Object Type

Object type adalah tipe data yang tidak hanya menyimpan data, tetapi juga dapat berisi informasi tentang bagaimana data yang ada dapat dimanipulasi. Secara umum, tipe data seperti float, integer, dan string hanya berisi tipe datanya sendiri, tetapi tipe data yang termasuk dalam tipe data objek dapat berisi tipe data lainnya..

* 1. Inheritance

Inheritance adalah konsep OOP dimana sebuah class dapat menurunkan property dan method yang dimilikinya kepada class lain. Konsep inheritance dipakai untuk memanfaatkan fitur code reuse, yakni menghindari terjadinya duplikasi kode program.Konsep inheritance membuat sebuah struktur atau hierarchy class dalam kode program. Class yang akan diturunkan bisa disebut sebagai class induk (parent class), super class, atau base class.

* 1. Overring

Overriding adalah 'memaksa' suatu method pada child class agar bekerja pada spesifikasi sendiri yang tidak sama dengan parent class-nya, Overriding di OOP ini berkaitan erat dengan Inheritance.

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 :

Pemrograman berorientasi objek (Object Oriented Programming atau disingkat OOP) adalah paradigma pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus meningkat. Objek adalah entitas yang memiliki atribut, karakter (bahavour) dan kadang kala disertai kondisi (state).

VI. Daftar Pustaka

1. <https://ejournal.bsi.ac.id/ejurnal/index.php>
2. [Summary of Pemrograman Berorientasi Objek (uinsu.ac.id)](https://elearning.uinsu.ac.id/course/info.php?id=2750)
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/>