

Nama : Tanya Dwi Agustin

NPM : G1F022014

Kelas : B

Proyek Pemrograman Berorientasi Objek

RESPONSI PEMROGRAMAN BERORIENTASI OBJEK

- Data

1) Conflict

```
data > conflict.php
1  <?php
2
3  // buat namespace data\satu
4  namespace data\satu {
5  // dengan class conflict
6  class conflict
7  {
8  }
9  // class sample
10 class stream_filter_prepend
11 {
12 }
13 // class dummy
14 class dummy
15 {
16 }
17 }
18 // buat namespace data\dua
19 namespace data\dua{
20 // dengan class conflict
21 class conflict
22 {
23 }
24 }
```

Penjelasan :

Pada gambar di atas, terdapat pembuatan namespace yang dikenal sebagai ruang lingkup nama. Di mana namespace adalah kumpulan jenis elemen dan nama atribut, yang masing-masing dapat diidentifikasi oleh namespace unik yang menjadi milik mereka. Namespace memungkinkan untuk membedakan objek dengan nama yang sama tetapi asal yang berbeda. Di mana di atas terdapat pembuatan namespace data\satu dengan class conflict, class sample dan class dummy. Kemudian gambar tersebut dibuat juga pada namespace data\dua yang berisi class conflict.

2) Helper

```
data > helper.php
1  <?php
2
3  namespace Helper;
4
5  function helpMe()
6  {
7      echo "HELP ME" . PHP_EOL;
8  }
9
10 const APPLICATION = "Belajar PHP OOP";
```

Penjelasan :

Pada gambar di atas diawali dengan cod PHP (Hypertext Preprocessor) yang berarti sebuah bahasa pemrograman server side scripting yang bersifat open source, PHP ini menjalankan proses pemrograman pada saat proses runtime. Hasil dari proses tersebut akan berbeda tergantung data yang diproses. Kemudian sebuah program dibuat namespace Helper, dan function helpMe() yang digunakan untuk mengerjakan suatu pekerjaan tertentu atau akan dieksekusi ketika dipanggil dari bagian lain dalam suatu program tersebut. Berhubung terdapat tanda “()” yang berarti sebuah method. Kemudian terdapat echo yang berfungsi untuk menampilkan teks ke layar, seperti di atas terdapat echo “help me” artinya akan menampilkan help me pada layar. Const atau konstanta merupakan variabel yang sifatnya tetap atau dengan kata lain nama dan isi dari variabel tidak bisa di ubah, seperti di atas terdapat const dengan nama application.

3) Manager

```
data > manager.php
2
3 // buat kelas manager dengan properti nama dan function sayHello
4 class Manager
5 {
6     var string $nama;
7
8     function sayHello(string $nama): void
9     {
10         echo "Hi $nama, my name is $this->nama" . PHP_EOL;
11     }
12 }
13
14 // buat kelas VicePresident dengan extends manager
15 class VicePresident extends Manager
16 {
17     var string $alamat;
18 }
19
```

Penjelasan :

Kemudian untuk bagian manager, terdapat sebuah class manager yang dapat dijelaskan bahwa class dalam program Java sendiri saling berhubungan dengan cara memberikan akses terhadap anggota mereka. Seperti di atas terdapat kata kunci var yang merupakan tipe statis, artinya, compiler hanya menginfer satu kali terhadap tipe data dari variabel dengan kata kunci var saat inialisasi. Kemudian untuk function mengerjakan suatu kegiatan yaitu sayHello dengan nama tiap void yaitu pengembalian. Kemudian terdapat perintah echo untuk menampilkan nama. Kemudian di atas erjadinya pembuatan class vicePresident dengan extend Manager, di mana bahwasannya extends pada Java berfungsi untuk memanggil fungsi dari class lain, sehingga kita tidak perlu lagi membuat script yang sama pada class yang akan kita buat. Kemudian dalam class vicePresident terdapat kata kunci var string alamat.

4) Person

```

data > person.php
1  <?php
2
3  // membuat kelas person
4  class Person{
5      // membuat properti
6      var string $nama;
7      // gunakan nullable properti
8      var ?string $alamat = null;
9      // gunakan default value untuk properti
10     var string $negara = "indonesia";
11     // buat function sayHello
12     function sayHello(string $nama){
13         echo "Hello $nama" . PHP_EOL;
14     }
15     // buat function sayHello nullable dengan percabangan
16     function sayHelloNull(?string $nama)
17     {
18         if (is_null($nama)) {
19             echo "Hi, my nama is $this->nama" . PHP_EOL;
20         } else {
21             echo "Hi $nama, my nama is $this->nama" . PHP_EOL;
22         }
23     }
24     // buat const author
25     const AUTHOR = "Tanya Dwi Agustin";
26     // buat function info untuk self keyword
27     function info()
28     {
29         echo "Author : " . self::AUTHOR . PHP_EOL;
30     }
31     // buat function constructor
32     function __construct(string $nama, ?string $alamat)
33     {
34         $this->nama = $nama;
35         $this->alamat = $alamat;
36     }
37     // buat function destructor
38
39     function __destruct()
40     {
41         echo "Object person $this->nama is destroyed" . PHP_EOL;
42     }
43

```

Penjelasan :

Pada proses di atas terdapat PHP untuk menjalankan proses pemrograman pada saat proses runtime. Dengan membuat sebuah class yang berfungsi untuk saling berhubungan dengan cara memberikan akses terhadap anggota mereka. Di mana dengan kelas nama, dan ikuti sebuah objek yaitu, nama. Alamat, dan negara bertipe var string. Kemudian terdapat function sayHello dengan tipe string yang berfungsi untuk melakukan program yang dirancang. Kemudian terdapat pembuatan function sayHello dengan percabangan yaitu menggunakan if, else dan echo untuk memanggil program tersebut pada layar. Untuk fungsi if else sendiri yaitu terjadi dalam setiap kondisi, di mana isi variabel nilai akan di diperiksa apakah berisi sesuai dengan karakter atau tidak. Jika salah satu kondisi terpenuhi, maka block kode program yang sesuai akan di eksekusi. Setelah itu terdapat function constructor yang berisi nama dan alamat dengan tipe string. Kemudian dibuatnya function destructor sebagai method atau berguna dalam mengerjakan suatu pekerjaan tertentu atau akan dieksekusi ketika dipanggil, dengan bantuan echo untuk menampilkan pada layar pada objek orang yang berisi nama.

5) Product

```

data > product.php
1  <?php
2
3  class Product
4  {
5      protected string $name;
6      protected int $price;
7
8      public function __construct(string $name, int $price)
9      {
10         $this->name = $name;
11         $this->price = $price;
12     }
13
14     public function getName(): string
15     {
16         return $this->name;
17     }
18
19     public function getPrice(): int
20     {
21         return $this->price;
22     }
23 }
24
25 class ProductDummy extends Product
26 {
27
28     public function info()
29     {
30         echo "Name $this->name" . PHP_EOL;
31         echo "Price $this->price" . PHP_EOL;
32     }
33 }
34

```

Penjelasan :

Pada gambar di atas diawali dengan <?php yang menjalankan proses pemrograman pada saat proses runtime. Kemudian terdapat pembuatan class product dengan objek nama dengan tipe string dan objek price dengan tipe int, kemudian terdapat public function yang berarti bersifat public dengan nama dan price. Kemudian terdapat return sebagai method yang mengembalikan nilai secara langsung atau sebuah nilai dari variable dari nama dan nilai dari price. Kemudian membuat class ProductDummy dengan memanggil fungsi dari kelas lain product. Dengan function digunakan agar suatu bagian daripada suatu program bisa digunakan berulang kali. Function yang tidak menghasilkan nilai apapun bisa disebut dengan method(metode). Di mana terdapat echo untuk menampilkan output nama dan price ada php.

6) Programmer

```

data > programmer.php
1  <?php
2
3  class Programmer
4  {
5
6      public string $name;
7
8      public function __construct(string $name)
9      {
10         $this->name = $name;
11     }
12 }
13
14
15 class BackendProgrammer extends Programmer
16 {
17 }
18
19 class FrontendProgrammer extends Programmer
20 {
21 }
22
23 class Company
24 {
25     public Programmer $programmer;
26 }
27 function sayHelloProgrammer(Programmer $programmer)
28 {
29     if ($programmer instanceof BackendProgrammer) {
30         echo "Hello Backend Programmer $programmer->name" . PHP_EOL;
31     } else if ($programmer instanceof FrontendProgrammer) {
32         echo "Hello Frontend Programmer $programmer->name" . PHP_EOL;
33     } else if ($programmer instanceof Programmer) {
34         echo "Hello Programmer $programmer->name" . PHP_EOL;
35     }
36 }

```

Penjelasan :

Gambar di atas terdapat php yang menjalankan proses pemrograman pada saat proses runtime. Dengan class programmer dan objek pada nama yang bertipe string, kemudian terdapat public function pada konstruktor yang berfungsi untuk dipanggil pada objek setelah objek dibuat, dan merupakan tempat yang baik untuk meletakkan kode inisialisasi. Kemudian terdapat class backendprogrammer dan frontendprogrammer yang memanggil dari kelas lain pada programmer. Kemudian terdapat class company yang menggunakan if else, di mana suatu opsi akan dijalankan sesuai maka perintah if akan dijalankan, dan sebaliknya jika tidak sesuai maka perintah else yang akan dijalankan. Kemudian echo berfungsi untuk memanggil perintah untuk menampilkan pada layar tersebut.

7) Shape

```
data > shape.php
1  <?php
2
3  namespace Data;
4
5  class Shape
6  {
7
8      public function getCorner()
9      {
10         return -1;
11     }
12 }
13
14
15 class Rectangle extends Shape
16 {
17
18     public function getCorner()
19     {
20         return 4;
21     }
22
23     public function getParentCorner()
24     {
25         return parent::getCorner();
26     }
27 }
28
```

Penjelasan :

Pada gambar di atas diawali dengan php yang menjalankan proses pemrograman pada saat proses runtime. Terdapat namespace data, di mana namespace berfungsi untuk menyediakan alat luar biasa untuk mengatur kode dan membuat program lebih mudah dibaca. Kemudian terdapat public function getCorner sebagai method yang berguna untuk mengembalikan Component di sudut yang ditentukan. Seperti di atas yaitu return -1. Kemudian terdapat class rectangle yang memanggil dari kelas lain pada shape. Dengan return 4.

- Constant

```
constant.php
1  <?php
2
3  // import data/person.php
4  require_once "data/Person.php";
5
6  // buat define
7  define("TITLE", "Responsi Pemrograman Berorientasi Objek");
8
9  // buat const app version
10 const APP_VERSION = "1.0.0";
11
12 // tampilkan hasil
13 echo TITLE . PHP_EOL;
14 echo APP_VERSION . PHP_EOL;
15 echo Person::AUTHOR . PHP_EOL;
16
```

Penjelasan :

Gambar di atas terdapat php yang menjalankan proses pemrograman pada saat proses runtime. Gambar tersebut terdapat `require_once` sama seperti `require`, namun kali ini, file hanya disertakan sekali, sehingga jika ada beberapa kali `require_once`, maka hanya file pertama yang disertakan. Di mana file pertamanya yaitu `data/person`. Dengan definisi berisi title, yaitu "Responsi Pemrograman Berorientasi Objek". Dengan `const app_versionnya 1.0.0`. kemudian untuk menampilkannya dengan code `echo`.

- Constructor

```
1  <?php
2
3  // import data/person.php
4  require_once "data/Person.php";
5
6  // buat object new person dengan 2 parameter
7  $tanya = new Person("Tanya Dwi Agustin", "Lebong");
8
9  // vardump object
10 var_dump($tanya);
11
```

Penjelasan :

Pada gambar di atas diawali dengan php yang menjalankan proses pemrograman pada saat proses runtime. Dengan mengimport `data/person` menggunakan `require_once` yang hanya file pertama disertakan. Kemudian membuat new objek dua yang berisi nama dan alamat, di mana ber jenis string dan int sehingga terdapat `var_dump` berfungsi digunakan hanya untuk proses debugging , dimana ketika anda perlu mengetahui sebuah struktur berupa nilai dan tipe data dari suatu variabel.

- Destructor

```
1  <?php
2
3  // import data/person.php
4  require_once "data/Person.php";
5
6  // buat 2 object new peson dengan parameter yang berbeda
7  $tanya = new Person("Tanya Dwi Agustin", "Lebong");
8  $Dio = new Person("Dio Sinebha", "Curup");
9
10 // tambahkan echo "Program Selesai" . PHP_EOL;
11 echo "Program Selesai" . PHP_EOL;
12
```

Penjelasan :

Gambar di atas terdapat php yang menjalankan proses pemrograman pada saat proses runtime. Dengan mengimport `data/person` `require_once` yang disertakan, kemudian terdapat dua objek new person yang ditulis, di mana `person1` yaitu Tanya Dwi Agustin dengan alamat Lebong dan `person2` yaitu Dio Sinebha dengan alamat Curup. Kemudian ditambahkan `echo`

untuk memanggil dan menampilkan kalimat “Program Selesai” pada sebuah sistem yang dijalankan.

- Function

```
function.php
1  <?php
2
3  // import data/person.php
4  require "data/person.php";
5
6  // buat object baru dari kelas person
7  $person1 = new Person("Tanya Dwi Agustin","Lebong");
8
9  // panggil function
10 $person1->sayHello("Tanya Dwi Agustin");
11
```

Penjelasan:

Pada gambar di atas diawali dengan php yang menjalankan proses pemrograman pada saat proses runtime. Dengan mengimport data pertama yaitu person dengan code require. Kemudian dibuat objek pada kelas person dengan ketentuan person1 pada gambar sebelumnya yaitu berisi nama Tanya Dwi Agustin dengan beralamatkan Lebong. Kemudian function ini digunakan pada kode program yang dirancang untuk menyelesaikan suatu tugas tertentu. Seperti pada kelas person dengan sayHello kepada (“Tanya Dwi Agustin”) dengan diletakkan tanda ; untuk menandai akhir dari sebuah perintah.

- Import

```
import.php
1  <?php
2
3  require_once "data/Conflict.php";
4  require_once "data/Helper.php";
5
6  use data\ satu\ Conflict;
7  use function Helper\helpMe;
8  use const Helper\APPLICATION;
9
10 $conflict1 = new Conflict();
11 $conflict2 = new data\dua\Conflict();
12
13 helpMe();
14
15 echo APPLICATION . PHP_EOL;
```

Penjelasan :

Gambar di atas terdapat php yang menjalankan proses pemrograman pada saat proses runtime. Dengan code require_once yang berarti membutuhkan file tersebut. Di mana file yang dibutuhkan adalah data/conflict dan data/helper. Kemudian menggunakan use pada file data/satu/conflict, dan use function pada file helper/helpMe, kemudian use const pada helper.application. kemudian gambar di atas terdapat pembuatan conflict baru yaitu conflict1 dan conflict2. Dan terdapat method helpme yang dikerjakan dengan code echo untuk memanggilkan pada layar di atas.

- importAlias

```

importAlias.php
1  <?php
2
3  require_once "data/Conflict.php";
4  require_once "data/Helper.php";
5
6  use data\satu\Conflict as Conflict1;
7  use data\dua\Conflict as Conflict2;
8  use function Helper\helpMe as help;
9  use const Helper\APPLICATION as APP;
10
11  $conflict1 = new Conflict1();
12  $conflict2 = new Conflict2();
13
14  help();
15
16  echo APP . PHP_EOL;

```

Penjelasan :

Import adalah sesuatu yang membuat kita dapat menggunakan kode dari kelas yang berada di paket berbeda. Pada gambar di atas diawali dengan php yang menjalankan proses pemrograman pada saat proses runtime. Di mana file yang dibutuhkan adalah data/conflict dan data/helper. Dengan menambah conflict baru yaitu conflict1 dan conflict2. Kemudian terdapat method help yang dikerjakan dengan code echo untuk memanggilnya pada layar di atas.

- Inheritance

```

inheritance.php
1  <?php
2
3  // import data/person.php
4  require_once "data/Manager.php";
5
6  // buat object new manager dan tambahkan value nama kemudian panggil function
7  $manager = new Manager();
8  $manager->nama = "Tanya";
9  $manager->sayHello("Dio");
10
11 // buat object new vicepresident dan tambahkan value nama kemudian panggil function
12 $vicePresident1 = new vicePresident();
13 $vicePresident1->nama = "Dio";
14 $vicePresident1->alamat = "Curup";
15 $vicePresident1->sayHello("Tanya");
16

```

Penjelasan :

Inheritance adalah cara membuat salinan kelas yang sudah ada sebagai titik awal untuk kelas yang lain. Selain istilah ‘subclass’, kelas inheritance juga disebut kelas turunan. Gambar di atas terdapat php yang menjalankan proses pemrograman pada saat proses runtime. Dengan mengimport file data/manager, kemudian membuat manager baru dengan nama Tanya dan manager melakukan kegiatan sayHello terhadap Dio. Kemudian untuk cabangnya yaitu dengan membuat vicePresident baru pada nama Dio dengan alamat Curup. Kemudian tugas diatas dibalikkan sehingga pada objek baru ini melakukan perintah sayHello terhadap Tanya.

- nameSpace


```

nameSpace.php
1  <?php
2
3  // buat namespace
4  // import data dari conflict
5  require "data/conflict.php";
6  // buat oobject dari namespace yang di buat
7  $conflict1 = new data\satu\conflict();
8  $conflict1 = new data\dua\conflict();
9
10 // import data helper
11 require "data/helper.php";
12 // tampilkan helper menggunakan echo
13 echo Helper\APPLICATION . PHP_EOL;
14 // masukan Helper\helpMe();
15 Helper\helpMe();

```

Penjelasan :

Namespace digunakan untuk secara unik mengidentifikasi satu atau lebih nama dari nama serupa lainnya dari berbagai objek, grup atau namespace secara umum. Dengan php yang menjalankan proses pemrograman pada saat proses runtime. Dengan mengimport file data/conflict dari objek . kemudian terdapat file data/helper yang kemudian digunakan echo untuk menampilkan echo\application dengan method Helper/helpMe.

- Object

```

object.php
1  <?php
2
3  // import data/person.php
4  require "data/person.php";
5
6  // buat object baru dari kelas person
7  $person = new Person("Tanya Dwi Agustin","Lebong");
8
9  // manipulasi properti nama, alamat, negara
10 $person1->nama = "Tanya Dwi Agustin";
11
12 // menampilkan hasil
13 echo "nama = {$person->nama}";

```

Penjelasan :

Objek merupakan turunan dari kelas. Di mana untuk import data tersebut dibutuhkannya file data/person. Kemudian membuat objek dengan person baru yang berisi nama dan alamat. Kemudian untuk memanggilnya menggunakan echo.

- Parents

```

parent.php
1  <?php
2
3  require_once "data/Shape.php";
4
5  use Data\{Shape, Rectangle};
6
7  $shape = new Shape();
8  echo $shape->getCorner() . PHP_EOL;
9
10 $rectangle = new Rectangle();
11 echo $rectangle->getCorner() . PHP_EOL;
12 echo $rectangle->getParentCorner() . PHP_EOL;

```

Penjelasan :

Gambar di atas terdapat php yang menjalankan proses pemrograman pada saat proses runtime. Dengan Class Shape adalah kelas abstrak, karena dia punya metode abstrak. Di atas pembuatan data pada shape dan rectangle.

- Polymorphism

```
polymorphism.php
1  <?php
2
3  require_once "data/Programmer.php";
4
5  $company = new Company();
6  $company->programmer = new Programmer("Tanya Dwi Agustin");
7  var_dump($company);
8
9  $company->programmer = new BackendProgrammer("Tanya Dwi Agustin");
10 var_dump($company);
11
12 $company->programmer = new FrontendProgrammer("Tanya Dwi Agustin");
13 var_dump($company);
14
15 sayHelloProgrammer(new Programmer("Tanya Dwi Agustin"));
16 sayHelloProgrammer(new BackendProgrammer("Tanya Dwi Agustin"));
17 sayHelloProgrammer(new FrontendProgrammer("Tanya Dwi Agustin"));
```

Penjelasan :

Gambar di atas terdapat file yang ingin ditampilkan yaitu file data/programmer dengan membuat objek baru yaitu company beserta methodnya. Seperti var_dump() berfungsi untuk memastikan tipe data dari suatu variabel dan untuk proses pengujian dan pencarian kesalahan. Dengan sayHelloProgrammer dengan membuat new programmer, backendprogrammer dan frontendprogrammer sebagai Tanya Dwi Agustin.

- Properti

```
properti.php
1  <?php
2
3  // import data/person.php
4  require_once "data/person.php";
5
6  // buat object baru dari kelas person
7  $person1 = new Person("Tanya Dwi Agustin","Lebong");
8
9  // manipulasi properti nama person
10 $person1->nama = "Tanya";
11
12 // menampilkan hasil
13 echo "Nama = {$person1->nama}" . PHP_EOL;
14 echo "Alamat = {$person1->alamat}" . PHP_EOL;
15 echo "Negara = {$person1->negara}" . PHP_EOL;
16
```

Penjelasan :

Dari properti tersebut terdapat file data/person dengan membuat kelas baru dari person dengan objek nama dan alamat. Kemudian untuk memanggil hasil diperlukan code echo sehingga ditampilkan pada layar.

- SelfKeyword

```

selfKeyword.php
1  <?php
2
3  // import data/person.php
4  require "data/person.php";
5
6  // buat object baru dari kelas person
7  $person1 = new Person("Tanya Dwi Agustin", "Lebong");
8
9  // panggil function
10 $person1->sayHello("Tanya Dwi Agustin");
11
12 // panggil self keyword
13 $person1->info() . PHP_EOL;
14

```

Penjelasan :

Pada gambar di atas terdapat php yang menjalankan proses pemrograman pada saat proses runtime. Dengan membuat require data/person dan objek baru pada kelas person dengan memanggil function sayHello pada nama yang telah ditulis. Kemudian untuk dipanggil maka diletakkan method pada info yang akan dipanggil sesuai dengan perintah.

- ThisKeyword

```

thisKeyword.php
C: > xampp > htdocs > bahan-ajar-pbo-main > thisKeyword.php
1  <?php
2
3  // import data/person.php
4  require_once "data/person.php";
5
6  // buat object dari kelas person
7  $person1 = new Person("Tanya Dwi Agustin", "Lebong");
8
9  // tambahkan value nama di object
10 $person1->nama = "Tanya Dwi Agustin";
11
12 // panggil function sayHelloNull dengan parameter
13 $person1->sayHelloNull("Dio Sinebha");
14
15 // buat object dari kelas person
16 $person2 = new Person("Dio Sinebha", "Curup");
17
18 // tambahkan value nama di object
19 $person2->nama = "Dio";
20
21 // panggil function sayHelloNull dengan parameter null
22 $person2->sayHelloNull(null);
23

```

Penjelasan :

Pada gambar di atas terdapat import data dan pembuatan objek baru dari kelas person. Kemudian untuk function di atas berguna sebagai method yaitu pemanggil person dengan nama yang berbeda dari tiap value yang dibuat.

- Visibility

```

visibility.php
1  <?php
2
3  require_once "data/Product.php";
4
5  $product = new Product("laptop", 1000000);
6
7  // tampilkan product get name
8  echo $product->getName() . PHP_EOL;
9  // tampilkan product get price
10 echo $product->getPrice() . PHP_EOL;
11 $dummy = new ProductDummy("Dummy", 1000000);
12 $dummy->info();

```

Penjelasan :

Pada gambar di atas terdapat php berfungsi untuk menjalankan proses pemrograman pada saat proses runtime. Dengan file data yang dipilih yaitu product, dengan membuat produk baru dengan menampilkan produk menggunakan echo product pada getName, dan getPrice. Kemudian membuat produk baru yaitu Data dummy merupakan data palsu yang diciptakan dengan tujuan sebagai data sampel dengan meletakkan method info yang dijalankan.