

RESPONSI
PROYEK PEMROGRAMAN BERORIENTASI OBJEK



Disusun Oleh :

Nama : Ikhsanudin Pratama

NPM : G1F022054

Asisten Dosen :

- | | |
|-----------------------|-------------|
| 1. Fadia Nur Shafitri | (G1F021010) |
| 2. Alvin Indrawan | (G1F021020) |

Dosen Pengampu :

1. Ferzha Putra Utama, S.T., M.Eng.
2. Arie Vatesia, S.T., M.TI., Ph.D.

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
T.A 2023/2024

```

1  <?php
2
3  // buat namespace data\satu
4  // dengan class conflict
5  // class sample
6  // class dummy
7  namespace data\satu;
8  class Conflict {
9      public function __construct() {
10         echo "ini adalah class " . __CLASS__;
11         echo "<br>";
12     }
13 }
14 $conflict1 = new Conflict();
15 class sample {
16 }
17 }
18
19 class dummy {
20 }
21 }
22 // buat namespace data\dua
23 // dengan class conflict
24 namespace data\dua;
25 class Conflict {
26     public function __construct() {
27         echo "ini adalah class " . __CLASS__;
28         echo "<br>";
29     }
30 }
31 $conflict2 = new Conflict();
32
33 spl_autoload_register(function($Conflict){
34     $Conflict = explode('\\', $Conflict);
35     $Conflict = end($Conflict);
36     require_once __DIR__ . '/data/satu/' . $Conflict . '.php';
37 });
38
39 spl_autoload_register(function($Conflict){
40     $Conflict = explode('\\', $Conflict);
41     $Conflict = end($Conflict);
42     require_once __DIR__ . '/data/dua/' . $Conflict . '.php';
43 });
44

```

G1F022054

Gambar 1. Data Conflict

Penjelasan :

Pada file PHP di atas merupakan file yang nantinya akan dipanggil dalam file lainnya, yang artinya file PHP di atas memiliki code yang berisikan isian yang akan memproses output yang akan ditampilkan pada localhost. Yang pasti, kita selalu mengawali pembuatan file program PHP dengan tanda “<?PHP” dan menutupnya dengan tanda “?>”. Akan tetapi, itu biasanya dilakukan ketika membuat PHP yang diselipkan di dalam HTML. Jika kita hanya membuat file PHP saja, maka tidak masalah ketika kita tidak menambahkan tanda penutup “?>”. Kita bisa melihat pada gambar 1 yang mana di sana terdapat beberapa class yang dibuat.

Tapi pada gambar tersebut kita dapat melihat bahwa terdapat dua kelas yang memiliki nama yang sama, yaitu “class conflict”. Itu akan menjadi eror biasanya. Kita bisa melakukan itu karena pada file PHP itu kita membuat namespace yang dijadikan sebagai pembatas akses atau tabir di antara dua class yang sama itu. Kemudian, di sana juga terdapat fungsi construct yang dapat menerima argument di dalam class yang kita buat. Namun, fungsi itu boleh saja tidak ditulis karena PHP akan otomatis menggunakan fungsi default-nya. Lalu, untuk dapat memunculkan isian dari class, kita perlu membuat objek untuk dipanggil. Seperti pada gambar di atas, kita bisa melihat pada namespace data\satu terdapat “\$Conflict1” yang digunakan sebagai objek. Tujuan dari pembuatan objek itu sendiri adalah untuk mempermudah kita untuk memanggil isian dari class conflict yang terdapat pada namespace tersebut. Kemudian, yang terakhir, di sana juga terdapat “spl_autoload_register()...” yang memungkinkan kita untuk memanggil lebih dari satu fungsi autoload sebelum dipanggil melalui code. Walaupun yang dipanggil hanya dengan menggunakan code “require_once “data/conflict”,” maka semua isian dari setiap class \$Conflict akan ditampilkan baik itu dari data\satu atau data\dua.



Gambar 2. Import Alias

Penjelasan :

Pada gambar di atas kita bisa melihat bahwasannya file PHP dari importAlias ini mengimport data dan juga fungsi dari file data/Conflict.php dan juga data/Helper.php. Karena pada file Conflict.php sudah dilakukan pemanggilan objek dan juga autoload, maka kita tidak

perlu repot untuk mendeklarasikan fungsinya kembali. Akan tetapi, berbeda halnya dengan file helper.php yang belum dilakukan autoload di dalamnya. Kita perlu melakukan deklarasi ulang pada fungsi yang terdapat pada file helper agar dapat memunculkan outputnya pada localhost. Untuk memudahkan proses inisiasi, kita dapat menggunakan fungsi alias dengan code “as” kemudian diikuti dengan alias yang ingin kita gunakan pada sebuah objek. Berulah di bagian akhir kita panggil atau kita deklarasikan fungsi untuk isian dari helper.php.



```
1  <?php
2
3  // buat namespace
4  namespace masalah;
5  // import data dari conflict
6  // buat oboject dari namespace yang di buat
7  require_once "data/Conflict.php";
8  require_once "data/helper.php";
9
10 class Conflict {
11     public function __const() {
12         echo "ini adalah class " . __CLASS__;
13     }
14 }
15
16 $str = new Conflict();
17
18 // import data helper
19 // tampilkan helper menggunakan echo
20 // masukan Helper\helpMe();
21 use function Helper\helpMe as help;
22 help();
```

G1F022054

Gambar 3. Name Space

Penjelasan :

Pada gambar 3 ini isinya hampir sama dengan import alias. Pada file ini kita membuat namespace baru dengan nama masalah. Dan pada program ini kita mencoba memanggil isi dari conflict dengan cara membuat atau mendeklarasikan sebuah objek baru. Dan sama dengan sebelumnya, pada helper juga kita deklarasikan isinya. Tapi, hanya pada bagian help saja dan application tidak kita panggil. Karena pada file helper.php fungsi echo untuk memanggil sudah dilakukan, maka kita tidak perlu repot untuk menambahkannya kembali. Kita bisa memanggilnya dengan hanya memanggil alias dari variabel yang mau kita tampilkan untuk menampilkan outputnya.