

Nama : Muhammad Syafiq Akmal

Nim : 1900018181

Kelas : D

Makul : PRPL

Ujian Akhir PRPL

1. Pengertian dan keuntungan :

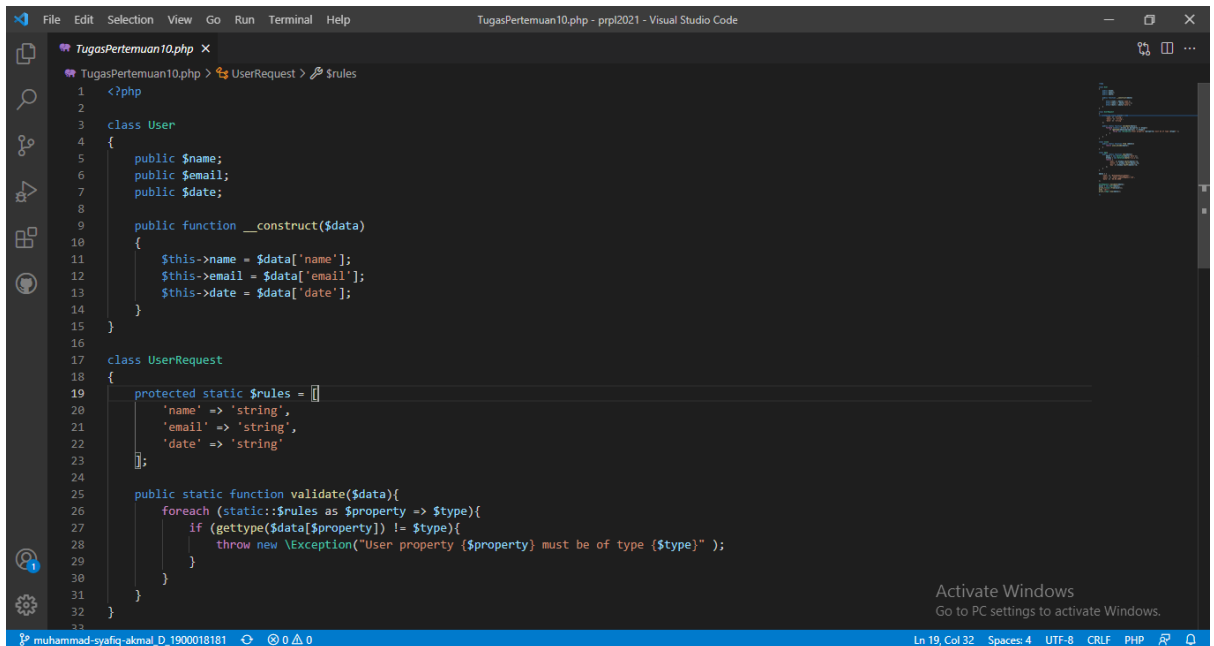
- Prinsip single responsibility

Prinsip ini memiliki arti bahwa satu class seharusnya hanya memiliki satu pekerjaan saja. Dimana di dalam satu class tersebut seharusnya hanya bertanggung jawab atas satu bagian dari fungsionalitas yang disediakan oleh perangkat lunak, dan tanggung jawab itu dienkapsulasi oleh class tersebut. Keuntungannya adalah prinsip ini membuat pengkhususan class agar class yang dimaksud bisa fokus mengerjakan tugas yang diberikan, mudah maintenance codenya.

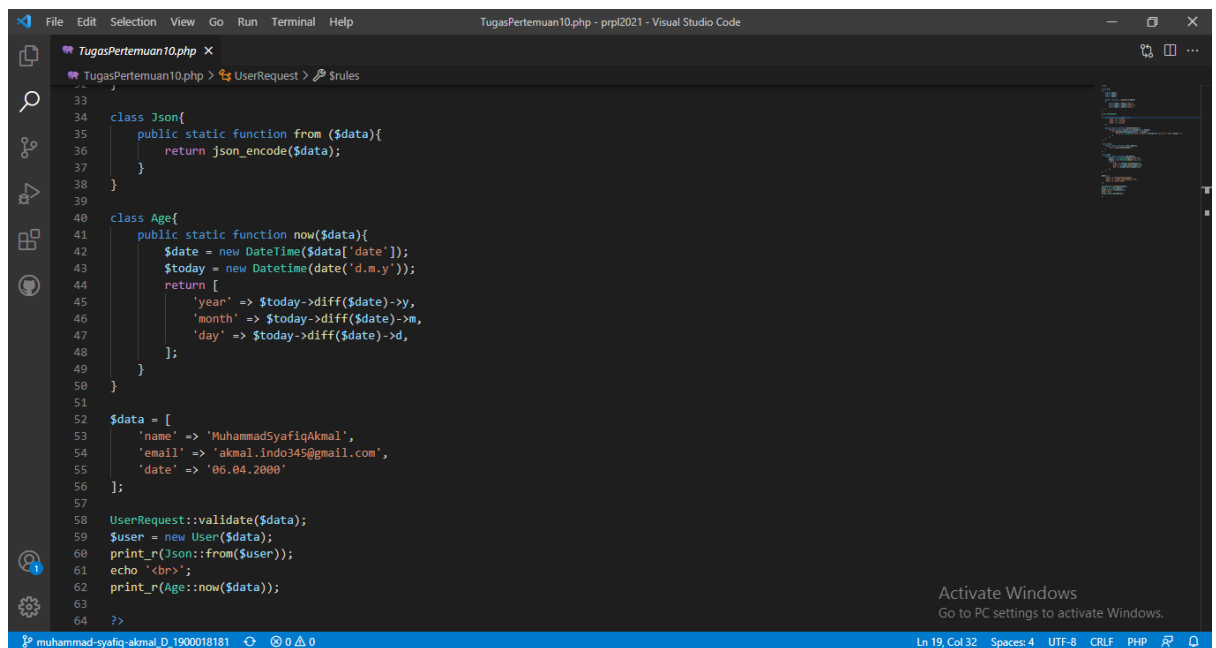
- Prinsip open/closed

Prinsip ini memiliki arti bahwa memaksa kita untuk dapat merancang sebuah entitas yang bersifat terbuka untuk di perluas, sehingga sebuah method dapat diubah dengan cara menambahkan kode program baru. Lalu untuk code yang sudah ada mungkin bisa dibuat secara tertutup atau setidaknya jarang dan tidak untuk di modifikasi. Keuntungannya adalah code bisa digunakan secara berulang.

2. Source code:

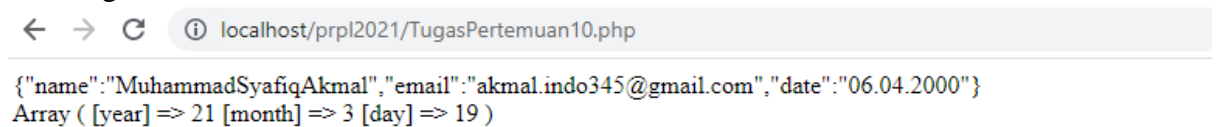


```
1 <?php
2
3 class User
4 {
5     public $name;
6     public $email;
7     public $date;
8
9     public function __construct($data)
10    {
11        $this->name = $data['name'];
12        $this->email = $data['email'];
13        $this->date = $data['date'];
14    }
15 }
16
17 class UserRequest
18 {
19     protected static $rules = [
20         'name' => 'string',
21         'email' => 'string',
22         'date' => 'string'
23     ];
24
25     public static function validate($data){
26         foreach (static::$rules as $property => $type){
27             if (gettype($data[$property]) != $type){
28                 throw new \Exception("User property {$property} must be of type {$type}");
29             }
30         }
31     }
32 }
```



```
33
34 class Json{
35     public static function from($data){
36         return json_encode($data);
37     }
38 }
39
40 class Age{
41     public static function now($data){
42         $date = new DateTime($data['date']);
43         $today = new DateTime(date('d.m.y'));
44         return [
45             'year' => $today->diff($date)->y,
46             'month' => $today->diff($date)->m,
47             'day' => $today->diff($date)->d,
48         ];
49     }
50 }
51
52 $data = [
53     'name' => 'MuhammadSyafiqAkmal',
54     'email' => 'akmal.indo345@gmail.com',
55     'date' => '06.04.2000'
56 ];
57
58 UserRequest::validate($data);
59 $user = new User($data);
60 print_r(Json::from($user));
61 echo "<br>";
62 print_r(Age::now($data));
63
64 >>
```

Running:



```
{ "name": "MuhammadSyafiqAkmal", "email": "akmal.indo345@gmail.com", "date": "06.04.2000" }
Array ( [year] => 21 [month] => 3 [day] => 19 )
```

Link Github :

https://github.com/muhammadsyafiqakmal06/prpl2021/tree/muhammad-syafiq-akmal_D_1900018181

3.

- a) Interface ialah Class yang tak punya tubuh pada methodnya. Dimana Method interface tersebut harus diterapkan pada Class turunannya. Pada Interface, pendeklarasian variabel punya atribut bersifat absolut. Sehingga membuat objek dapat menggunakan objek lainnya, namun ada sebuah syarat objek harus menerapkan method dari interface. Lalu untuk pemanggilannya bisa menggunakan interface MyInterface dan public abstract class Entitas.

b) Source code:

```
tugas12_prpl.php x
tugas12_prpl.php > Volbola > cal
1 <?php
2
3 interface Prokalkulator
4 {
5     public function cal();
6 }
7
8 class Lpersegi implements Prokalkulator
9 {
10     public $panjang;
11     public $lebar;
12     public function cal()
13     {
14         return $this->panjang * $this->lebar;
15     }
16 }
17
18 class Volbola implements Prokalkulator
19 {
20     public $jari;
21     public $phi;
22     public function cal()
23     {
24         return (4/3) * $this->phi * $this->jari * $this->jari;
25     }
26 }
27
28 class Volkerucut implements Prokalkulator
29 {
30     public $tinggi;
31     public $jari;
32     public $phi;
33     public function cal()
```

```
tugas12_prpl.php x
tugas12_prpl.php > Volbola > cal
33     public function cal()
34     {
35         return (1/3) * $this->phi * $this->jari * $this->jari * $this->jari * $this->tinggi;
36     }
37 }
38
39 class Volkubus implements Prokalkulator
40 {
41     public $rusuk;
42     public function cal()
43     {
44         return $this->rusuk * $this->rusuk * $this->rusuk;
45     }
46 }
47
48 class Kelingkaran implements Prokalkulator
49 {
50     public $jari;
51     public $phi;
52     public function cal()
53     {
54         return 2 * $this->phi * $this->jari;
55     }
56 }
57
58 class BangunRuangFactory
59 {
60     public function initializeBangunRuang($type,$satuan)
61     {
62         if ($type === 'lpersegi') {
63             $data1 = new LPersegi();
64         }
```

```
63
64
65 $data1 = new LPersegi();
66 $data1 -> panjang = $satuan['panjang'];
67 $data1 -> lebar = $satuan['lebar'];
68
69 echo "Hasil panjang : " , $satuan['panjang'], "<br>";
70 echo "Hasil lebar : " , $satuan['lebar'], "<br>";
71 echo "Hasil volume : ";
72 return $data1;
73
74 if ($type === 'vbola') {
75 $data1 = new Volbola();
76 $data1 -> jari = $satuan['jari'];
77 $data1 -> phi = $satuan['phi'];
78
79 echo "Hasil jari - jari : " , $satuan['jari'], "<br>";
80 echo "Hasil volume : ";
81 return $data1;
82
83 if ($type === 'vkerucut') {
84 $data1 = new Vkerucut();
85 $data1 -> tinggi = $satuan['tinggi'];
86 $data1 -> jari = $satuan['jari'];
87 $data1 -> phi = $satuan['phi'];
88
89 echo "Hasil jari - jari : " , $satuan['jari'], "<br>";
90 echo "Hasil tinggi : " , $satuan['tinggi'], "<br>";
91 echo "Hasil volume : ";
92 return $data1;
93
94 if ($type === 'vkubus') {
95 $data1 = new Vkubus();
96
97 return $data1;
98
99 if ($type === 'vkubus') {
100 $data1 = new Vkubus();
101 $data1 -> rusuk = $satuan['rusuk'];
102
103 echo "Hasil rusuk : " , $satuan['rusuk'], "<br>";
104 echo "Hasil volume : ";
105 return $data1;
106
107 if ($type === 'klingkaran') {
108 $data1 = new Kelingkaran();
109 $data1 -> jari = $satuan['jari'];
110 $data1 -> phi = $satuan['phi'];
111
112 echo "Hasil jari - jari : " , $satuan['jari'], "<br>";
113 echo "Hasil volume : ";
114 return $data1;
115
116 throw new Exception("ERROR!! Incorected input!");
117
118 }
119
120 }
121
122 $type = ['rusuk' => 12, 'panjang' => 0, 'lebar' => 0, 'jari' => 0, 'tinggi' => 0, 'phi' => 22/7];
123
124 $pilihan = 'vkubus';
125 $bangunRuangFactory = new BangunRuangFactory();
126 $bangunRuang = $bangunRuangFactory->initializeBangunRuang($pilihan, $type);
127 print_r($bangunRuang->cal());
128
129 ?>
```

Running:

localhost/thesis/tugas12_prpl.php

Hasil rusuk : 12
Hasil volume : 1728

Link Github:

https://github.com/muhammadsyafiqakmal06/thesis/blob/muhammadsyafiqakmal_d_1900018181/tugas12_prpl.php

4. Link:

<http://proyek-laravel-181.herokuapp.com/public/>

