

Nama : Hanny Handayani Sucinta

NIM : 1900018227

Kelas : TIF D

Matkul: PRPL

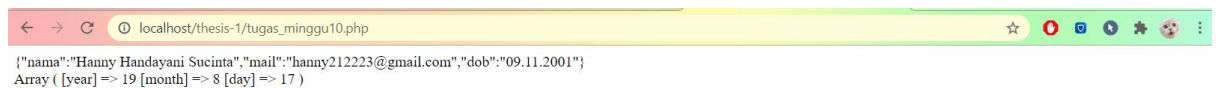
1. Jelaskan apa pengertian dan keuntungan dari penerapan prinsip berikut pada pengembangan perangkat lunak berikut :
 - Single Responsibility, yaitu sebuah prinsip dalam pemrograman komputer yang menyatakan bahwa setiap modul, kelas maupun fungsi yang ada pada program komputer harus memiliki satu tanggung jawab saja, keuntungan prinsip ini terletak pada cara penyelesaian tugasnya yang hanya terfokus pada tiap – tiap tugas yang diberikan saja.
 - Prinsip Open/Closed, yaitu sebuah prinsip pada OOP yang mengharuskan kita untuk dapat merancang sebuah entitas yang dapat dirubah tanpa harus memodifikasi source code program yang sudah ada. Keuntungan yang didapatkan dari prinsip ini adalah program yang telah kita buat dapat kita ulang pemakaiannya tanpa harus memodifikasi ulang.

2. Source code :

```
tugas_minggu10.php > PHP Intelephense > UserReq > $rules
1  <?php
2
3  class Pengguna
4  {
5      public $nama;
6      public $mail;
7      public $dob;
8
9      public function __construct($data)
10     {
11         $this->nama = $data['nama'];
12         $this->mail = $data['mail'];
13         $this->dob = $data['dob'];
14     }
15 }
16
17 class UserReq
18 {
19     protected static $rules = [
20         'nama' => 'string',
21         'mail' => 'string',
22         'dob' => 'string'
23     ];
24
25     public static function validate($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         $dob = new DateTime($data['dob']);
43         $today = new Datetime(date('d.m.y'));
44         return [
45             'year' => $today->diff($dob)->y,
46             'month' => $today->diff($dob)->m,
```

```
tugas_minggu10.php > PHP Intelephense > UserReq > $rules
34 class Json{
35     public static function from ($data){
36         return json_encode($data);
37     }
38 }
39
40 class Age{
41     public static function now($data){
42         $dob = new DateTime($data['dob']);
43         $today = new Datetime(date('d.m.y'));
44         return [
45             'year' => $today->diff($dob)->y,
46             'month' => $today->diff($dob)->m,
47             'day' => $today->diff($dob)->d,
48         ];
49     }
50 }
51
52 $data = [
53     'nama' => 'Hanny Handayani Sucinta',
54     'mail' => 'hanny212223@gmail.com',
55     'dob' => '09.11.2001'
56 ];
57
58 UserReq::validate($data);
59 $user = new Pengguna($data);
60 print_r(Json::from($user));
61 echo '<br>';
62 print_r(Age::now($data));
63 >>
```

Output :



```
{ "nama": "Hanny Handayani Sucinta", "mail": "hanny212223@gmail.com", "dob": "09.11.2001" }  
Array ( [year] => 19 [month] => 8 [day] => 17 )
```

Link Github :

https://github.com/HannySucinta/thesis/blob/hannyhsucinta_d_1900018227/tugas_minggu10.php

3. a. Jika ditambahkan Class baru, maka yg akan terjadi adalah interface tersebut akan dapat membuat objek yang baru lagi dari Class tersebut. Namun, syaratnya adalah objek tersebut harus mengimplementasikan method dari interface. Cara memanggilnya adalah :
 - interface [nama interface] { public function [namafunction()]; }
 - class [nama class] implements [nama interface] { ____; } \
- b. Source code :

```

Welcome tugas_minggu10.php tugas_minggu12.php X
tugas_minggu12.php > PHP Intelephense > Calculator
1 <?php
2
3 interface Calculator
4 {
5     public function cal();
6 }
7
8 class Lpersegi implements Calculator
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 Calculator
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 Calculator
29 {
30     public $tinggi;
31     public $jari;
32     public $phi;
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 Calculator
40 {
41     public $rusuk;
42     public function cal()
43     {
44         return $this->rusuk * $this->rusuk * $this->rusuk;
45     }
46 }

```

```

39 class Volkubus implements Calculator
40 {
41     public $rusuk;
42     public function cal()
43     {
44         return $this->rusuk * $this->rusuk * $this->rusuk;
45     }
46 }
47
48 class Kelingkaran implements Calculator
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
64             $data1 = new LPersegi();
65             $data1 -> panjang = $satuan['panjang'] ;
66             $data1 -> lebar = $satuan['lebar'] ;
67
68             echo "panjangnya adalah : " , $satuan['panjang'], "<br>";
69             echo "lebarnya adalah : " , $satuan['lebar'], "<br>";
70             echo "Hasil volumenya adalah : ";
71             return $data1;
72         }
73         if ($type === 'vbola') {
74             $data1 = new Volbola();
75             $data1 -> jari = $satuan['jari'] ;
76             $data1 -> phi = $satuan['phi'] ;
77
78             echo "Jari - jarinya adalah : " , $satuan['jari'], "<br>";
79             echo "Hasil volumenya adalah : ";
80             return $data1;
81         }
82         if ($type === 'vkerucut') {

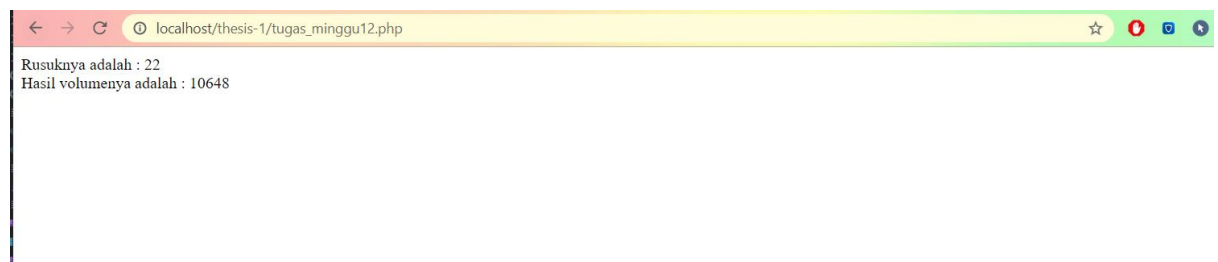
```

```

81     }
82     if ($type === 'vkerucut') {
83         $data1 = new Volkerucut();
84         $data1 -> tinggi = $satuan['tinggi'] ;
85         $data1 -> jari = $satuan['jari'] ;
86         $data1 -> phi = $satuan['phi'] ;
87
88         echo "Jari- jarinya adalah : " , $satuan['jari'], "<br>";
89         echo "tingginya adalah : " , $satuan['tinggi'], "<br>";
90         echo "Hasil volumenya adalah : ";
91         return $data1;
92     }
93     if ($type === 'vkubus') {
94         $data1 = new Volkubus();
95         $data1 -> rusuk = $satuan['rusuk'] ;
96
97         echo "Rusuknya adalah : " , $satuan['rusuk'], "<br>";
98         echo "Hasil volumenya adalah : ";
99         return $data1;
100    }
101    if ($type === 'klingkaran') {
102        $data1 = new Kelingkaran();
103        $data1 -> jari = $satuan['jari'] ;
104        $data1 -> phi = $satuan['phi'] ;
105
106        echo "Jari - jarinya adalah : " , $satuan['jari'], "<br>";
107        echo "Hasil volumenya adalah : ";
108        return $data1;
109    }
110
111    throw new Exception("Maaf, input tidak valid!");
112 }
113 }
114
115
116 $type = ['rusuk' => 22, 'panjang'=>30, 'lebar'=>21, 'jari'=>15, 'tinggi' => 30, 'phi' => 22/7];
117
118 $pilihan = 'vkubus';
119 $bangunRuangFactory = new BangunRuangFactory();
120 $bangunRuang = $bangunRuangFactory->initializeBangunRuang($pilihan,$type);
121 print_r($bangunRuang->cal());
122
123 >>

```

Output :



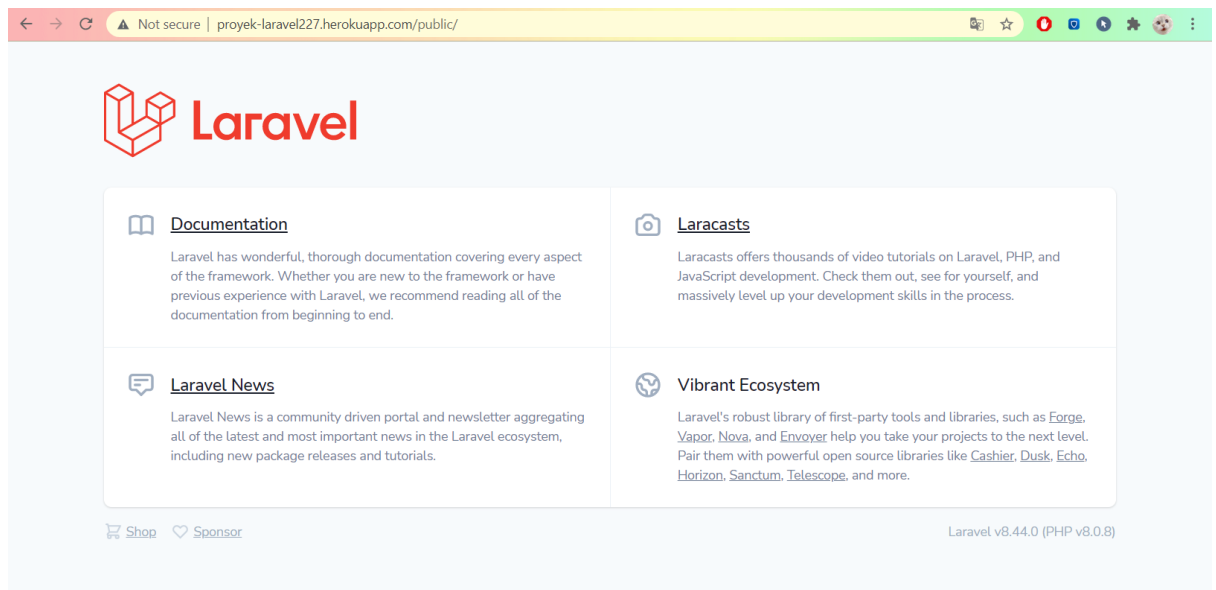
Link

Github

:

https://github.com/HannySucinta/thesis/blob/hannyhsucinta_d_1900018227/tugas_minggu12.php

4. Laravel



Link Laravel : <http://proyek-laravel227.herokuapp.com/public/>