LAPORAN PRAKTIKUM 4



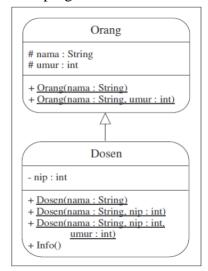
Oleh:

Alvin Febrianto

21091397031

D4 MANAJEMEN INFORMATIKA
FAKULTAS VOKASI
UNIVERSITAS NEGERI SURABAYA
TAHUN AJARAN 2022/2023

1. Buat program berdasarkan UML berikut



• Source code

```
∢▶
      no1.php
      <html lang="id">
          <link rel="stylesheet" type="text/css" href="no1.css">
          <title>Praktikum 4</title>
     </head>
     <div class="container">
     <h2>Soal 1</h2>
      <?php
          class Orang {
              protected $nama, $umur;
              public function __construct($nama,$umur){
              $this->nama = $nama;
              $this->umur = $umur;
         }
          class Dosen extends Orang {
              private $nip;
              public function __construct($nama, $umur, $nip) {
                  $this->nip = $nip;
                 parent::__construct($nama, $umur);
```

```
public function hasil() {
           return $this->nama . ' , umur ' . $this->umur. ' , NIP ' . $this->nip;
   $dosen1 = new Dosen('Alvin', 34, 22021397100);
   $dosen2 = new Dosen('Febrianto', 30, 22021397101);
   $dosen3 = new Dosen('Kurniawan', 33, 22021397102);
   echo $dosen1->hasil();
   echo $dosen2->hasil();
       echo '<br> <br>';
   echo $dosen3->hasil();
       echo '<br> <br>';
no1.css
    align-items: center;
    background-color: aliceblue;
    text-align: center;
.container {
   margin: auto;
   text-align: center;
   height: 180px;
    border-style: groove;
```

Output



Analisis

Program di atas merupakan sebuah praktikum yang menggunakan pewarisan sifat dari orang tua ke anak (orang ke dosen) yang mana semua dosen adalah orang dan setiap orang pasti memiliki identitas seperti Nama, Umur dan NIP. Ketika dosen dipanggil maka dosen akan mengambil data yang diwariskan dari orang tuanya yaitu orang.

2. Buat program berdasarkan UML berikut

```
RerataNilai
+ int average(int, int)
+ double average(double, double)
+ int average(int, int, int)
```

Source code

```
no2.php
<?php
    require_once 'ini.php';
?>
<html lang="id">
    <link rel="stylesheet" href="no2.css">
    <title>Praktikum 4</title>
</head>
<div class="container">
<h2>Nomor 2</h2>
<div class="form">
    <form action="" method="post">
            <input type="number" name="input1">
            <input type="number" name="input2">
            <input type="number" name="input3">
            <button type="submit" name="submit">Submit
          </form>
        </div>
          <?php
                if (isset($_POST['submit']))
                                                {
                        if (!$_POST['input3']) {
                            $value1 = new RerataNilai2($_POST['input1'], $_POST['input2']);
                            echo $value1->average(0);
                        } else {
                            $value1 = new RerataNilai2($_POST['input1'], $_POST['input2']);
                            echo $value1->average($_POST['input3']);
                            }
                    }
</div>
</body>
</html>
```

```
∢▶
       ini.php
      <?php
      class RerataNilai {
          protected $value, $value2;
          public function __construct($value, $value2){
              $this->value = $value;
              $this->value2 = $value2;
          public function average($value3 = 0){
              if ($value3 == 0) {
                  $str = $this->value . ' + ' . $this->value2 . ' / 2 = ';
                  return $str . ($this->value += $this->value2 ) / 2;
                  $str ='(' . $this->value . ' + ' . $this->value2 . ' + ' . $this->value2 . ') / 3 = ';
                  return $str . ($this->value += $this->value2 += $value3 ) / 3;
      class RerataNilai2 extends RerataNilai{
          public function __construct($value, $value2){
              parent::__construct($value, $value2);
          public function average($value3 = 0){
              return parent::average($value3);
```

```
no2.css x

1 .container {
2    margin: auto;
3    text-align: center;
4    width: 650px;
5    height: 150px;
6    border-style: groove;
7 }
```

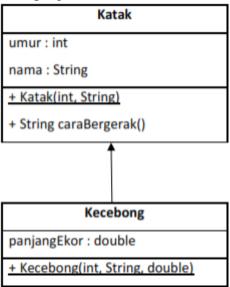
• Output

Praktikum 4 × +		- a x
← → C 🏠 🛈 thttp://localhost/praktikum4/0	☆ 🖈 🗓	
	Nomor 2 [Submit] (80 + 80 + 80) / 3 = 80	

Analisis

Program di atas merupakan pewarisan sifat dari orang tua ke anak. Tetapi dalam hal ini orang tua hanya mewariskan cara menghitung rata-ratanya saja, sedangkan yang mengeksekusi bilangannya tetap anak dengan panutan warisan orang tua tersebut. Program ini menghitung sebuah rata-rata dari 3 nilai yang diinputkan oleh user kemudian dibagi 3 untuk mendapatkan hasil rata-rata nilai tersebut.

3. Buat program berdasarkan UML berikut



Dan buat objek dengan karakteristik seperti tabel dibawah ini

Obyek	umur	nama	panjangEkor	caraBergerak
O1	5	Froggy		melompat
O2	2	Junior Frog	10	berenang

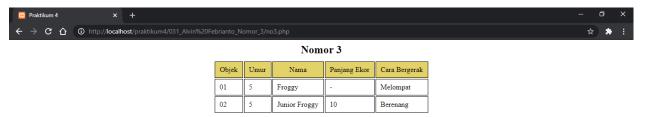
Source code

```
no3.php
<?php
  require_once 'ini.php';
<html lang="id">
  <link rel="stylesheet" type="text/css" href="no3.css">
  <title>Praktikum 4</title>
</head>
<h2><center>Nomor 3</center></h2>
0bjek
  Umur
  Nama
  Panjang Ekor
  Cara Bergerak
01
  <?=$katak->getUmur() ?> 
  <?=$katak->getnama() ?> 
  -
  <?=$katak->carabergerak() ?> 
02
  <?=$kecebong->getUmur() ?>
  <?=$kecebong->getNama() ?>
  <?=$kecebong->getpanjangekor() ?>
  <?=$kecebong->carabergerak() ?>
</body>
</html>
```

```
\blacktriangleleft
        ini.php
       <?php
       class Katak{
           protected $umur, $nama;
           public function __construct($umur, $nama){
               $this->umur = $umur;
               $this->nama = $nama;
           }
           public function getUmur(){
               return $this->umur;
           ŀ
           public function getNama(){
               return $this->nama;
 13
           }
 14
           public function caraBergerak(){
               return 'Melompat';
           }
      }
      class Kecebong extends Katak{
           private $panjangEkor;
           public function __construct($umur, $nama, $ekor){
           $this->panjangEkor = $ekor;
           parent::__construct($umur, $nama);
           public function getpanjangekor(){
               return $this->panjangEkor;
           }
           public function carabergerak(){
               return 'Berenang';
           }
       }
       $katak = new katak(5, 'Froggy');
       $kecebong = new kecebong(5, 'Junior Froggy', 10);
       ?>
```

```
no3.css
     h2{
         margin: auto;
         text-align: center;
         padding-bottom: 10px;
     }
     .table{
         margin: auto;
         font-family: sans-serif;
         border-collapse: collapse;
11
12
13
     .table, td{
         border: 1px solid;
         padding: 8px 10px;
```

• Output



Analisis

Program di atas merupakan pewarisan sifat antara orang tua dan anak. Tetapi ada beberapa hal yang diubah karena anak tidak puas dengan warisan orang tua tersebut seperti pada program di atas adalah katak dan kecebong. Katak adalah orang tua dari kecebong, tetapi ada beberapa hal dari mereka yang tidak sama seperti katak melompat dan kecebong berenang, hal tersebut menunjukkan bahwa terdapat beberapa informasi yang dipakai oleh kecebong dan ada beberapa informasi yang diganti oleh kecebong kemudian ditampilkan oleh class yang lain dengan memanggil objek dari katak maupun kecebong.