

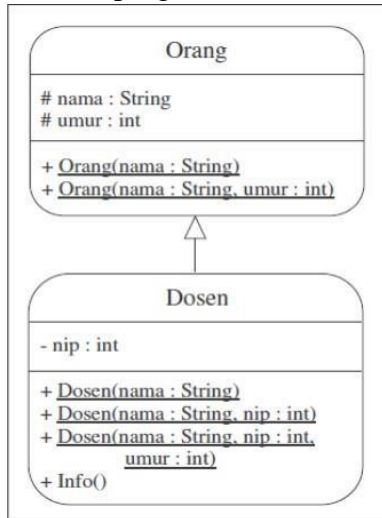
# LAPORAN PRAKTIKUM 4

Nama : Filusive Nathan Fernanda

NIM : 21091397073

Kelas : A

## 1. Buat program berdasarkan UML berikut



- Source Code

```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4 <link rel="stylesheet" type="text/css" href="nomor 1.css">
5 <title>Praktikum 4</title>
6 </head>
7 <body>
8 <div class="container">
9 <h2>Soal 1</h2>
10
11
12 <?php
13
14 class Orang {
15     protected $nama,$umur;
16     public function __construct($nama,$umur){
17         $this->nama = $nama;
18         $this->umur = $umur;
19     }
20 }
21
22 class Dosen extends Orang {
23     private $nip;
24     public function __construct($nama, $umur, $nip) {
25         $this->nip = $nip;
26         parent::__construct($nama, $umur);
27     }
28
29     public function hasil() {
30         return $this->nama . ' , umur ' . $this->umur . ' , NIP ' . $this->nip;
31     }
32 }
```

```

34 $dosen1 = new Dosen('Alvin', 34, 22021397100);
35 $dosen2 = new Dosen('Febrianto', 30, 22021397101);
36 $dosen3 = new Dosen('Kurniawan', 33, 22021397102);
37
38 echo $dosen1->hasil();
39 echo '<br> <br>';
40 echo $dosen2->hasil();
41 echo '<br> <br>';
42 echo $dosen3->hasil();
43 echo '<br> <br>';
44
45 >>
46
47 </div>
48 </body>
49 </html>

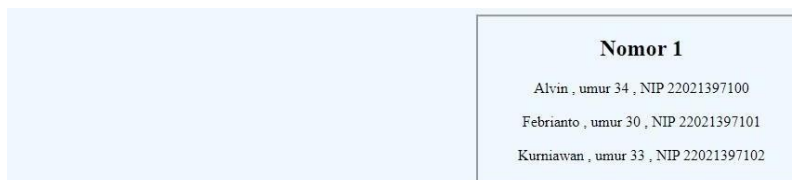
```

```

1 body {
2   align-items: center;
3   background-color: #aliceblue;
4 }
5 h2 {
6   text-align: center;
7 }
8
9 .container {
10  margin: auto;
11  text-align: center;
12  height: 180px;
13  width: 350px;
14  border-style: groove;
15 }
16

```

- 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

```

1 <?php
2     require_once 'ini.php';
3 }
4
5 <html lang="id">
6 <head>
7     <link rel="stylesheet" href="no2.css">
8     <title>Praktikum 4</title>
9 </head>
10 <body>
11     <div class="container">
12         <h2>Nomor 2</h2>
13         <div class="form">
14             <form action="" method="post">
15                 <input type="number" name="input1">
16                 <input type="number" name="input2">
17                 <input type="number" name="input3">
18                 <button type="submit" name="submit">Submit</button>
19             </form>
20         </div>
21
22         <?php
23             if (isset($_POST['submit'])) {
24                 if (isset($_POST['input3'])) {
25                     $value1 = new RerataNilai2($_POST['input1'], $_POST['input2']);
26                     echo $value1->average(0);
27                 } else {
28                     $value1 = new RerataNilai2($_POST['input1'], $_POST['input2']);
29                     echo $value1->average($_POST['input3']);
30                 }
31             }
32         </?php>
33     </div>
34 </body>
35 </html>

```

```

1 <?php
2
3 class RerataNilai {
4     protected $value, $value2;
5     public function __construct($value, $value2){
6         $this->value = $value;
7         $this->value2 = $value2;
8     }
9
10    public function average($value3 = 0){
11        if ($value3 == 0) {
12            $str = $this->value . ' + ' . $this->value2 . ' / 2 = ';
13            return $str . ($this->value + $this->value2) / 2;
14        } else {
15            $str = '(' . $this->value . ' + ' . $this->value2 . ' + ' . $this->value2 . ' ) / 3 = ';
16            return $str . ($this->value + $this->value2 + $value3) / 3;
17        }
18    }
19 }
20
21
22
23 class RerataNilai2 extends RerataNilai{
24     public function __construct($value, $value2){
25         parent::__construct($value, $value2);
26     }
27
28     public function average($value3 = 0){
29         return parent::average($value3);
30     }
31 }
32
33 >>

```

```

1 .container {
2     margin: auto;
3     text-align: center;
4     width: 650px;
5     height: 150px;
6     border-style: groove;
7 }
8 Footer
9 © 2022 GitHub, Inc.

```

- Output :

**Nomor 2**

$(80 + 80 + 80) / 3 = 80$

- 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

```
1 <?php
2     require_once 'ini.php';
3 >
4
5 <html lang="id">
6 <head>
7     <link rel="stylesheet" type="text/css" href="no3.css">
8     <title>Praktikum 4</title>
9 </head>
10 <body>
11     <h2><center>Nomor 3</center></h2>
12     <table align="center">
13
14         <tr align="center" bgcolor="e3d268">
15             <td>Objek</td>
16             <td>Umur</td>
17             <td>Nama</td>
18             <td>Panjang Ekor</td>
19             <td>Cara Bergerak</td>
20         </tr>
21
22         <tr>
23             <td>O1</td>
24             <td>= $katak-&gt;getUmur() ?></td>
25             <td>= $katak-&gt;getNama() ?></td>
26             <td></td>
27             <td>= $katak-&gt;caraBergerak() ?></td>
28         </tr>
29
30         <tr>
31             <td>O2</td>
32             <td>= $kecebong-&gt;getUmur() ?></td>
33             <td>= $kecebong-&gt;getNama() ?></td>
34             <td>= $kecebong-&gt;getpanjangekor() ?></td>
35             <td>= $kecebong-&gt;caraBergerak() ?></td>
36         </tr>
37     </table>
38 </body>
39 </html>
```

```

36 </tr>
37
38 </table>
39 </body>
40 </html>

```

```

1 <?php
2
3 class Katak{
4     protected $umur, $nama;
5     public function __construct($umur, $nama){
6         $this->umur = $umur;
7         $this->nama = $nama;
8     }
9     public function getUmur(){
10        return $this->umur;
11    }
12    public function getNama(){
13        return $this->nama;
14    }
15    public function caraBergerak(){
16        return 'Melompat';
17    }
18 }
19
20 class Kecebong extends Katak{
21     private $panjangEkor;
22     public function __construct($umur, $nama, $ekor){
23         $this->panjangEkor = $ekor;
24         parent::__construct($umur, $nama);
25     }
26     public function getpanjangEkor(){
27        return $this->panjangEkor;
28    }
29     public function carabergarak(){
30        return 'Berenang';
31    }
32 }

```

```

34 $katak = new Katak(5, 'Froggy');
35 $kecebong = new Kecebong(5, 'Junior Froggy', 10);
36
37 >>

```

```

1 h2{
2     margin: auto;
3     text-align: center;
4     padding-bottom: 10px;
5 }
6
7 .table{
8     margin: auto;
9     font-family: sans-serif;
10    border-collapse: collapse;
11 }
12
13 .table, td{
14     border: 1px solid;
15     padding: 8px 10px;
16 }

```

- Output :

Nomor 3

Objek	Umur	Nama	Panjang Ekor	Cara Bergerak
01	5	Froggy	-	Melompat
02	5	Junior Froggy	10	Berenang

- 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.