

**LAPORAN PRAKTIKUM 5 POLIMORFISME**  
**UJIAN TENGAH SEMESTER**



**Oleh:**

Fisma Meividianugraha Subani

21091397017

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

1. Suatu program terdiri dari class Pegawai sebagai parent class, class Manajer dan class Kurir sebagai subclass. Buatlah suatu program yang menerapkan konsep polymorphic argument (Polymorphic arguments adalah tipe suatu parameter yang menerima suatu nilai yang bertipe subclass-nya).

#### a) Source Code PHP

```
C:\xampp\htdocs\Praktikum 5\No 1\017_fisma_praktikum5_no1.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

017_fisma_praktikum5_no1.php x 017_fisma_praktikum5_no1.css x 017_fisma_praktikum5_no2.php x 017_fisma_praktikum5_no2.css x

1 <!-- Nama : Fisma Meividianugraha Subani
2 NIM : 21091397017 -->
3
4 <!DOCTYPE html>
5 <html lang="id">
6 <head>
7 <link rel="stylesheet" type="text/css" href="017_fisma_praktikum5_no1.css">
8 <title>Praktikum 5</title>
9 </head>
10 <body>
11 <div class="container">
12 <h2><u>Soal No.1</u></h2>
13
14 <?php
15 class Pegawai
16 {
17     public $nama;
18     public function __construct($nama)
19     {
20         $this->nama = $nama;
21     }
22     public function getName()
23     {
24         return $this->nama;
25     }
26 }
27
28 class Manajer extends Pegawai
29 {
30     public $tunjangan;
31     public function __construct($nama, $tunjangan)
32     {
33         parent::__construct($nama);
34         $this->tunjangan = $tunjangan;
35     }
36     public function getTunjangan()
37     {
38         return $this->tunjangan;
39     }
40 }
41
42 class Kurir extends Pegawai
43 {
44     public $gaji;
45
46     public function __construct($nama, $gaji)
47     {
48         parent::__construct($nama);
49         $this->gaji = $gaji;
50     }
51     public function getGaji()
52     {
53         return $this->gaji;
54     }
55 }
56
57 class Soal1
58 {
59     public static
60     function Proses($peg)
61     {
62         if ($peg instanceof Manajer)
63         {
64             $man = $peg;
65             echo "Nama Manajer: ". $man->nama, "\n";
66             echo "<br>Tunjangan: Rp. ".strval($man->tunjangan), "\n";
67         }
68         else if ($peg instanceof Kurir)
69         {
70             $kur = $peg;
71             echo "Nama Kurir: ". $kur->nama, "\n";
72             echo "<br>Gaji: Rp. ".strval($kur->gaji), "\n";
73         }
74     }
75     public static
76     function main($args)
77     {
78         $peg1 = new Manajer("Fisma Meividianugraha Subani", 5000000);
79         Soal1::Proses($peg1);
80         echo "<br>", "<br>";
81         $peg2 = new Kurir("Anton Wijaya", 1000000);
82         Soal1::Proses($peg2);
83     }
84 }
85
86 Soal1::main($args);
87
88 </div>
89 </body>
90 </html>
```

#### b) Source Code CSS

```
C:\xampp\htdocs\Praktikum 5\No 1\017_fisma_praktikum5_no1.css - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

017_fisma_praktikum5_no1.php x 017_fisma_praktikum5_no1.css x 017_fisma_praktikum5_no2.php x 017_fisma_praktikum5_no2.css x

1 body {
2     align-items: center;
3 }
4 h2{
5     text-align: center;
6 }
7 .container{
8     margin: auto;
9     padding: 15px;
10    width: 350px;
11    border-style: groove;
12 }
```

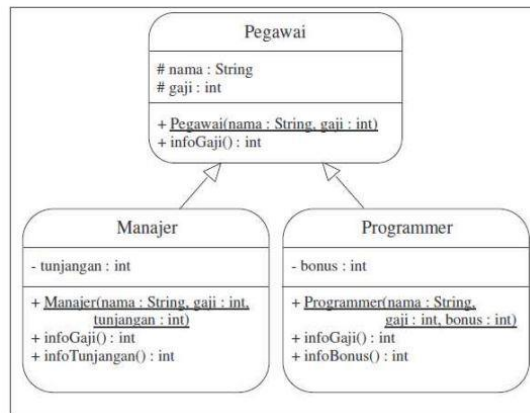
c) Output yang dihasilkan



d) Analisis

Dalam kasus diatas implementasi dari polymorphic argument berada di class info dimana method info dibuat static supaya pemanggilannya tidak perlu diinisiasi, sehingga bisa langsung dimasukkan menjadi parameter pada method info di class info.

## 2. Buat program berdasarkan UML berikut!



### a) Source Code PHP

```
1 <!-- Nama : Fisma Melvidianugraha Subani
2 NIM : 21091337017 -->
3
4 <!DOCTYPE html>
5 <html lang="id">
6 <head>
7 <link rel="stylesheet" type="text/css" href="017_fisma_praktikum5_no2.css">
8 <title>Praktikum 5</title>
9 </head>
10 <body>
11 <div class="container">
12 <h2><u>Soal No.2</u></h2>
13
14 <?php
15 class Pegawai
16 {
17     public $nama;
18     public $gaji;
19     public function __construct($nama, $gaji)
20     {
21         $this->nama = $nama;
22         $this->gaji = $gaji;
23     }
24     public function infoGaji()
25     {
26         return $this->gaji;
27     }
28 }
29
30 class Manajer extends Pegawai
31 {
32     private $tunjangan;
33     public function __construct($nama, $gaji, $tunjangan)
34     {
35         parent::__construct($nama, $gaji);
36         $this->tunjangan = $tunjangan;
37     }
38     public function infoGaji()
39     {
40         return $this->gaji;
41     }
42     public function infoTunjangan()
43     {
44         return $this->tunjangan;
45     }
46 }
47
48 class Programmer extends Pegawai
49 {
50     private $bonus;
51     public function __construct($nama, $gaji, $bonus)
52     {
53         parent::__construct($nama, $gaji);
54         $this->bonus = $bonus;
55     }
56     public function infoGaji()
57     {
58         return $this->gaji;
59     }
60     public function infoBonus()
61     {
62         return $this->bonus;
63     }
64 }
65
66 class Bayaran
67 {
68     public function hitungBayaran($peg)
69     {
70         $uang = $peg->infoGaji();
71         return $uang;
72     }
73 }
74
75 public static function main($args)
76 {
77     $man = new Manajer("Fisma Melvidianugraha Subani", 10000000, 5000000);
78     $prog = new Programmer("Bryan Bramasta", 8000000, 3000000);
79     $shr = new Bayaran();
80     echo "<br> Nama Manajer : ".$man->nama."<br> Gaji : Rp. ".strval($shr->hitungBayaran($man)). "<br>";
81     echo "<br>";
82     echo "<br> Nama Programmer : ".$prog->nama."<br> Gaji : Rp. ".strval($shr->hitungBayaran($prog)). "<br>";
83 }
84
85 }
86 Bayaran::main(array());
87
88 </div>
89 </body>
90 </html>
```

## b) Source Code CSS



```
1 body {  
2     align-items: center;  
3 }  
4 h2 {  
5     text-align: center;  
6 }  
7 .container {  
8     margin: auto;  
9     padding: 15px;  
10    width: 350px;  
11    border-style: groove;  
12 }
```

## c) Output yang dihasilkan



## d) Analisis

Program diatas adalah penerapan inheritance dengan konsep overriding, dimana pemanggilan construct di masing-masing class turunan hanya akan menginisiasi properti yang dimiliki dengan visibilitas private dan properti lain yang diturunkan akan langsung diinisiasi dengan construct dari parentnya.