

LAPORAN PRAKTIKUM 5



Oleh :

Aditya Putra Pratama

21091397043

**D4 MANAJEMEN INFORMATIKA
FAKULTAS VOKASI
UNIVERSITAS NEGERI SURABAYA
TAHUN 2022**

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).

- **Source Code :**

```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Praktikum 5</title>
8 </head>
9 <body style="align-items: center;">
10 <div class="container">
11 <h2>Nomor 1</h2>
12
13 <?php
14 class Pegawai
15 {
16     public $nama;
17     public function __construct($nama)
18     {
19         $this->nama = $nama;
20     }
21     public function getName()
22     {
23         return $this->nama;
24     }
25 }
26 class Manajer extends Pegawai
27 {
28     public $tunjangan;
29     public function __construct($nama, $tunjangan)
30     {
31         parent::__construct($nama);
32         $this->tunjangan = $tunjangan;
33     }
34     public function getTunjangan()
35     {
36         return $this->tunjangan;
37     }
38 }
39 class Kurir extends Pegawai
40 {
41     public $gaji;
42
43     public function __construct($nama, $gaji)
44     {
45         parent::__construct($nama);
46         $this->gaji = $gaji;
47     }
48     public function getGaji()
49     {
50         return $this->gaji;
51     }
52 }
53
54 class Nomor1
55 {
56     public static function Proses($peg)
57     {
58         if ($peg instanceof Manajer) {
59             $man = $peg;
60             echo "<br>Nama Manajer: " . $man->nama, "\n";
61             echo "<br>Tunjangan: Rp. " . strval($man->tunjangan), "\n";
62         } elseif ($peg instanceof Kurir) {
63             $kur = $peg;
64             echo "<br>Nama Kurir: " . $kur->nama, "\n";
65             echo "<br>Gaji: Rp. " . strval($kur->gaji), "\n";
66         }
67     }
68     public static function main($args)
69     {
70         echo "Aditya Putra Pratama - 21091397043", "\n";
```

```

71         echo "<br>", "<br>";
72         $peg1 = new Manajer("Anri", 7000000);
73         Nomor1::Proses($peg1);
74         echo "<br>", "<br>";
75         $peg2 = new Kurir("Salah", 2500000);
76         Nomor1::Proses($peg2);
77     }
78     }
79     Nomor1::main([]);
80 }
81
82 </div>
83 </body>
84 </html>

```

- **Output**

Nomor 1

Aditya Putra Pratama - 21091397043

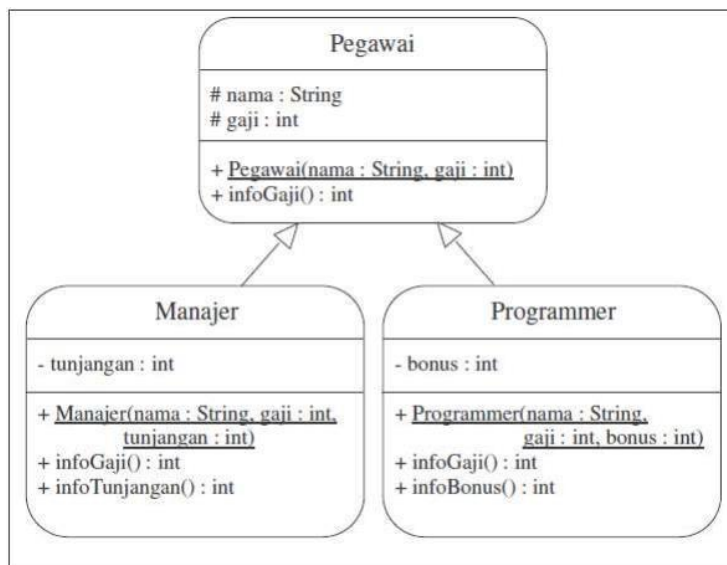
Nama Manajer: Anri
Tunjangan: Rp. 7000000

Nama Kurir: Salah
Gaji: Rp. 2500000

- **Analisis**

Pada kasus di atas, implementasi dari polymorphic argument berada di class info dimana method info dibuat static agar pemanggilannya tidak perlu diinisialisasi, sehingga bisalangsung dimasukkan menjadi parameter pada method info di class info.

2. Buat program berdasarkan UML berikut



- **Source Code**

```

1  <!DOCTYPE html>
2  <html lang="id">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Praktikum 5</title>
8  </head>
9  <body>
10     <div class="container">
11         <h2>Nomor 2</h2>
12
13         <?php
14         class Pegawai
15         {
16             public $nama;
17             public $gaji;
18             public function __construct($nama, $gaji)
19             {
20                 $this->nama = $nama;
21                 $this->gaji = $gaji;
22             }
23             public function infoGaji()
24             {
25                 return $this->gaji;
26             }
27         }
28         class Manajer extends Pegawai
29         {
30             private $tunjangan;
31             public function __construct($nama, $gaji, $tunjangan)
32             {
33                 parent::__construct($nama, $gaji);
34                 $this->tunjangan = $tunjangan;
35             }

```

```

36             public function infoGaji()
37             {
38                 return $this->gaji;
39             }
40             public function infoTunjangan()
41             {
42                 return $this->tunjangan;
43             }
44         }
45         class Programmer extends Pegawai
46         {
47             private $bonus;
48             public function __construct($nama, $gaji, $bonus)
49             {
50                 parent::__construct($nama, $gaji);
51                 $this->bonus = $bonus;
52             }
53             public function infoGaji()
54             {
55                 return $this->gaji;
56             }
57             public function infoBonus()
58             {
59                 return $this->bonus;
60             }
61         }
62         class Bayaran
63         {
64             public function hitungBayaran($peg)
65             {
66                 $uang = $peg->infoGaji();
67
68                 return $uang;
69             }
70             public static function main($args)

```

```

71     {
72         echo "Aditya Putra Pratama - 21091397043";
73         echo "<br>";
74         $man = new Manajer("Anri", 7000000, 500000);
75         echo "<br>", "<br>";
76         $prog = new Programmer("Mané", 5500000, 400000);
77         echo "<br>", "<br>";
78         $hr = new Bayaran();
79         echo "<br> Gaji untuk manajer bernama " .
80             $man->nama .
81             " adalah: Rp. " .
82             strval($hr->hitungBayaran($man)),
83             "\n";
84         echo "<br>", "<br>";
85         echo "<br>Gaji untuk programmer bernama " .
86             $prog->nama .
87             " adalah: Rp. " .
88             strval($hr->hitungBayaran($prog)),
89             "\n";
90     }
91 }
92 Bayaran::main([]);
93 >>
94 </div>
95 </body>
96 </html>

```

- **Output**

Nomor 2

Aditya Putra Pratama - 21091397043

Gaji untuk manajer bernama Anri adalah: Rp. 7000000

Gaji untuk programmer bernama Mané adalah: Rp. 5500000

- **Analisis**

Program di atas adalah penerapan inheritance dengan konsep overriding, di mana pemanggilan construct di masing-masing class turunan hanya akan menginisialisasi properti yang dimiliki dengan visibilitas private dan properti lain yang diturunkan akan langsung diinisialisasi dengan construct dari parentnya.