

**LAPORAN PRATIKUM 5**  
**Kuliah Pemograman Berorientasi Obyek**



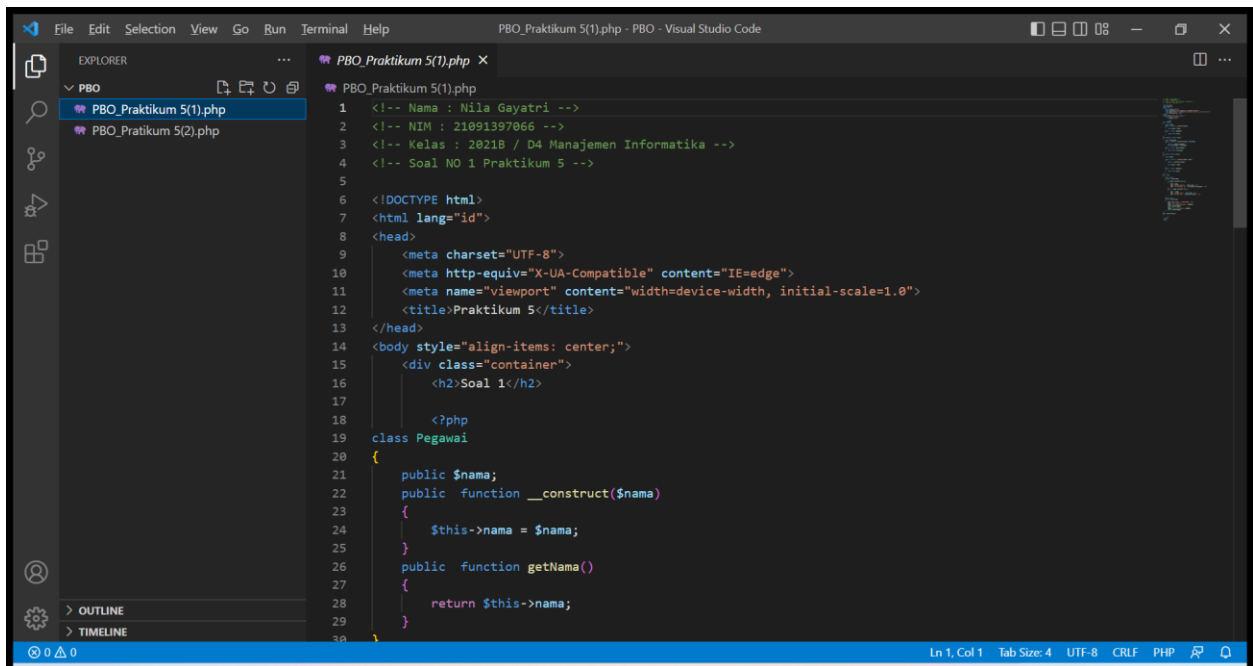
**Nama**  
**Nila Gayatri**  
**21091397066**

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA**  
**FAKULTAS VOKASI**  
**2022**

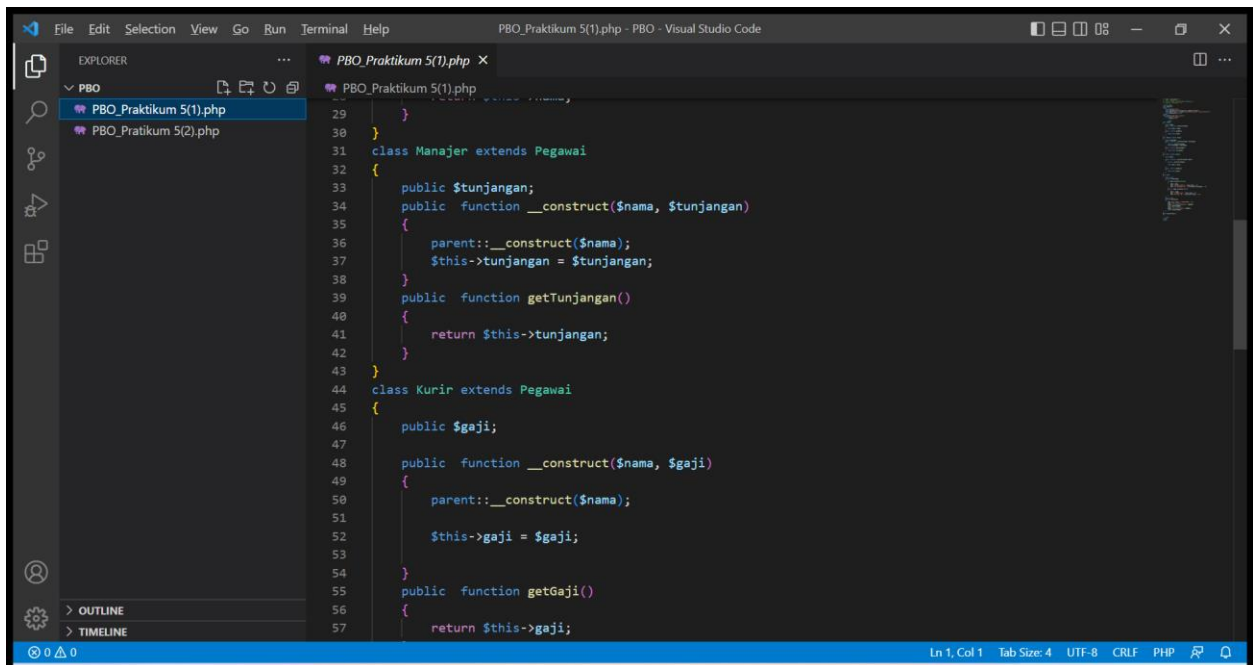
1.

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).

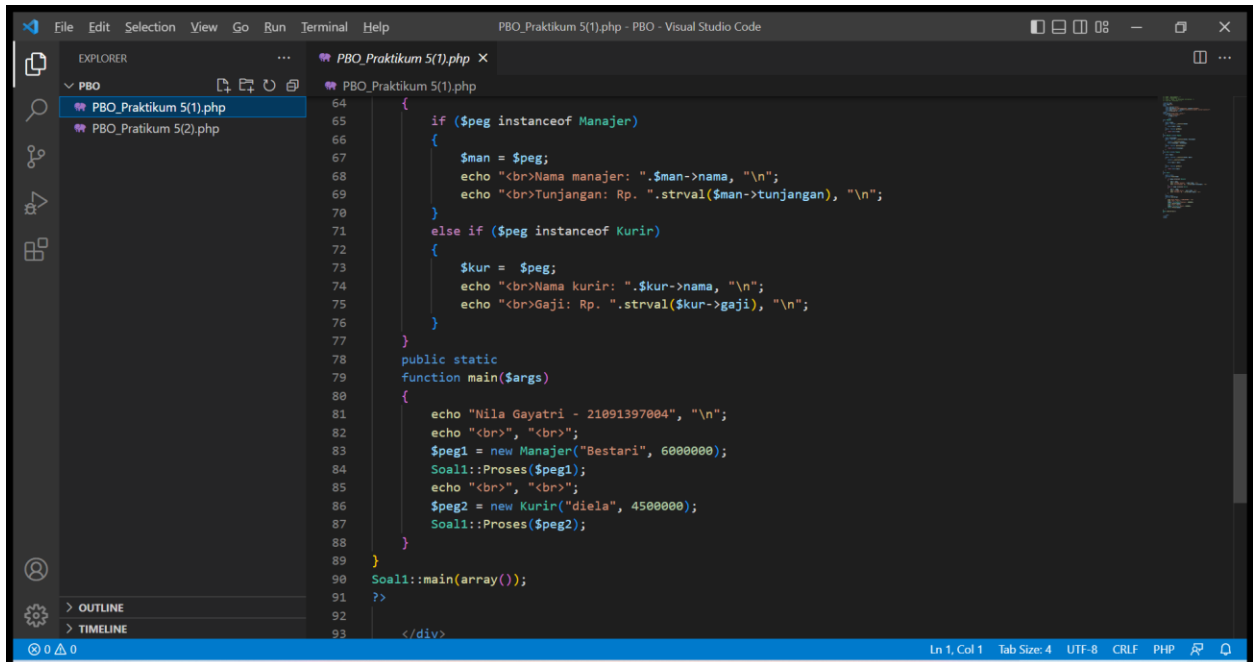
## Source Code



```
1 <!-- Nama : Nila Gayatri -->
2 <!-- NIM : 21091397066 -->
3 <!-- Kelas : 2021B / D4 Manajemen Informatika -->
4 <!-- Soal NO 1 Praktikum 5 -->
5
6 <!DOCTYPE html>
7 <html lang="id">
8 <head>
9     <meta charset="UTF-8">
10    <meta http-equiv="X-UA-Compatible" content="IE=edge">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12    <title>Praktikum 5</title>
13 </head>
14 <body style="align-items: center;">
15     <div class="container">
16         <h2>Soal 1</h2>
17     </div>
18 <?php
19 class Pegawai
20 {
21     public $nama;
22     public function __construct($nama)
23     {
24         $this->nama = $nama;
25     }
26     public function getNama()
27     {
28         return $this->nama;
29     }
30 }
```



```
29     }
30 }
31 class Manajer extends Pegawai
32 {
33     public $tunjangan;
34     public function __construct($nama, $tunjangan)
35     {
36         parent::__construct($nama);
37         $this->tunjangan = $tunjangan;
38     }
39     public function getTunjangan()
40     {
41         return $this->tunjangan;
42     }
43 }
44 class Kurir extends Pegawai
45 {
46     public $gaji;
47     public function __construct($nama, $gaji)
48     {
49         parent::__construct($nama);
50         $this->gaji = $gaji;
51     }
52     public function getGaji()
53     {
54         return $this->gaji;
55     }
56 }
57 }
```



```
64 {
65     if ($peg instanceof Manajer)
66     {
67         $man = $peg;
68         echo "<br>Nama manajer: ".$man->nama, "\n";
69         echo "<br>Tunjangan: Rp. ".strval($man->tunjangan), "\n";
70     }
71     else if ($peg instanceof Kurir)
72     {
73         $kur = $peg;
74         echo "<br>Nama kurir: ".$kur->nama, "\n";
75         echo "<br>Gaji: Rp. ".strval($kur->gaji), "\n";
76     }
77 }
78 public static
79 function main($args)
80 {
81     echo "Nila Gayatri - 21091397004", "\n";
82     echo "<br>", "<br>";
83     $peg1 = new Manajer("Bestari", 6000000);
84     Soal1::Proses($peg1);
85     echo "<br>", "<br>";
86     $peg2 = new Kurir("diela", 4500000);
87     Soal1::Proses($peg2);
88 }
89 }
90 Soal1::main(array());
91 ?>
92
93 </div>
```

Output 1.

## Soal 1

Nila Gayatri - 21091397004

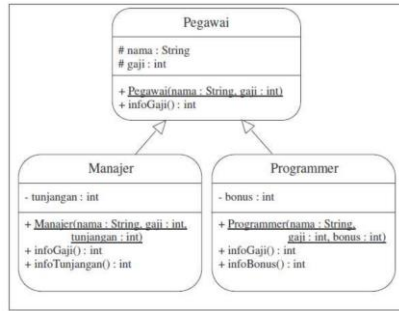
Nama manajer: Bestari  
Tunjangan: Rp. 6000000

Nama kurir: diela  
Gaji: Rp. 4500000

## Analisa

Terdapat kasus yang ada pada soale di atas dengan implementasi dari polymorphic argument dengan berada di class info dimana method info dibuat static agar pemanggilnya tidak memerlukan diinisiasi, sehingga bisa langsung dimasukkan dengan menajdi parameter pada method info di class info.

## 2. Buat program berdasarkan UML berikut



## Source code

### Php

```
1 <!-- Nama : Nila Gayatri-->
2 <!-- NIM : 21091397066 -->
3 <!-- Kelas : 2021B / D4 Manajemen Informatika -->
4 <!-- Soal NO 2 Praktikum 5 -->
5
6 <!DOCTYPE html>
7 <html lang="id">
8 <head>
9     <meta charset="UTF-8">
10    <meta http-equiv="X-UA-Compatible" content="IE=edge">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12    <title>Praktikum 5</title>
13 </head>
14 <body>
15     <div class="container">
16         <h2>Soal 2</h2>
17     </div>
18     <?php
19     class Pegawai
20     {
21         public $nama;
22         public $gaji;
23         public function __construct($nama, $gaji)
24         {
25             $this->nama = $nama;
26             $this->gaji = $gaji;
27         }
28         public function infoGaji()
29         {
30             return $this->gaji;
```

```
33 class Manajer extends Pegawai
34 {
35     private $tunjangan;
36     public function __construct($nama, $gaji, $tunjangan)
37     {
38         parent::__construct($nama, $gaji);
39         $this->tunjangan = $tunjangan;
40     }
41     public function infoGaji()
42     {
43         return $this->gaji;
44     }
45     public function infoTunjangan()
46     {
47         return $this->tunjangan;
48     }
49 }
50 class Programmer extends Pegawai
51 {
52     private $bonus;
53     public function __construct($nama, $gaji, $bonus)
54     {
55         parent::__construct($nama, $gaji);
56         $this->bonus = $bonus;
57     }
58     public function infoGaji()
59     {
60         return $this->gaji;
61     }
62 }
```

```
64     return $this->bonus;
65 }
66 }
67 class Bayaran
68 {
69     public function hitungBayaran($peg)
70     {
71         $uang = $peg->infoGaji();
72         return $uang;
73     }
74     public static function main($args)
75     {
76         echo "Nila Gayatri - 21091397066";
77         echo "<br>";
78         $man = new Manajer("Bestari", 7000000, 900000);
79         echo "<br>", "<br>";
80         $prog = new Programmer("Diana", 9000000, 800000);
81         echo "<br>", "<br>";
82         $shr = new Bayaran();
83         echo "<br> Gaji untuk Manajer Bernama ". $man->nama. " : Rp. ".strval($shr->hitungBayaran($man)) , "
84         echo "<br>", "<br>";
85         echo "<br>Gaji untuk Programmer Bernama ". $prog->nama. " : Rp. ".strval($shr->hitungBayaran($prog))
86     }
87 }
88 }
89 }
90 Bayaran::main(array());
91 ?>
92 <!-- END PHP -->
93 </div>
```

Output

## **Soal 2**

Nila Gayatri - 21091397066

Gaji untuk Manajer Bernama Bestari : Rp. 7000000

Gaji untuk Programmer Bernama Diela : Rp. 9000000

### **Analisa**

Terdapat program yang terletak pada atas ialah penerapan dari inheritance dengan konsep overriding, dimana pemanggilan construct di masing masing class turunan hanya akan menginiasi property yang dimiliki dengan visibilitas private dan property lain yang diturunkan akan langsung diinisiasi dengan construct dari parentnya.