

Pemrograman Berorientasi Objek

Praktikum 5 Polimorfisme



Disusun oleh:

Widya Sari Wibowo

(21091397070)

PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

TAHUN 2022

NOMOR 1

➤ Codigan

```
<!-- Nama: Widya Sari Wibowo -->
<!-- NIM: 21091397070 -->
<!-- Kelas: 2021 B -->

<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Praktikum 5</title>
</head>
<body style="align-items: center;">
    <div class="container">
        <h2>PBO Praktikum 5 Nomor 1</h2>

        <?php
class Pegawai
{
    public $nama;
    public function __construct($nama)
    {
        $this->nama = $nama;
    }
    public function getName()
    {
        return $this->nama;
    }
}
class Manajer extends Pegawai
{
    public $tunjangan;
    public function __construct($nama, $tunjangan)
    {
        parent::__construct($nama);
        $this->tunjangan = $tunjangan;
    }
    public function getTunjangan()
    {
        return $this->tunjangan;
    }
}
class Kurir extends Pegawai
{
    public $gaji;
```

```

        public function __construct($nama, $gaji)
        {
            parent::__construct($nama);

            $this->gaji = $gaji;

        }
        public function getGaji()
        {
            return $this->gaji;
        }
    }
}
class Soal1
{
    public static
    function Proses($peg)
    {
        if ($peg instanceof Manajer)
        {
            $man = $peg;
            echo "<br>Nama manajer: ".$man->nama, "\n";
            echo "<br>Tunjangan: Rp. ".strval($man->tunjangan), "\n";
        }
        else if ($peg instanceof Kurir)
        {
            $kur = $peg;
            echo "<br>Nama kurir: ".$kur->nama, "\n";
            echo "<br>Gaji: Rp. ".strval($kur->gaji), "\n";
        }
    }
    public static
    function main($args)
    {
        echo "Widya Sari Wibowo - 21091397070 - Kelas B", "\n";
        echo "<br>", "<br>";
        $peg1 = new Manajer("Arina", 5000000);
        Soal1::Proses($peg1);
        echo "<br>", "<br>";
        $peg2 = new Kurir("Sari", 950000);
        Soal1::Proses($peg2);
    }
}
Soal1::main(array());
?>
</div>
</body>
</html>

```

➤ Output

PBO Praktikum 5 Nomor 1

Widya Sari Wibowo - 21091397070 - Kelas B

Nama manajer: Arina
Tunjangan: Rp. 5000000

Nama kurir: Sari
Gaji: Rp. 950000

➤ Analisis

Pada kasus yang terdapat pada soal diatas implementasi dari polymorphic argument berada di class info dimana method info dibuat static agar pemanggilannya tidak memerlukan diinisiasi, sehingga bisa langsung dimasukkan menjadi parameter pada method info di class info.

NOMOR 2

➤ Codingan

```
<!-- Nama: Widya Sari Wibowo -->
<!-- NIM: 21091397070 -->
<!-- Kelas: 2021 B -->

<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Praktikum 5</title>
</head>
<body>
    <div class="container">
        <h2>PBO Praktikum 5 Nomor 2</h2>

        <?php
class Pegawai
{
    public $nama;
    public $gaji;
    public function __construct($nama, $gaji)
    {
        $this->nama = $nama;
        $this->gaji = $gaji;
    }
}
```

```

        public function infoGaji()
        {
            return $this->gaji;
        }
    }
class Manajer extends Pegawai
{
    private $tunjangan;
    public function __construct($nama, $gaji, $tunjangan)
    {
        parent::__construct($nama, $gaji);
        $this->tunjangan = $tunjangan;
    }
    public function infoGaji()
    {
        return $this->gaji;
    }
    public function infoTunjangan()
    {
        return $this->tunjangan;
    }
}
class Programmer extends Pegawai
{
    private $bonus;
    public function __construct($nama, $gaji, $bonus)
    {
        parent::__construct($nama, $gaji);
        $this->bonus = $bonus;
    }
    public function infoGaji()
    {
        return $this->gaji;
    }
    public function infoBonus()
    {
        return $this->bonus;
    }
}
class Bayaran
{
    public function hitungBayaran($peg)
    {
        $uang = $peg->infoGaji();

        return $uang;
    }
    public static function main($args)

```

```

{
    echo "Widya Sari Wibowo - 21091397070 - Kelas B";
    echo "<br>";
    $man = new Manajer("Arina", 1000000, 1000000);
    echo "<br>", "<br>";
    $prog = new Programmer("Widya", 600000, 600000);
    echo "<br>", "<br>";
    $hr = new Bayaran();
    echo "<br> Gaji Manajer, Nama: ". $man->nama. " = Rp. ".strval($hr-
>hitungBayaran($man)) , "\n";
    echo "<br>", "<br>";
    echo "<br>Gaji Programmer, Nama: ".$prog->nama. " = Rp. ".strval($hr-
>hitungBayaran($prog)) , "\n";

}
}
Bayaran::main(array());
?>
<!-- END PHP -->
</div>
</body>
</html>

```

➤ Output

PBO Praktikum 5 Nomor 2

Widya Sari Wibowo - 21091397070 - Kelas B

Gaji Manajer, Nama: Arina = Rp. 1000000

Gaji Programmer, Nama: Widya = Rp. 600000

➤ 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.