LAPORAN PRAKTIKUM 5

Object Oriented Programming

Oleh:

Rendi Nicolas Mahendra
21091397071

PROGRAM STUDI D4 MANAJEMEN

INFORMATIKA FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

1. Source Code

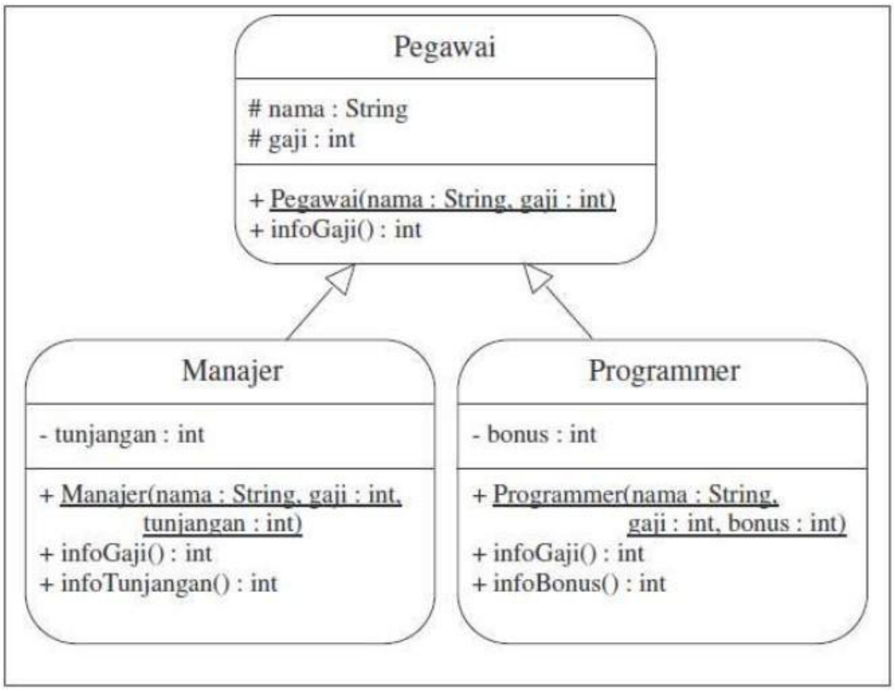
polymorphic_argument.php

```php
<?php
class Pegawai
{
    public $name;
    public function __construct($name)
    {
        $this->name = $name;
    }
    public function getName()
    {
        return $this->name;

    }
}

class Manager extends Pegawai
{
    public $tunjangan;
    public function __construct($name, $tunjangan)
    {
        parent::__construct($name);
        $this->tunjangan = $tunjangan;
    }
    public function getTunjangan()
    {
        return $this->tunjangan;
    }
}

class Kurir extends Pegawai
{
    public $gaji;
    public function __construct($name, $gaji)
    {
        parent::__construct($name);
        $this->gaji = $gaji;
    }
    public function getGaji()
    {
        return $this->gaji;
    }
}

class SoalNo1
{
    public static
    function Proses($peg)
    {
        if ($peg instanceof Manager)
        {
            $man = $peg;
            echo "<br>Nama Manager: ".$man->name, "\n";
            echo "<br>Tunjangan: RP. ".strval($man->tunjangan),
"\n";
        }
        else if ($peg instanceof Kurir)
        {
            $kur = $peg;
            echo "<br>Nama Kurir: ".$kur->name, "\n";
            echo "<br>Gaji= RP. ".strval($kur->gaji), "\n";
        }
    }
    public static
    function main($args)
    {
        echo "21091397071 Rendi Nicolas Mahendra", "\n";
        echo "<br>", "<br>";
        $peg1 = new Manager("Rendi", 20000000);
        SoalNo1::Proses($peg1);
        echo "<br>", "<br>";
        $peg2 = new Kurir("Mahendra", 15000000);
        SoalNo1::Proses($peg2);
    }
}
SoalNo1::main(array());
?>
```

Analisa

Dalam implementasi dari polymorphic argument berada di class dimana method 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



Source Code

uml.php

```php
<?php

class Pegawai
{
    public $name;
    public $gaji;
    public function __construct($name, $gaji)
    {
        $this->name = $name;
        $this->gaji = $gaji;
    }
    public function infoGaji()
    {
        return $this->gaji;
    }
}

class Manager extends Pegawai
{
    private $tunjangan;
    public function __construct($name, $gaji, $tunjangan)
    {
        parent::__construct($name, $gaji);
        $this->tunjangan = $tunjangan;
    }
    public function infoGaji()
    {
        return $this->gaji;
    }
    public function infoTunjangan()
    {
        return $this->tunjangan;
    }
}

class Programmer extends Pegawai
{
    private $bonus;
    public function __construct($name, $gaji, $bonus)
    {
        parent::__construct($name, $gaji);
        $this->bonus = $bonus;
    }
    public function infoBonus()
    {
        return $this->bonus;
    }
}

class Bayaran
{
    public function hitungBayaran($peg)
    {
        $uang = $peg->infoGaji();

        return $uang;
    }
    public static function main($args)
    {
        echo "Rendi Nicolas Mahendra", "\n";
        echo "21091397071", "\n";
        echo "<br>", "<br>";
        $man = new Manager("Rendi", 20000000, 45);
        $prog = new Programmer("Bariq", 18000000, 30);
        $hr = new Bayaran();
        echo "<br>Gaji Manager ". $man->name." : RP.
".strval($hr->hitungBayaran($man)), "\n";
        echo "<br>Gaji Programmer ". $prog->name." : RP.
".strval($hr->hitungBayaran($prog)), "\n";
```

```
    }
}

Bayaran::main(array());
?>
```

Analisa

Program diatas adalah penerapan inheritance dengan konsep overriding yang terdapat parent dan child di dalam nya, dimana pemanggilan constructor 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.