Tugas Pemrograman Berorientasi Objek

Praktikum 6 (Abstract dan Interface)

Disusun oleh :

Muhammad Yuda Pratama (21091397025)

Program Studi D4 Manajemen Informatika

Fakultas Vokasi

Universitas Negeri Surabaya

2022

1. Source Code :

```php
<!--- Muhammad Yuda Pratama - 21091397025 --->

<?php
    require_once 'no1.php';
?>

<!DOCTYPE html>
<html lang="id">

<head>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <title>PBO - Praktikum 6</title>
</head>

<body>
    <div class="container">
        <br>
        <h2 class="text-center">PBO - Praktikum 6</h2>
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <h4 class="text-center"><strong>Soal Nomor 1</strong></h4>
                <br><br>
                <b><?= $truck->getMaxLoad() . ' kg'; ?> <br></b>
                <br>
                <?= $truck->addBox(3000) . ' kg'; ?> <br>
                <?= $truck->addBox(1000) . ' kg'; ?> <br>
                <?= $truck->addBox(6000) . ' kg'; ?> <br>

                    <?php
                        echo "Jadi, Butuh Bahan Bakar sebanyak " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
                    ?>
                <br>
                ------------------------------------------------
                <br>
                <br>
                <b><?= $riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
                <br>
                <?= $riverBarge->addBox(1000) . ' kg'; ?> <br>
                <?= $riverBarge->addBox(6000) . ' kg'; ?> <br>
                <?= $riverBarge->addBox(8000) . ' kg'; ?> <br>

                    <?php
                        echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge->calcFuelNeeds() . ' Liter';
                    ?>
            </div>
        </div>
    </div>
</body>
</html>
```

```php
<!--- Muhammad Yuda Pratama - 21091397025 --->

<?php

require_once '../No1/abstract.php';

class Truck extends Vehicle {
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
}
```

```php
class RiverBarge extends Vehicle {
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
}
$truck = new Truck(10000, 'Truk');
$riverBarge = new RiverBarge(15000, 'Perahu');
```

```php
<!--- Muhammad Yuda Pratama - 21091397025 --->

<?php

abstract class Vehicle {
    private $load = 0;
    protected $maxLoad = 0, $name;

    protected function __construct($maxLoad, $name) {
        $this->$maxLoad = $maxLoad;
        $this->$name = $name;
    }

    public function getLoad() {
        return $this->load;
    }

    public function getMaxLoad() {
        echo 'Maksimal muatan ' . $this->name . ' ' ;
        return $this->maxLoad;
    }
```

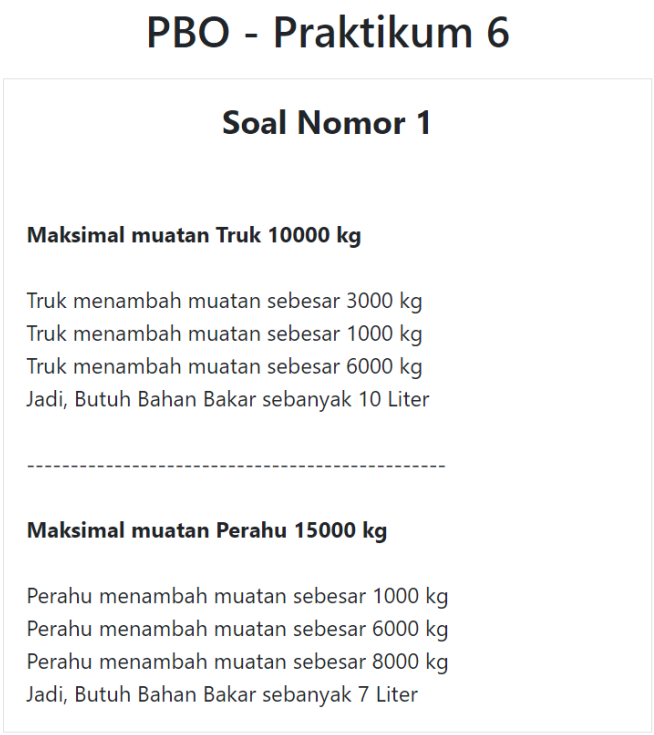
```php
    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name menambah muatan sebesar $weight <br>";
            echo 'Muatan telah penuh tidak bisa menambah lagi';
        }else {
            $this->load += $weight;
            echo "$this->name menambah muatan sebesar $weight";
        }
    }

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 50000000;
        $range /= $this->load;
        return $range;
    }

    protected function calcTripDistance() {
        return 500;
    }
}
```

Output :



Analisis :

Implementasi dari abstract class pada class Vehicle, method calcFuelNeeds digunakan untuk menghitung bahan bakar yang digunakan. Abstract method di letakkan pada class Vehicle sebagai parent class dan diakses oleh child classnya yaitu class Truk, dan class RiverBarge yang akan mengembalikan nilai yang dihasilkan dari pembagian 2 method yaitu calcFuelEfficiency dan calcTripDistance.

2. Source Code :

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php
    require_once 'No2.php';
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <title>PBO - Praktikum 6</title>
</head>

<body>
    <div class="container">
        <br>
        <h2 class="text-center">PBO - Praktikum 6</h2>
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <h4 class="text-center"><strong>Soal Nomor 2</strong></h4>
                <br><br>
                <b><?php
                        echo "Superman";
                    ?></b> <br>
                    <?= $superman->land(); ?> <br>
                    <?= $superman->takeOff(); ?> <br>
                    <?= $superman->fly(); ?> <br>
                    <?= $superman->leapBuilding(); ?> <br>
                    <?= $superman->stopBullet(); ?> <br>
                    <br>
                    <b><?php
                            echo "Bird";
                        ?></b> <br>
                    <?= $bird->buildNest(); ?> <br>
                    <?= $bird->takeOff(); ?> <br>
                    <?= $bird->fly(); ?> <br>
                    <?= $bird->land(); ?> <br>
                    <?= $bird->layEggs(); ?> <br>
                    <br>
                    <b><?php
                            echo "Airplane";
                        ?></b> <br>
                    <?= $airplane->takeOff(); ?> <br>
                    <?= $airplane->fly(); ?> <br>
                    <?= $airplane->land(); ?> <br>
                </div>
            </div>
        </div>
    </body>

    </html>
```

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->


<?php

require_once '../No2/interface.php';

class Airplane implements Flyer {
    public function takeOff() {
        return 'Pesawat lepas landas..';
    }

    public function land() {
        return 'Pesawat mendarat';
    }

    public function fly() {
        return 'Pesawat dalam perjalanan';
    }
}
```

```php
class Bird implements Flyer {
    public function takeOff() {
        return 'Burung mencari makan';
    }

    public function land() {
        return 'Burung kembali pulang';
    }

    public function fly() {
        return 'Burung terbang';
    }

    public function buildNest() {
        return 'Burung membuat sarang';
    }

    public function layEggs() {
        return 'Burung bertelur';
    }
}
```

```php
class Superman implements Flyer {
    public function takeOff() {
        return 'Superman mengejar Batman';
    }

    public function land() {
        return 'Superman melawan Batman';
    }

    public function fly() {
        return 'Superman melancarkan pukulan';
    }

    public function leapBuilding() {
        return 'Batman terpental menabrak bangunan pencakar langit';
    }

    public function stopBullet() {
        return 'Polisi menembaki superman namun ditangkis';
    }
}

$airplane = new Airplane;
$bird = new Bird;
$superman = new Superman;
```

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php

interface Flyer {
    public function takeOff();
    public function land();
    public function fly();
}

interface Sailer {
    public function dock();
    public function cruise();
}
```

Output :

PBO - Praktikum 6

Soal Nomor 2

Superman
Superman melawan Batman
Superman mengejar Batman
Superman melancarkan pukulan
Batman terpental menabrak bangunan pencakar langit
Polisi menembaki superman namun ditangkis

Bird
Burung membuat sarang
Burung mencari makan
Burung terbang
Burung kembali pulang
Burung bertelur

Airplane
Pesawat lepas landas..
Pesawat dalam perjalanan
Pesawat mendarat

Analisis :

Implementasi Polymhorpism dengan penggunaan Interface Flyer sehingga semua class yang Implements dari interface Fyler harus memiliki method takeoff, land, dan fly.

3. Source Code

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php
    require_once 'No3.php';
?>

<!DOCTYPE html>
<html lang="id">

<head>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <title>PBO - Praktikum 6</title>
</head>
<body>
    <div class="container">
        <h2 class="text-center">PBO - Praktikum 6</h2>
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <h4 class="text-center"><strong>Soal Nomor 3</strong></h4>
                <br><br>
                <?= $singa->eat(); ?> <br>
                <?= $manusia->eat(); ?> <br>
                <br>
                <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $airplane2->addBox(5000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(7000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(3000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(4000) . ' kg'; ?> <br>
                <?= $airplane2->takeOff(); ?> <br>
                <?= $airplane2->fly(); ?> <br>
                <?= $airplane2->land(); ?> <br>

                <?php
                    echo "Jadi, Butuh Bahan Bakar sebanyak " . $airplane2->calcFuelNeeds() . ' Liter'. '<br>';
                ?>
                <br>
                <?= $superman2->eat(); ?> <br>
                <?= $superman2->land(); ?> <br>
                <?= $superman2->takeOff(); ?> <br>
                <?= $superman2->fly(); ?> <br>
                <?= $superman2->leapBuilding(); ?> <br>
                <?= $superman2->stopBullet(); ?> <br>
            </div>
        </div>
    </div>
</body>

</html>
```

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php

abstract class Vehicle {
    private $load = 0;
    protected $maxLoad = 0, $name;

    protected function __construct($maxLoad, $name) {
        $this->$maxLoad = $maxLoad;
        $this->$name = $name;
    }

    public function getLoad() {
        return $this->load;
    }

    public function getMaxLoad() {
        echo 'Maksimal muatan ' . $this->name . ' ' ;
        return $this->maxLoad;
    }
```

```php
    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name menambah muatan sebesar $weight <br>";
            echo 'Muatan telah penuh tidak bisa menambah lagi';
        }else {
            $this->load += $weight;
            echo "$this->name menambah muatan sebesar $weight";
        }
    }

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 50000000;
        $range /= $this->load;
        return $range;
    }

    protected function calcTripDistance() {
        return 500;
    }
}
```

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php

interface Flyer {
    public function takeOff();
    public function land();
    public function fly();
}

interface Sailer {
    public function dock();
    public function cruise();
}
```

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php

require_once '../No3/abstract.php';
require_once '../No3/interface.php';

class Animal
{
    protected $name;

    public function __construct($name)
    {
        $this->name = $name;
    }

    public function eat()
    {
        return $this->name . ' sedang makan';
    }
}
```

```php
class Homosapiens extends Animal {}

class Airplane2 extends Vehicle implements Flyer
{
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function takeOff()
    {
        return "$this->name lepas landas";
    }

    public function land()
    {
        return "$this->name mendarat";
    }

    public function fly()
    {
        return "$this->name dalam perjalanan";
    }
```

```php
    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();



        return ceil($fuel /= $trip);
    }
}

class Superman2 extends Homosapiens implements Flyer
{
    public function takeOff()
    {
        return "$this->name mengejar Batman";
    }

    public function land()
    {
        return "$this->name melawan Batman";
    }

    public function fly()
    {
        return "$this->name melancarkan pukulan";
    }
```

```php
    public function leapBuilding()
    {
        return "Batman terpental menabrak bangunan pencakar langit";
    }

    public function stopBullet()
    {
        return "Polisi menembaki $this->name namun ditangkis";
    }
}

$singa = new Animal('Singa');
$manusia = new Homosapiens('Andi');
$airplane2 = new Airplane2(20000, 'Garuda Air');
$superman2 = new Superman2('Superman');
```

Output :

PBO - Praktikum 6

Soal Nomor 3

Singa sedang makan
Andi sedang makan

Maksimal muatan Garuda Air 20000 kg
Garuda Air menambah muatan sebesar 5000 kg
Garuda Air menambah muatan sebesar 7000 kg
Garuda Air menambah muatan sebesar 3000 kg
Garuda Air menambah muatan sebesar 4000 kg
Garuda Air lepas landas
Garuda Air dalam perjalanan
Garuda Air mendarat
Jadi, Butuh Bahan Bakar sebanyak 6 Liter

Superman sedang makan
Superman melawan Batman
Superman mengejar Batman
Superman melancarkan pukulan
Batman terpental menabrak bangunan pencakar langit
Polisi menembaki Superman namun ditangkis

Analisa :

Terdapat interface Flyer dan abtract class Vehicle. Class airplane implementasi dari interface Flyer dan turunan dari Vehicle sehingga class Airplane harus memiliki method calcFuelNeeds, takeoff, land, dan fly. Class Bird implementasi dari Flyer dan turunan dari Animal sehingga memiliki method takeoff, land, fly, dan eat. Class Superman turunan dari homosapiens yang juga turunan dari Animal, serta implementasi dari interface Flyer. Maka class Superman memiliki method eat, takeoff, land, fly.

4. Source Code

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php
    require_once 'no4.php';
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <title>PBO - Praktikum 6</title>
</head>
<body>
        <h2 class="text-center">PBO - Praktikum 6</h2>
        <div class="container">
  <div class="row">
  <h4 class="text-center"><strong>Soal Nomor 4</strong></h4>
  <br><br>
    <div class="col-4 mx-auto border p-2 mt-2">
                <b><?= $riverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $riverBarge2->addBox(12000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(14000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(1000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(3000) . ' kg'; ?> <br>
```

```php
                <?= $riverBarge2->dock(); ?> <br>
                <?= $riverBarge2->cruise(); ?> <br>
                <?php
                    echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
                ?>
    </div>
    <div class="col-4 mx-auto border p-2 mt-2">
                <b><?= $seaPlane->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $seaPlane->addBox(12000) . ' kg'; ?> <br>
                <?= $seaPlane->addBox(8000) . ' kg'; ?> <br>
                <?= $seaPlane->dock(); ?> <br>
                <?= $seaPlane->cruise(); ?> <br>
                <?= $seaPlane->takeOff(); ?> <br>
                <?= $seaPlane->fly(); ?> <br>
                <?= $seaPlane->land(); ?> <br>
                <?php
                    echo "Jadi, Butuh Bahan Bakar sebanyak " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
                ?>
```

```
    </div>
    <div class="col mx-auto border p-2 mt-2">
                <b><?= $helicopter->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $helicopter->addBox(8000) . ' kg'; ?> <br>
                <?= $helicopter->addBox(2000) . ' kg'; ?> <br>
                <?= $helicopter->takeOff(); ?> <br>
                <?= $helicopter->fly(); ?> <br>
                <?= $helicopter->land(); ?> <br>
                <?php
                    echo "Jadi, Butuh Bahan Bakar sebanyak " . $helicopter->calcFuelNeeds() . ' Liter'. '<br>';
                ?>
    </div>
  </div>
</div>
</body>

</html>
```

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php

require_once '../No4/abstract.php';
require_once '../No4/interface.php';

class RiverBarge2 extends Vehicle implements Sailer {
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }

    public function dock() {
        return $this->name . ' berada di dermaga';
    }

    public function cruise() {
        return $this->name . ' sedang berlayar';
    }
}
```

```php
class Airplane2 implements Flyer {
    public function takeOff() {
        return 'Pesawat lepas landas';
    }
    public function land() {
        return 'Pesawat mendarat';
    }
    public function fly() {
        return 'Pesawat dalam perjalanan';
    }
}

class SeaPlane extends Vehicle implements Sailer {
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }

    public function dock() {
        return $this->name . ' berada di dermaga';
    }
```

```php
    public function cruise() {
        return $this->name . ' sedang berlayar';
    }

    public function takeOff() {
        return $this->name . ' lepas landas';
    }

    public function land() {
        return $this->name . ' mendarat';
    }

    public function fly() {
        return $this->name . ' dalam perjalanan';
    }
}

class Helicopter extends Vehicle {
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
```

```php
    public function takeOff() {
        return $this->name . ' lepas landas';
    }

    public function land() {
        return $this->name . ' mendarat';
    }

    public function fly() {
        return $this->name . ' dalam perjalanan';
    }
}

$riverBarge2 = new RiverBarge2(30000, 'Atomic');
$seaPlane = new SeaPlane(20000, 'Titanic');
$helicopter = new Helicopter(10000, 'Brocklyn');
```

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php

interface Flyer {
    public function takeOff();
    public function land();
    public function fly();
}

interface Sailer {
    public function dock();
    public function cruise();
}
```

```php
<!-- Muhammad Yuda Pratama - 21091397025 -->

<?php

abstract class Vehicle {
    private $load = 0;
    protected $maxLoad = 0, $name;

    protected function __construct($maxLoad, $name) {
        $this->$maxLoad = $maxLoad;
        $this->$name = $name;
    }

    public function getLoad() {
        return $this->load;
    }

    public function getMaxLoad() {
        echo 'Maksimal muatan ' . $this->name . ' ' ;
        return $this->maxLoad;
    }
```

```php
    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name menambah muatan sebesar $weight <br>";
            echo 'Muatan telah penuh tidak bisa menambah lagi';
        }else {
            $this->load += $weight;
            echo "$this->name menambah muatan sebesar $weight";
        }
    }

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 50000000;
        $range /= $this->load;
        return $range;
    }

    protected function calcTripDistance() {
        return 500;
    }
}
```

Output :

PBO - Praktikum 6

Soal Nomor 4

Maksimal muatan Atomic 30000 kg	**Maksimal muatan Titanic 20000 kg**	**Maksimal muatan Brocklyn 10000 kg**
Atomic menambah muatan sebesar 12000 kg	Titanic menambah muatan sebesar 12000 kg	Brocklyn menambah muatan sebesar 8000 kg
Atomic menambah muatan sebesar 14000 kg	Titanic menambah muatan sebesar 8000 kg	Brocklyn menambah muatan sebesar 2000 kg
Atomic menambah muatan sebesar 1000 kg	Titanic berada di dermaga	Brocklyn lepas landas
Atomic menambah muatan sebesar 3000 kg	Titanic sedang berlayar	Brocklyn dalam perjalanan
Atomic berada di dermaga	Titanic lepas landas	Brocklyn mendarat
Atomic sedang berlayar	Titanic dalam perjalanan	Jadi, Butuh Bahan Bakar sebanyak 10 Liter
Jadi, Butuh Bahan Bakar sebanyak 4 Liter	Titanic mendarat	
	Jadi, Butuh Bahan Bakar sebanyak 5 Liter	

Analisa :

Implementasi polymhorphism dengan interface dan abstact class detinjukkan pada cals SeaPlane yang implements interface Sailer, turunan dari class Airplane yang implements Flyer dan child dari Vehicle sehingga class SeaPlane memiliki method dock, cruise, takeoff, land, fly, dan calcFuelNeeds.