LAPORAN PRAKTIKUM 6

Pemograman Berorientasi Objek

Disusun Oleh :

Affandika Febrian Putra Yunanto (21091397030)

PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

2022

1. Source code

PHP

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php

require_once 'Prak 6 Nomor 1 abstrak.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);
    }
}

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(20000, 'Truk');
$riverBarge = new RiverBarge(17000, 'Perahu');
```

Abstract

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?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;
    }

    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name ditambahkan muatan sebesar $weight <br>";
    }

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

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 60000000;
        $range /= $this->load;
        return $range;
    }

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

Index

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php
    require_once 'Prak 6 Nomor 1.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>Praktikum 6</title>
</head>

<body>
    <div class="container">
        <br>
        <h2 class="text-center">Praktikum 6 PBO</h2>
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <h4 class="text-center"><strong>Nomor 1</strong></h4>
                <br><br>
```

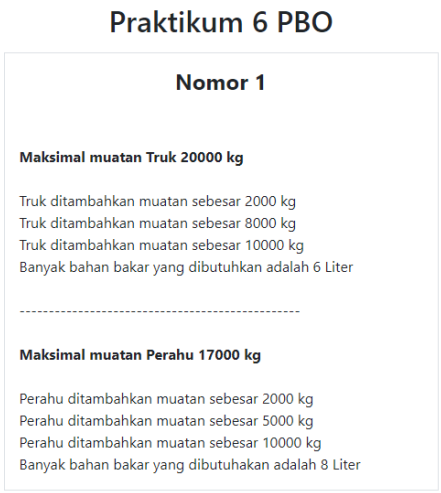
Go Run Terminal Help — Prak 6 Nomor 1 index.php - tugas pbo - Visual Studio Code

Prak 6 Nomor 1.php | Prak 6 Nomor 1 abstrak.php | Prak 6 Nomor 1 index.php

Prak 6 Nomor 1 index.php > ...

```php
                <b><?= $truck->getMaxLoad() . ' kg'; ?> <br></b>
                <br>
                <?= $truck->addBox(2000) . ' kg'; ?> <br>
                <?= $truck->addBox(8000) . ' kg'; ?> <br>
                <?= $truck->addBox(10000) . ' kg'; ?> <br>

                    <?php
                        echo "Banyak bahan bakar yang dibutuhkan adalah " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
                    ?>
                <br>
                ---------------------------------------------
                <br>
                <br>
                <b><?= $riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
                <br>
                <?= $riverBarge->addBox(2000) . ' kg'; ?> <br>
                <?= $riverBarge->addBox(5000) . ' kg'; ?> <br>
                <?= $riverBarge->addBox(10000) . ' kg'; ?> <br>

                    <?php
                        echo "Banyak bahan bakar yang dibutuhakan adalah " . $riverBarge->calcFuelNeeds() . ' Liter';
                    ?>
            </div>
        </div>
    </div>
</body>
```

Output:



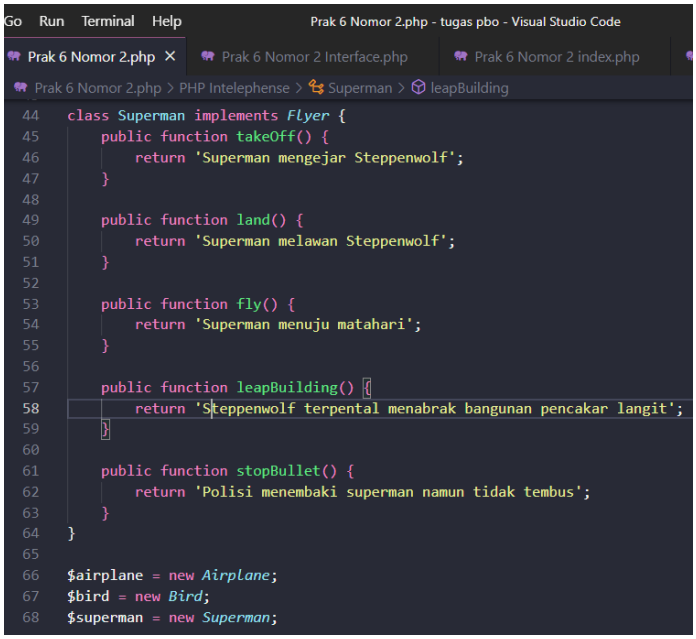
Analisis:

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

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php

require_once 'Prak 6 Nomor 2 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';
    }
}

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';
    }
}

class Superman implements Flyer {
    public function takeOff() {
        return 'Superman mengejar Steppenwolf';
    }

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



Interface

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php

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

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

Index

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php
    require_once 'Prak 6 Nomor 2.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>Praktikum 6</title>
</head>

<body>
    <div class="container">
        <br>
        <h2 class="text-center">Praktikum 6 PBO</h2>
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <h4 class="text-center"><strong><u>Nomor 2</u></strong></h4>
                <br><br>
```

```
                <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>
```

```
                        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>
```

Output:

Praktikum 6 PBO

Nomor 2

Superman
Superman melawan Steppenwolf
Superman mengejar Steppenwolf
Superman menuju matahari
Steppenwolf terpental menabrak bangunan pencakar langit
Polisi menembaki superman namun tidak tembus

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

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php

require_once 'Prak 6 Nomor 3 Abstrac.php';
require_once 'Prak 6 Nomor 3 Interface.php';

class Animal
{
    protected $name;

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

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

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";
    }

    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 Steppenwolf";
    }

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

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

    public function leapBuilding()
    {
        return "Steppenwolf terpental menabrak bangunan pencakar langit";
    }

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

$burung = new Animal('Burung');
$manusia = new Homosapiens('Aransha');
$airplane2 = new Airplane2(27000, 'Lion Air');
$superman2 = new Superman2('Superman');
```

Interface

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php

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

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

Abstract

Prak 6 Nomor 3 Abstrac.php > ...

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?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;
    }

    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name ditambahkan muatan sebesar $weight <br>";
```

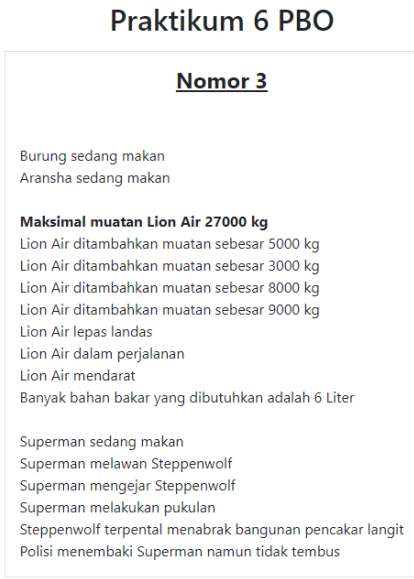
```php
    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name ditambahkan muatan sebesar $weight <br>";
            echo 'Muatan telah penuh tidak bisa menambah lagi';
        }else {
            $this->load += $weight;
            echo "$this->name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

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 70000000;
        $range /= $this->load;
        return $range;
    }

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

Index

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php
    require_once 'Prak 6 Nomor 3.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>Praktikum 6</title>
</head>
<body>
    <div class="container">
        <h2 class="text-center">Praktikum 6 PBO</h2>
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <h4 class="text-center"><strong><u>Nomor 3</u></strong></h4>
                <br><br>
                <?= $burung->eat(); ?> <br>
                <?= $manusia->eat(); ?> <br>
```

```
                <br>

                <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $airplane2->addBox(5000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(3000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(8000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(9000) . ' kg'; ?> <br>
                <?= $airplane2->takeOff(); ?> <br>
                <?= $airplane2->fly(); ?> <br>
                <?= $airplane2->land(); ?> <br>

                <?php
                        echo "Banyak bahan bakar yang dibutuhkan adalah " . $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>
```

Output:



Analisis:

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

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php

require_once 'Prak 6 Nomor 4 Abstract.php';
require_once 'Prak 6 Nomor 4 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';
    }
}

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';
    }

    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);
    }
    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(31000, 'Atomic');
$seaPlane = new SeaPlane(25000, 'Titanic');
$helicopter = new Helicopter(13000, 'Brocklyn');
```

Interface

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?php

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

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

Abstract

```php
<!--- Affandika --->
<!--- 21091397030 --->

<?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;
    }

    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name menambah muatan sebesar $weight <br>";
```

```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 = 70000000;
        $range /= $this->load;
        return $range;
    }

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

Index

```php
<!--- Affandika --->
<!--- 21091397030 --->

    <?php
    require_once 'Prak 6 Nomor 4.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">Praktikum 6 PBO</h2>
        <div class="container">
  <div class="row">
  <h4 class="text-center"><strong>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(10000) . ' kg'; ?> <br>
                <b><?= $riverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $riverBarge2->addBox(10000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(15000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(2000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(4000) . ' kg'; ?> <br>
                <?= $riverBarge2->dock(); ?> <br>
                <?= $riverBarge2->cruise(); ?> <br>
                <?php
                        echo "Banyak bahan bakar yang dibutuhkan " . $riverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
                    ?>
    </div>
    <div class="col-4 mx-auto border p-2 mt-2">
                <b><?= $seaPlane->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $seaPlane->addBox(16000) . ' kg'; ?> <br>
                <?= $seaPlane->addBox(9000) . ' kg'; ?> <br>
                <?= $seaPlane->dock(); ?> <br>
                <?= $seaPlane->cruise(); ?> <br>
                <?= $seaPlane->takeOff(); ?> <br>
                <?= $seaPlane->fly(); ?> <br>
                <?= $seaPlane->land(); ?> <br>
                <?php
                        echo "Banyak bahan bakar yang dibutuhkan " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
                    ?>
    </div>
    <div class="col mx-auto border p-2 mt-2">
                <?= $seaPlane->fly(); ?> <br>
                <?= $seaPlane->land(); ?> <br>
                <?php
                        echo "Banyak bahan bakar yang dibutuhkan " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
                    ?>
    </div>
    <div class="col mx-auto border p-2 mt-2">
                <b><?= $helicopter->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $helicopter->addBox(7000) . ' kg'; ?> <br>
                <?= $helicopter->addBox(6000) . ' kg'; ?> <br>
                <?= $helicopter->takeOff(); ?> <br>
                <?= $helicopter->fly(); ?> <br>
                <?= $helicopter->land(); ?> <br>
                <?php
                        echo "Banyak bahan bakar yang dibutuhkan " . $helicopter->calcFuelNeeds() . ' Liter'. '<br>';
                    ?>
    </div>
  </div>
</div>
</body>

</html>
```

Output:

Praktikum 6 PBO

Nomor 4

Maksimal muatan Atomic 31000 kg	**Maksimal muatan Titanic 25000 kg**	**Maksimal muatan Brocklyn 13000 kg**
Atomic menambah muatan sebesar 10000 kg	Titanic menambah muatan sebesar 16000 kg	Brocklyn menambah muatan sebesar 7000 kg
Atomic menambah muatan sebesar 15000 kg	Titanic menambah muatan sebesar 9000 kg	Brocklyn menambah muatan sebesar 6000 kg
Atomic menambah muatan sebesar 2000 kg	Titanic berada di dermaga	Brocklyn lepas landas
Atomic menambah muatan sebesar 4000 kg	Titanic sedang berlayar	Brocklyn dalam perjalanan
Atomic berada di dermaga	Titanic lepas landas	Brocklyn mendarat
Atomic sedang berlayar	Titanic dalam perjalanan	Banyak bahan bakar yang dibutuhkan 11 Liter
Banyak bahan bakar yang dibutuhkan 5 Liter	Titanic mendarat	
	Banyak bahan bakar yang dibutuhkan 6 Liter	

Analisis:

Implementasi polymhorphism dengan interface dan abstact class detinjukkan pada class 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