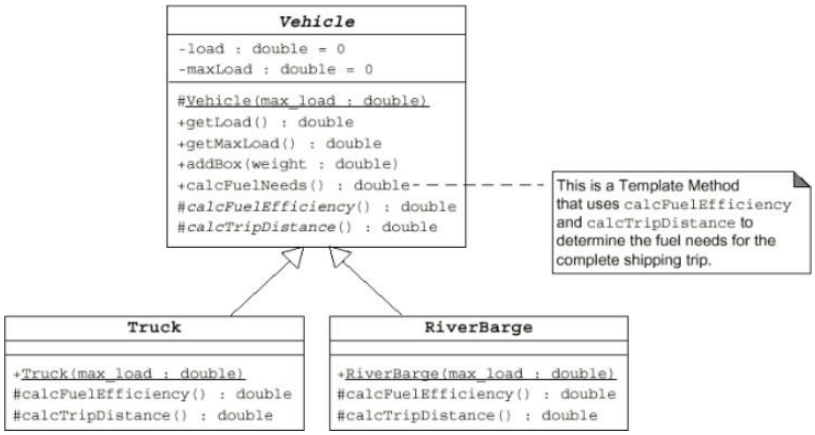
Nama : Dwi Ramadhaniasari

NIM : 21091397057

Kelas : 2021A

TUGAS PRAKTIKUM 6

1. Buat program berdasarkan UML berikut.



- PHP

index no1.php

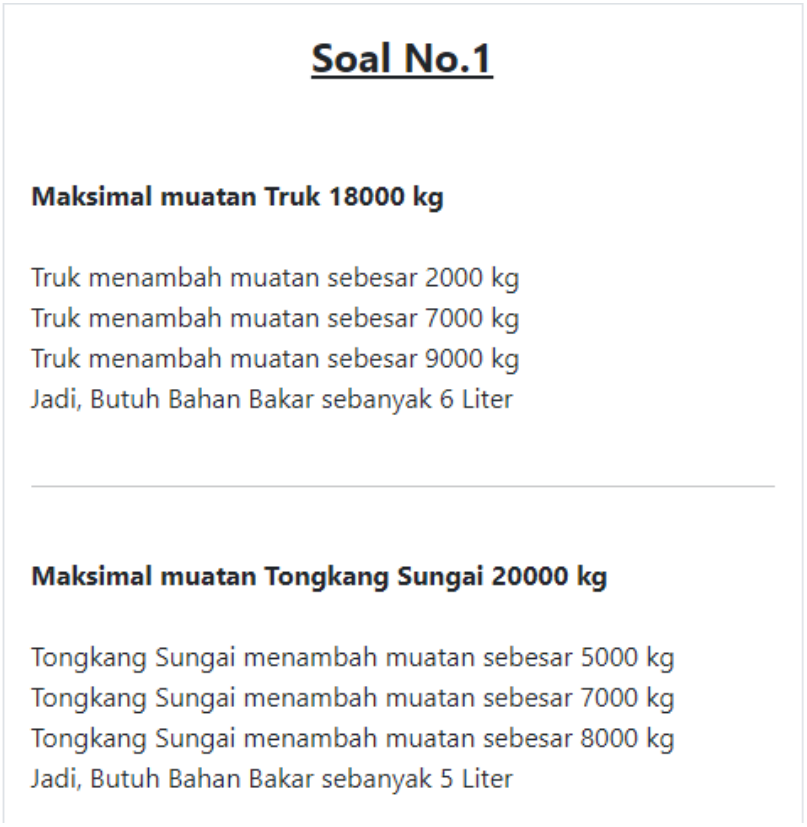
```php
<!-- Dwi ramadhaniasari
    21091397057 -->

<?php
    require_once 'prak6 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>Praktikum 6</title>
</head>

<body>
    <div class="container">
        <br>
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <h4 class="text-center"><strong><u>Soal No.1</u></strong></h4>
                <br><br>
                <b><?= $truck->getMaxLoad() . ' kg'; ?> <br></b>
                <br>
                <?= $truck->addBox(2000) . ' kg'; ?> <br>
                <?= $truck->addBox(7000) . ' kg'; ?> <br>
                <?= $truck->addBox(9000) . ' kg'; ?> <br>
```

```php
                <?php
                    echo "Jadi, Butuh Bahan Bakar sebanyak " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
                ?>
            <br>
            <hr>
            <br>
            <b><?= $riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
            <br>
            <?= $riverBarge->addBox(5000) . ' kg'; ?> <br>
            <?= $riverBarge->addBox(7000) . ' kg'; ?> <br>
            <?= $riverBarge->addBox(8000) . ' kg'; ?> <br>

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

</html>
```

prak6 no1.php

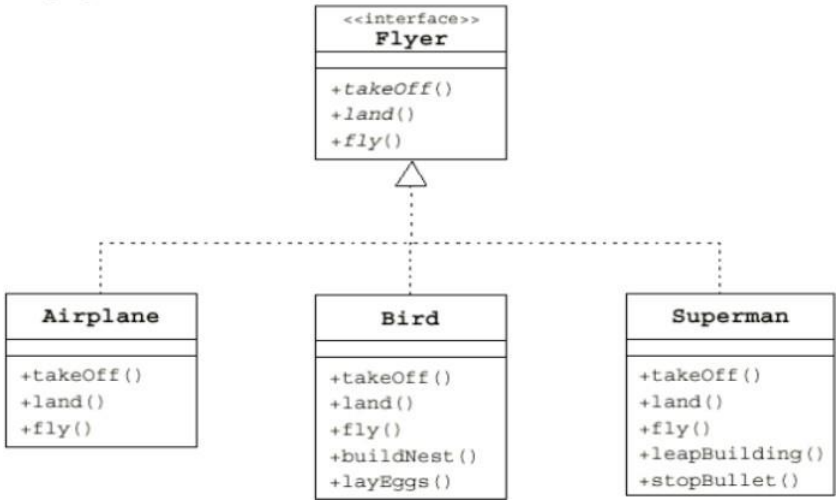
```php
<!-- Dwi ramadhaniasari
     21091397057 -->
<?php

require_once 'abstract no1.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(18000, 'Truk');
$riverBarge = new RiverBarge(20000, 'Tongkang Sungai');
```

- Output



- Analisa
  Program tersebut merupakan 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. Buat Program berdasarkan UML berikut.



- PHP

index no 2.php

```php
<!-- Dwi Ramadhaniasari
     21091397057 -->

<?php
    require_once 'no 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>
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <h4 class="text-center"><strong><u>Soal 2</u></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>
```

no 2.php

```php
<!-- Dwi ramadhaniasari
     21091397057 -->

<?php

require_once 'interface no2.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';
    }

```

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

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

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;
```

interface no2.php

```php
<!-- Dwi Ramadhaniasari
    21091397057 -->

<?php

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

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

- Output

Soal 2

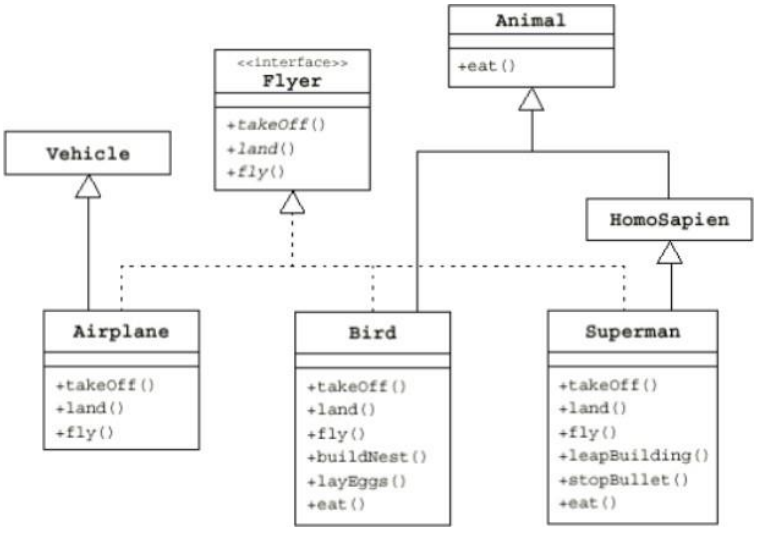
Superman
Superman melawan Batman
Superman mengejar Batman
Superman melancarkan pukulan
Batman terpental menabrak bangunan pencakar langit
Polisi menembaki superman namun ditangkis

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

Airplane
Pesawat lepas landas..
Pesawat dalam perjalanan
Pesawat mendarat

- Analisa
  Program tersebut merupakan implementasi Polymhorpism dengan penggunaan Interface Flyer. Sehingga semua class yang implementasi dari interface Flyer harus memiliki mehod take off, land, land, dan fly.

3. Buat program berdasarkan UML berikut.



- PHP

index no 3.php

```php
<!-- Dwi Ramadhaniasari
    21091397057 -->

<?php
    require_once 'no 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">
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <h4 class="text-center"><strong><u>Soal 3</u></strong></h4>
                <br><br>
                <?= $burung->eat(); ?> <br>
                <?= $manusia->eat(); ?> <br>
                <br>

                <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $airplane2->addBox(6000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(2000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(7000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(5000) . ' 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>
```

no 3.php

```php
<?php

require_once 'abstrac no 3.php';
require_once 'interface no 3.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 Batman";
    }

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

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

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

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

$burung = new Animal('Burung');
$manusia = new Homosapiens('Aransha');
$airplane2 = new Airplane2(25000, 'Batik Air');
$superman2 = new Superman2('Superman');
```

```php
<!-- Dwi Ramadhaniasari
     21091397057 -->

<?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>";
            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;
    }
}
```

index no 3.php | abstrac no 3.php | interface no 3.php | no 3.php

```php
<?php

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

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

- Output

Soal 3

Burung sedang makan
Aransha sedang makan

Maksimal muatan Batik Air 25000 kg
Batik Air menambah muatan sebesar 6000 kg
Batik Air menambah muatan sebesar 2000 kg
Batik Air menambah muatan sebesar 7000 kg
Batik Air menambah muatan sebesar 5000 kg
Batik Air lepas landas
Batik Air dalam perjalanan
Batik Air mendarat
Jadi, Butuh Bahan Bakar sebanyak 5 Liter

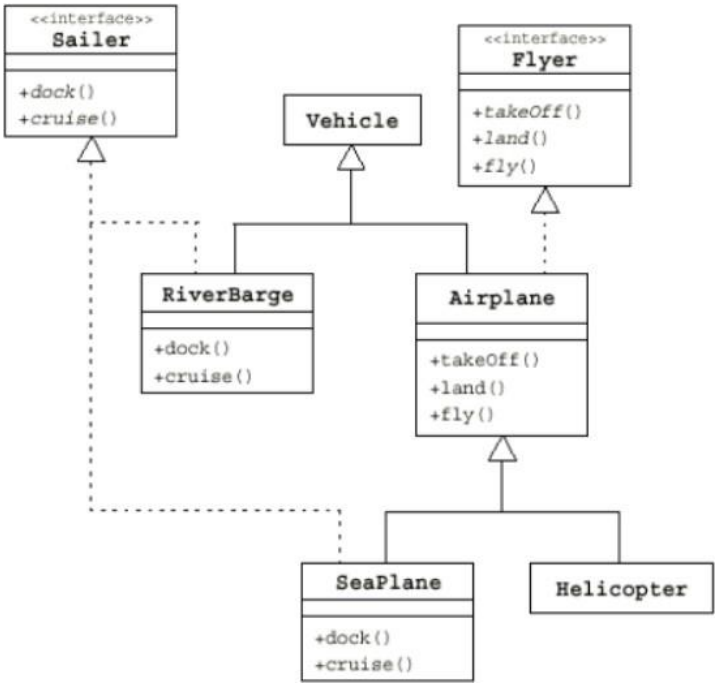
Superman sedang makan
Superman melawan Batman
Superman mengejar Batman
Superman melancarkan pukulan
Batman terpental menabrak bangunan pencakar langit
Polisi menembaki Superman namun ditangkis

- Analisa

Pada Program tersebut 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 dan implementasi dari interface Flyer. Maka class Superman memiliki method eat, takeoff, land, fly.

4. Buat Program UML



- PHP

abstract no 4.php | index no 4.php | prak 6 no 4.php | interface no 4.php

```php
<!-- Dwi Ramadyaniasari
    2109139701 -->

<?php
    require_once 'prak 6 no 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>Praktikum 6</title>
</head>
<body>
    <div class="container">
  <div class="row">
  <h4 class="text-center"><strong><u>Soal No.4</u></strong></h4>
  <br><br>
    <div class="col-4 mx-auto border p-2 mt-2">
                <b><?= $riverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $riverBarge2->addBox(15000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(10000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(8000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(2000) . ' kg'; ?> <br>
                <?= $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(15000) . ' kg'; ?> <br>
                <?= $seaPlane->addBox(7000) . ' 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(5000) . ' kg'; ?> <br>
                <?= $helicopter->addBox(7000) . ' 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>
```

abstract no 4.php | index no 4.php | prak 6 no 4.php | interface no 4.php

```php
<!-- Dwi ramadhaniasari
    21091397057 -->

<?php

require_once 'abstract no 4.php';
require_once 'interface no 4.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';
```

```php
<!-- Dwi ramadhaniasari
    21091397057 -->

<?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>";
            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

Soal No.4

Maksimal muatan Atomic 40000 kg	**Maksimal muatan Titanic 30000 kg**	**Maksimal muatan Brocklyn 15000 kg**
Atomic menambah muatan sebesar 15000 kg	Titanic menambah muatan sebesar 15000 kg	Brocklyn menambah muatan sebesar 5000 kg
Atomic menambah muatan sebesar 10000 kg	Titanic menambah muatan sebesar 7000 kg	Brocklyn menambah muatan sebesar 7000 kg
Atomic menambah muatan sebesar 8000 kg	Titanic berada di dermaga	Brocklyn lepas landas
Atomic menambah muatan sebesar 2000 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 9 Liter
Jadi, Butuh Bahan Bakar sebanyak 3 Liter	Titanic mendarat	
	Jadi, Butuh Bahan Bakar sebanyak 5 Liter	

- Analisa

  Program tersebut merupakan implementasi polymhorphism dengan interface dan abstract class di tunjukkan 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.