LAPORAN PEMROGRAMAN BERORIENTASI OBJEK
UJIAN TENGAH SEMESTER

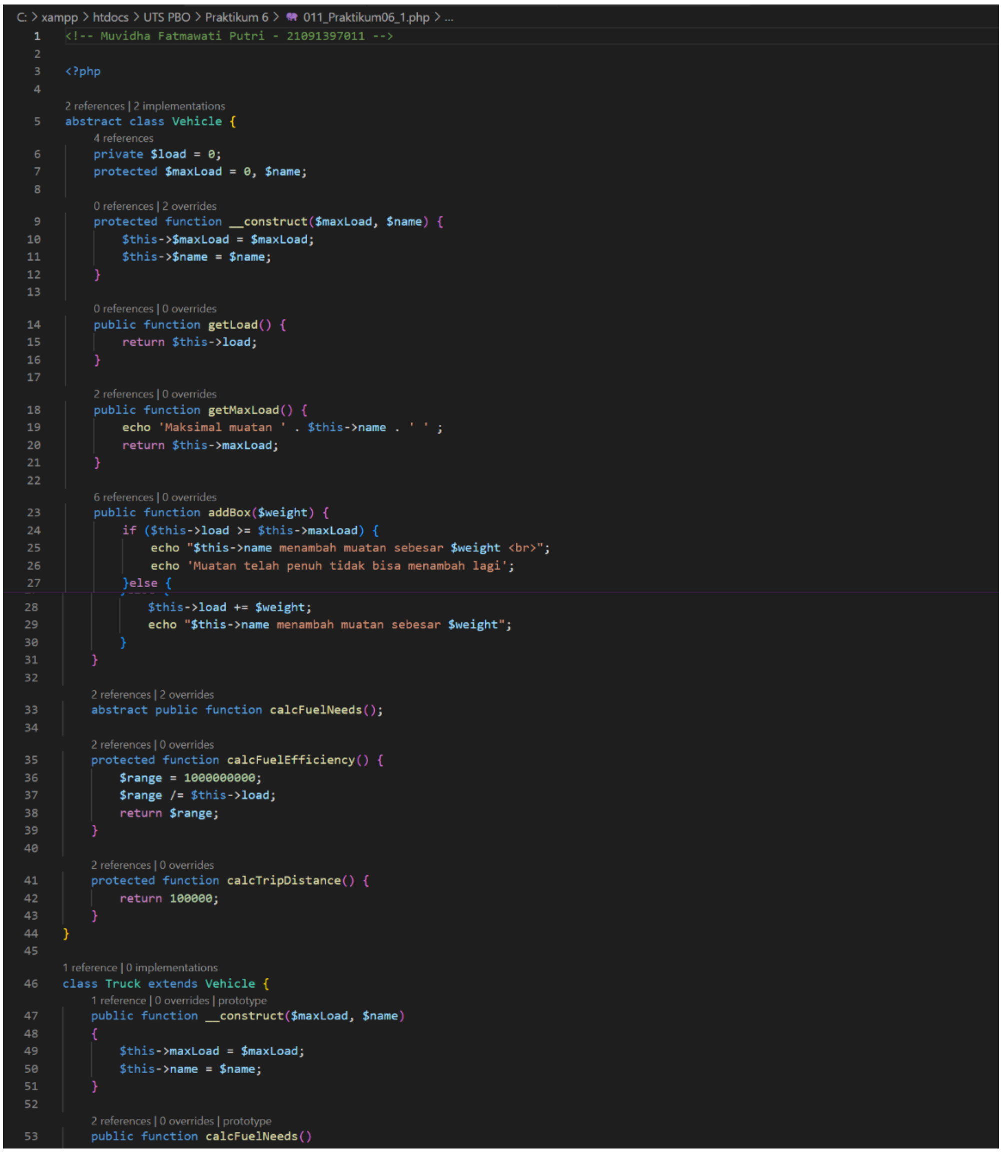
Disusun oleh :

Muvidha Fatmawati Putri (21091397011)
A2021 MI

PROGRAM STUDI D4 MANAJEMEN INFORMATIKA
FAKULTAS VOKASI
UNIVERSITAS NEGERI SURABAYA
2022

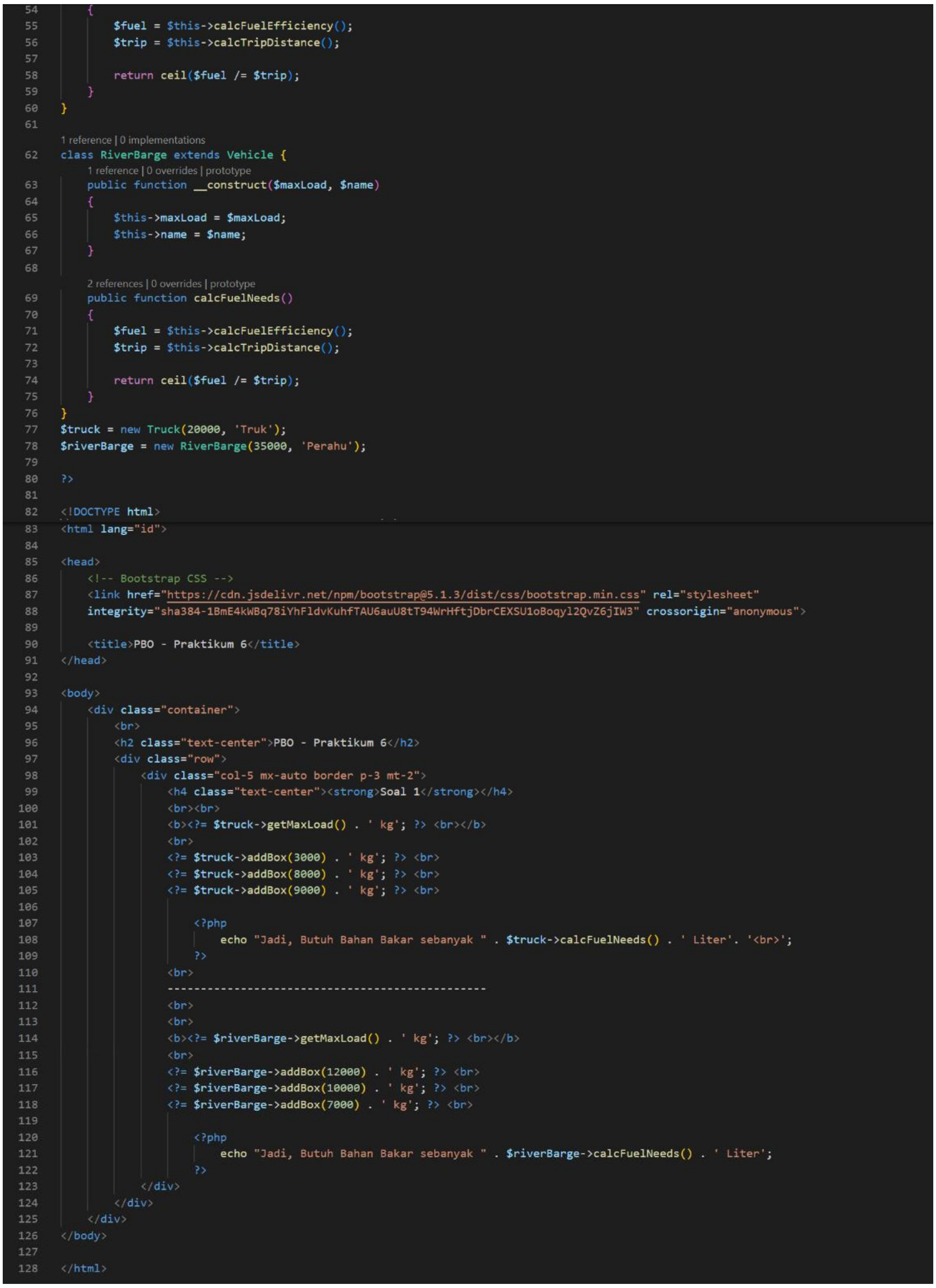
1. Buat program berdasarkan UML berikut

    - Kodingan :



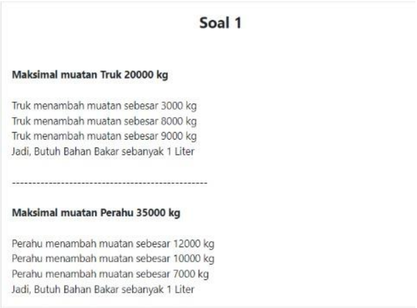
C: > xampp > htdocs > UTS PBO > Praktikum 6 > 011_Praktikum06_1.php > ...

```php
<!-- Muvidha Fatmawati Putri - 21091397011 -->

<?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 = 1000000000;
        $range /= $this->load;
        return $range;
    }

    protected function calcTripDistance() {
        return 100000;
    }
}

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);
        }
    }

1 reference | 0 implementations
class RiverBarge extends Vehicle {
    1 reference | 0 overrides | prototype
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    2 references | 0 overrides | prototype
    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
}
$truck = new Truck(20000, 'Truk');
$riverBarge = new RiverBarge(35000, 'Perahu');

?>

<!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 1</strong></h4>
                <br><br>
                <b><?= $truck->getMaxLoad() . ' kg'; ?> <br></b>
                <br>
                <?= $truck->addBox(3000) . ' kg'; ?> <br>
                <?= $truck->addBox(8000) . ' kg'; ?> <br>
                <?= $truck->addBox(9000) . ' 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(12000) . ' kg'; ?> <br>
                <?= $riverBarge->addBox(10000) . ' kg'; ?> <br>
                <?= $riverBarge->addBox(7000) . ' kg'; ?> <br>

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

</html>
```

- Output :



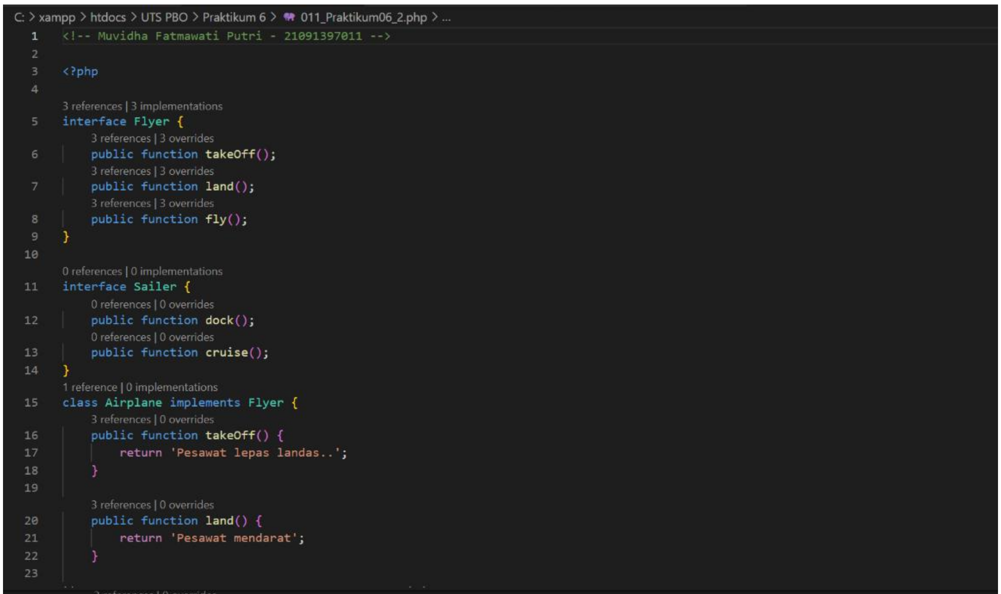
```
Soal 1

Maksimal muatan Truk 20000 kg

Truk menambah muatan sebesar 3000 kg
Truk menambah muatan sebesar 8000 kg
Truk menambah muatan sebesar 9000 kg
Jadi, Butuh Bahan Bakar sebanyak 1 Liter

------------------------------------------------

Maksimal muatan Perahu 35000 kg

Perahu menambah muatan sebesar 12000 kg
Perahu menambah muatan sebesar 10000 kg
Perahu menambah muatan sebesar 7000 kg
Jadi, Butuh Bahan Bakar sebanyak 1 Liter
```

- Penjelasan :

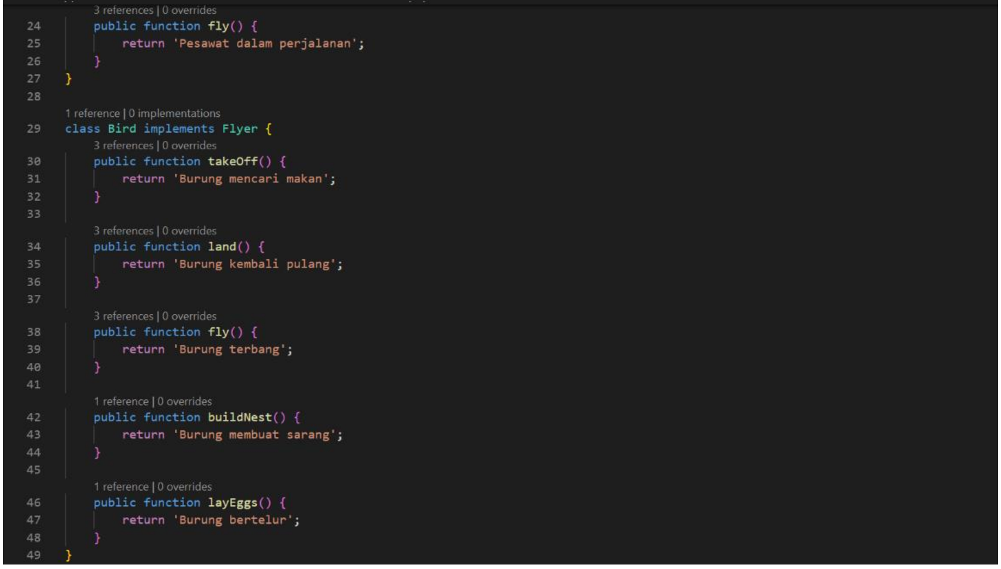
  Implementasi dari abstract class pada class Vehicle, method calcFuelNeeds digunakan untuk menghitung bahan bakar yang digunakan. Abstract method diletakkan pada class Vehicle sebagai parent class dan diakses oleh child classnya yaitu class Truk, dan class RiverBarge yang akan mengembalikan nilai yang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

2. Buat program berdasarkan UML berikut

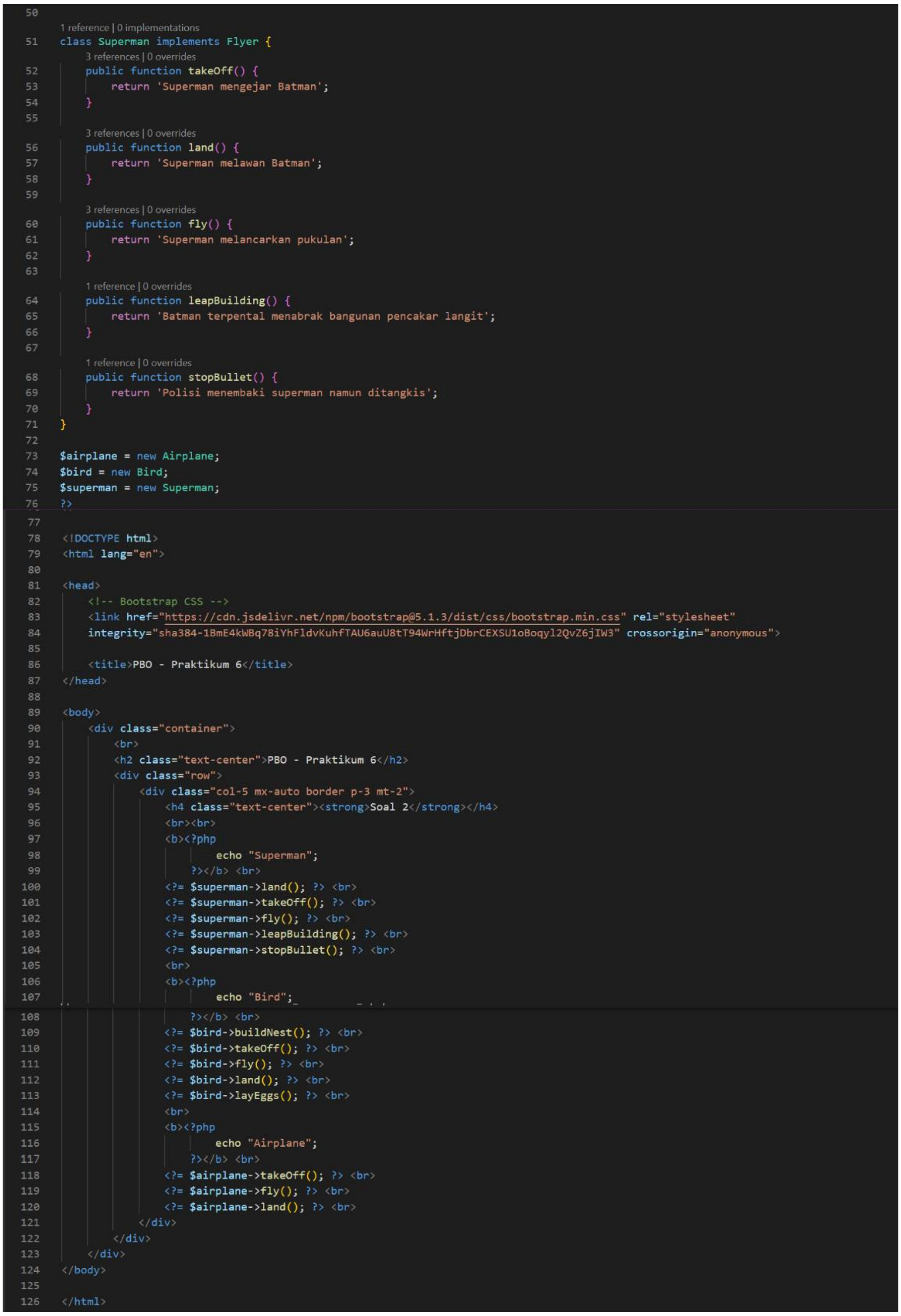
- Kodingan :


C: > xampp > htdocs > UTS PBO > Praktikum 6 > 011_Praktikum06_2.php > ...

```
<!-- Muvidha Fatmawati Putri - 21091397011 -->

<?php

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

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

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

```

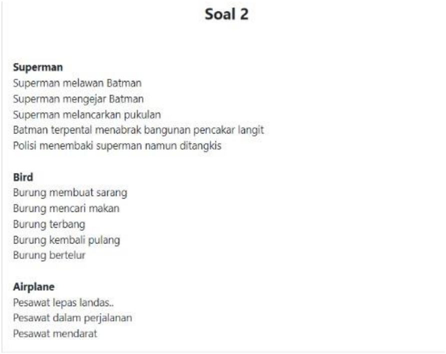


```
    public function fly() {
        return 'Pesawat dalam perjalanan';
    }
}

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';
    }
}
```



```
1 reference | 0 implementations
class Superman implements Flyer {
    3 references | 0 overrides
    public function takeOff() {
        return 'Superman mengejar Batman';
    }

    3 references | 0 overrides
    public function land() {
        return 'Superman melawan Batman';
    }

    3 references | 0 overrides
    public function fly() {
        return 'Superman melancarkan pukulan';
    }

    1 reference | 0 overrides
    public function leapBuilding() {
        return 'Batman terpental menabrak bangunan pencakar langit';
    }

    1 reference | 0 overrides
    public function stopBullet() {
        return 'Polisi menembaki superman namun ditangkis';
    }
}

$airplane = new Airplane;
$bird = new Bird;
$superman = new Superman;
?>

<!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 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>
```

- Output :



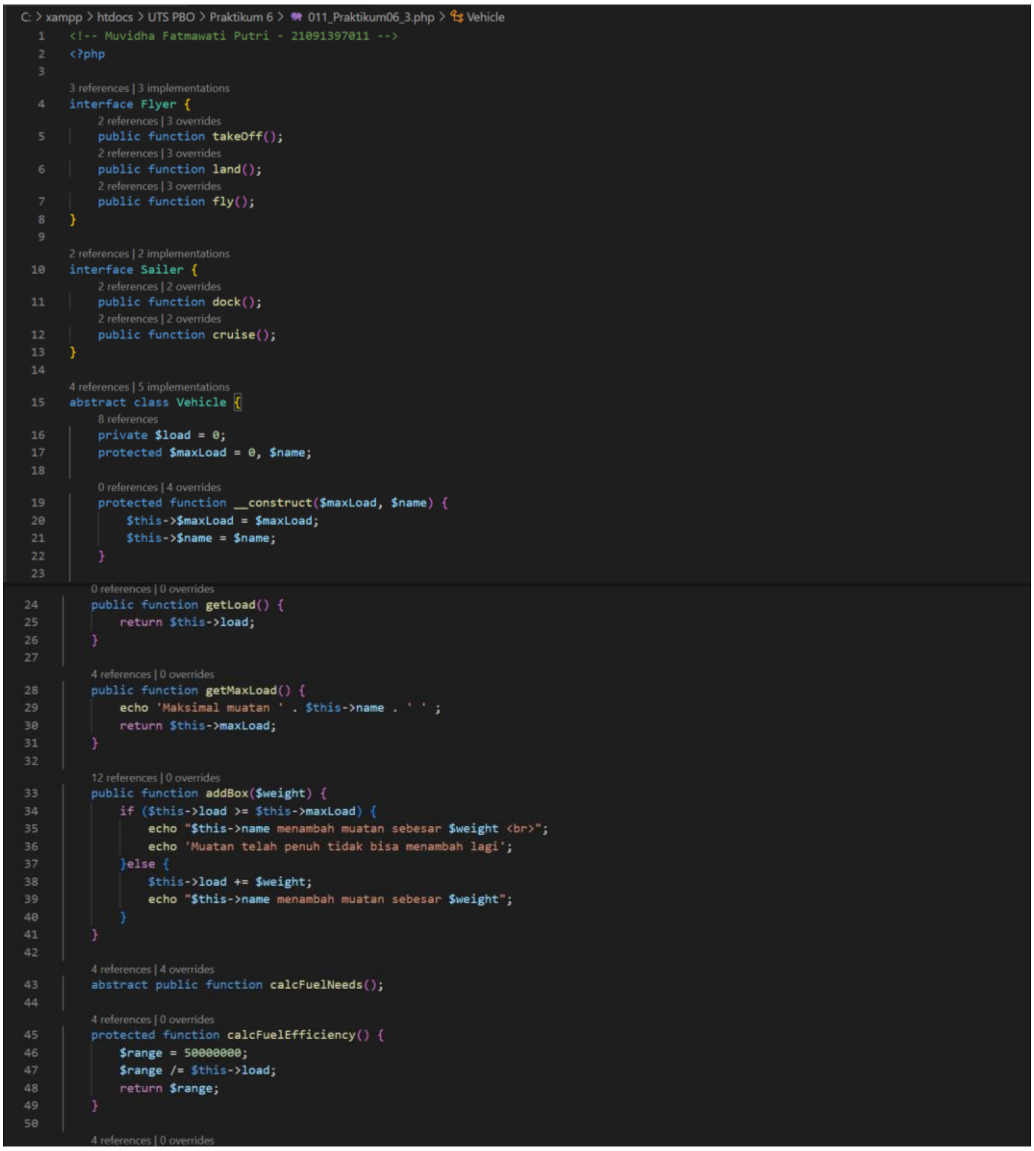
```
Soal 2

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
```

- Penjelasan :

  Implementasi Polymhorpism dengan penggunaan Interface Flyer sehinggasemua classyang Implements dari interface Fyler harus memiliki method takeoff, land, dan fly.

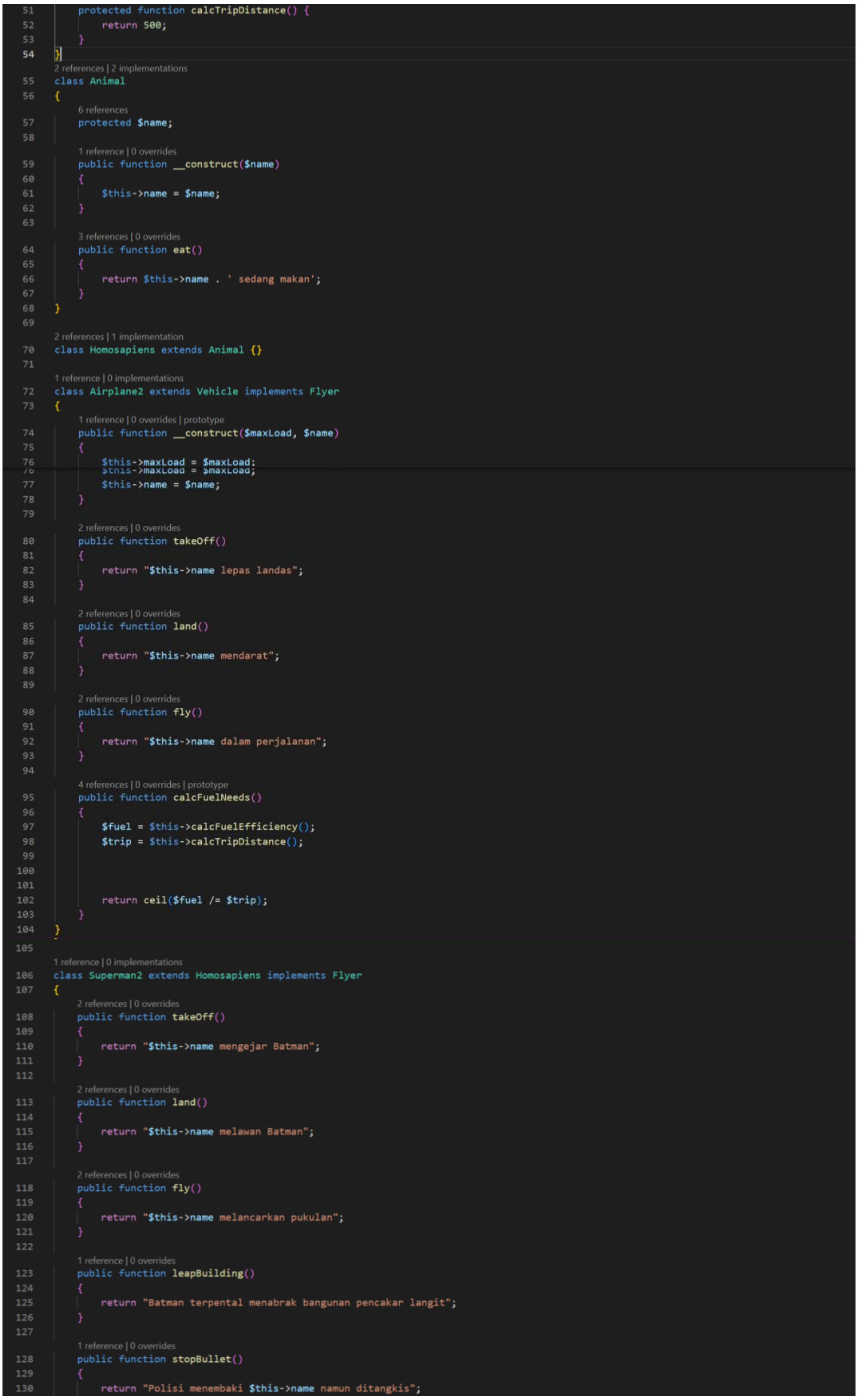
3. Buat program berdasarkan UML berikut

- Kodingan :

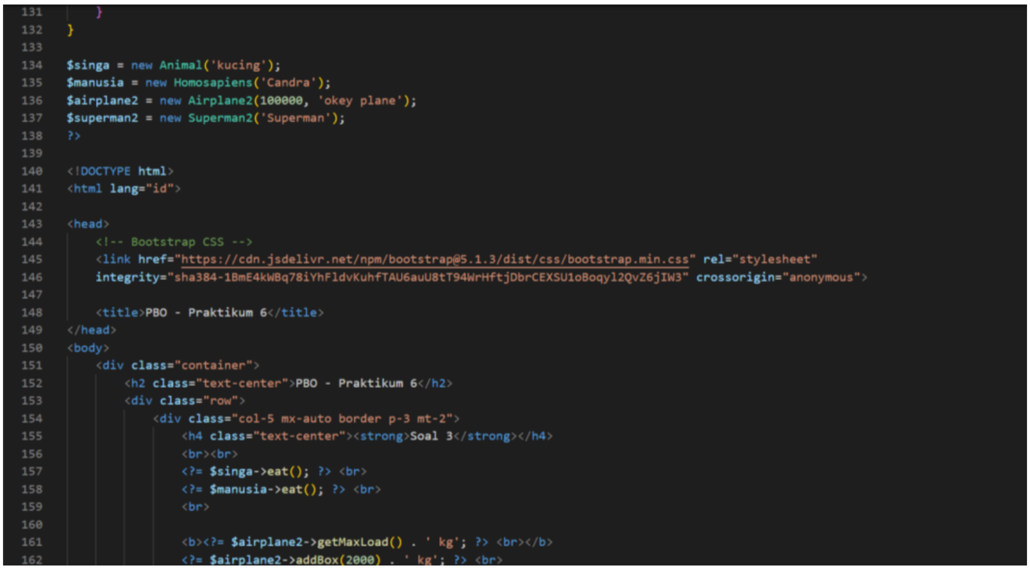


C: > xampp > htdocs > UTS PBO > Praktikum 6 > 011_Praktikum06_3.php > Vehicle

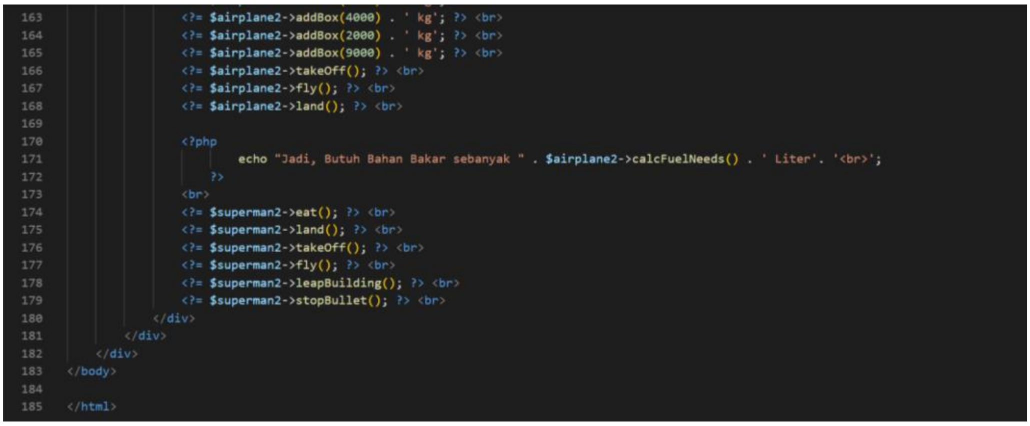
```php
<!-- Muvidha Fatmawati Putri - 21091397011 -->
<?php

3 references | 3 implementations
interface Flyer {
    2 references | 3 overrides
    public function takeOff();
    2 references | 3 overrides
    public function land();
    2 references | 3 overrides
    public function fly();
}

2 references | 2 implementations
interface Sailer {
    2 references | 2 overrides
    public function dock();
    2 references | 2 overrides
    public function cruise();
}

4 references | 5 implementations
abstract class Vehicle {
    8 references
    private $load = 0;
    protected $maxLoad = 0, $name;

    0 references | 4 overrides
    protected function __construct($maxLoad, $name) {
        $this->$maxLoad = $maxLoad;
        $this->$name = $name;
    }

    0 references | 0 overrides
    public function getLoad() {
        return $this->load;
    }

    4 references | 0 overrides
    public function getMaxLoad() {
        echo 'Maksimal muatan ' . $this->name . ' ' ;
        return $this->maxLoad;
    }

    12 references | 0 overrides
    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";
        }
    }

    4 references | 4 overrides
    abstract public function calcFuelNeeds();

    4 references | 0 overrides
    protected function calcFuelEfficiency() {
        $range = 50000000;
        $range /= $this->load;
        return $range;
    }

    4 references | 0 overrides
```



```
    protected function calcTripDistance() {
        return 500;
    }
}
2 references | 2 implementations
class Animal
{
    6 references
    protected $name;

    1 reference | 0 overrides
    public function __construct($name)
    {
        $this->name = $name;
    }

    3 references | 0 overrides
    public function eat()
    {
        return $this->name . ' sedang makan';
    }
}

2 references | 1 implementation
class Homosapiens extends Animal {}

1 reference | 0 implementations
class Airplane2 extends Vehicle implements Flyer
{
    1 reference | 0 overrides | prototype
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    2 references | 0 overrides
    public function takeOff()
    {
        return "$this->name lepas landas";
    }

    2 references | 0 overrides
    public function land()
    {
        return "$this->name mendarat";
    }

    2 references | 0 overrides
    public function fly()
    {
        return "$this->name dalam perjalanan";
    }

    4 references | 0 overrides | prototype
    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();



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

1 reference | 0 implementations
class Superman2 extends Homosapiens implements Flyer
{
    2 references | 0 overrides
    public function takeOff()
    {
        return "$this->name mengejar Batman";
    }

    2 references | 0 overrides
    public function land()
    {
        return "$this->name melawan Batman";
    }

    2 references | 0 overrides
    public function fly()
    {
        return "$this->name melancarkan pukulan";
    }

    1 reference | 0 overrides
    public function leapBuilding()
    {
        return "Batman terpental menabrak bangunan pencakar langit";
    }

    1 reference | 0 overrides
    public function stopBullet()
    {
        return "Polisi menembaki $this->name namun ditangkis";
```

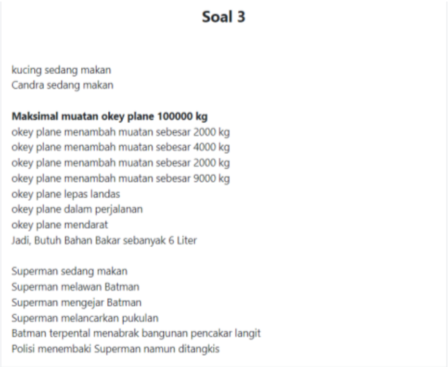


```
    }
}

$singa = new Animal('kucing');
$manusia = new Homosapiens('Candra');
$airplane2 = new Airplane2(100000, 'okey plane');
$superman2 = new Superman2('Superman');
?>

<!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 3</strong></h4>
                <br><br>
                <?= $singa->eat(); ?> <br>
                <?= $manusia->eat(); ?> <br>
                <br>

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



```
                <?= $airplane2->addBox(4000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(2000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(9000) . ' 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>
```

- Output :



Soal 3

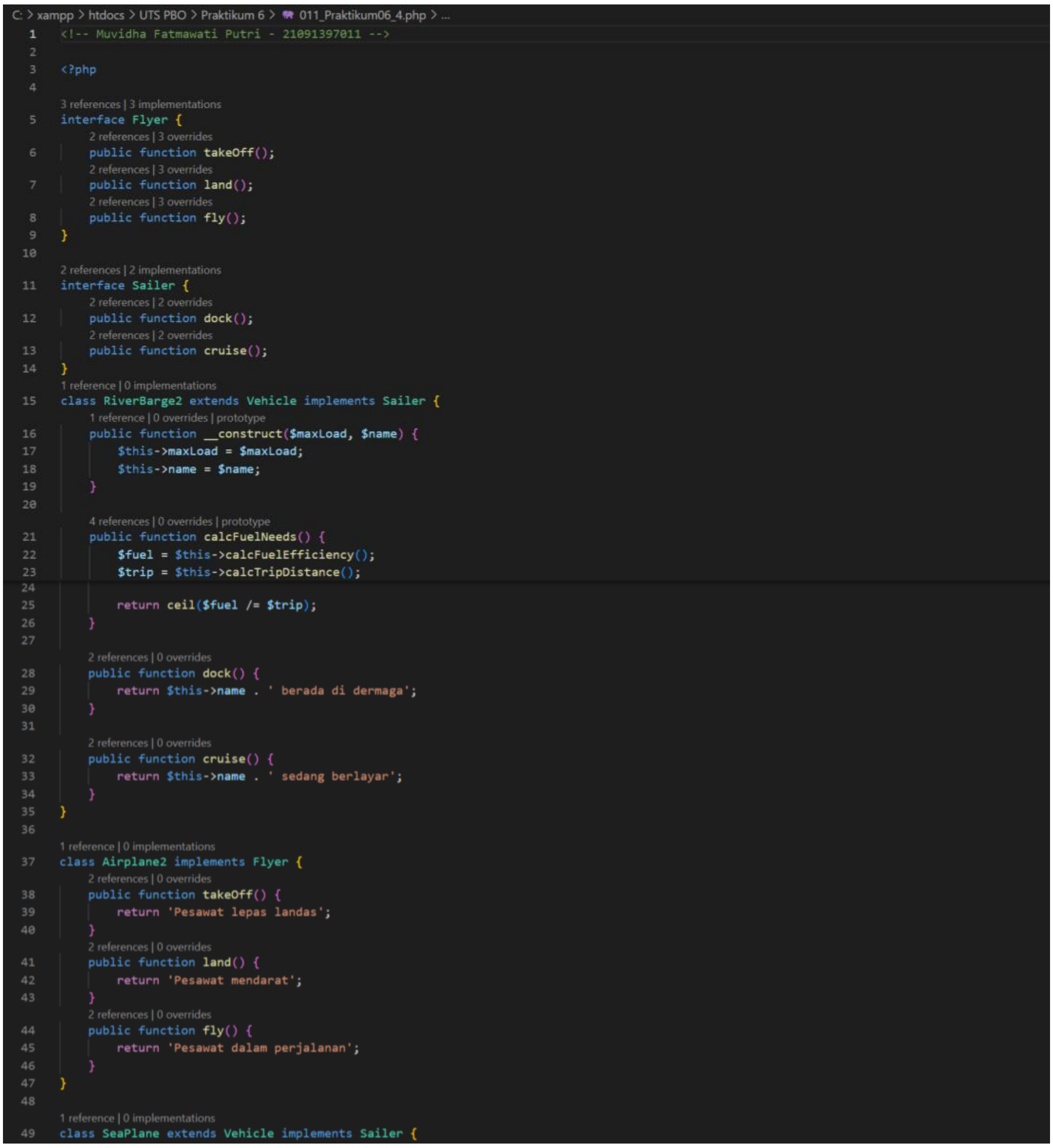
```
kucing sedang makan
Candra sedang makan

Maksimal muatan okey plane 100000 kg
okey plane menambah muatan sebesar 2000 kg
okey plane menambah muatan sebesar 4000 kg
okey plane menambah muatan sebesar 2000 kg
okey plane menambah muatan sebesar 9000 kg
okey plane lepas landas
okey plane dalam perjalanan
okey plane 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
```

- Penjelasan :

  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 homosapiensyang juga turunan dari Animal, serta implementasi dari interface Flyer. Maka class Superman memiliki method eat, takeoff, land, fly.

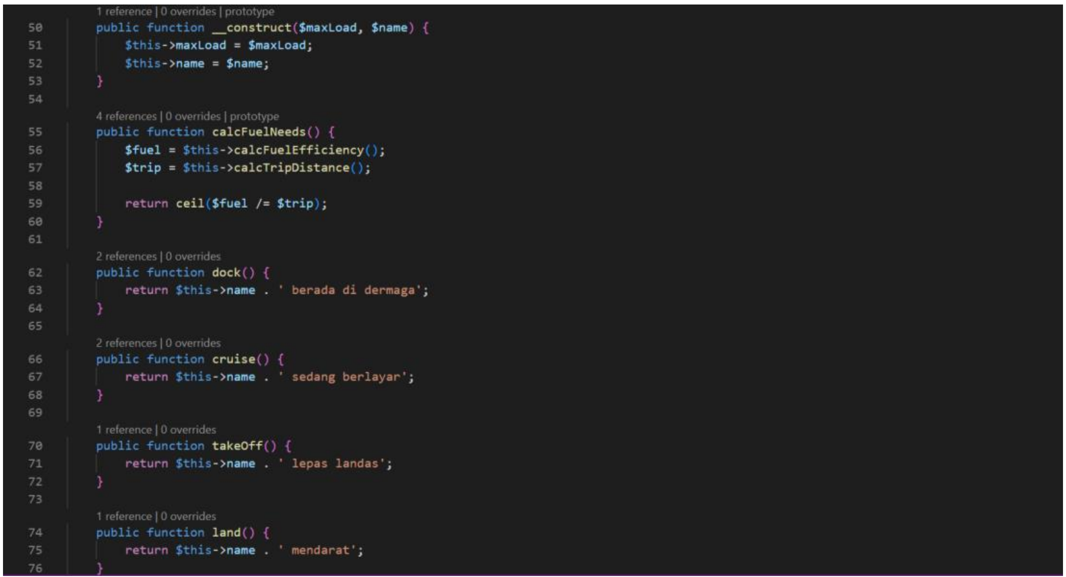
4. Buat program berdasarkan UML berikut

- Kodingan

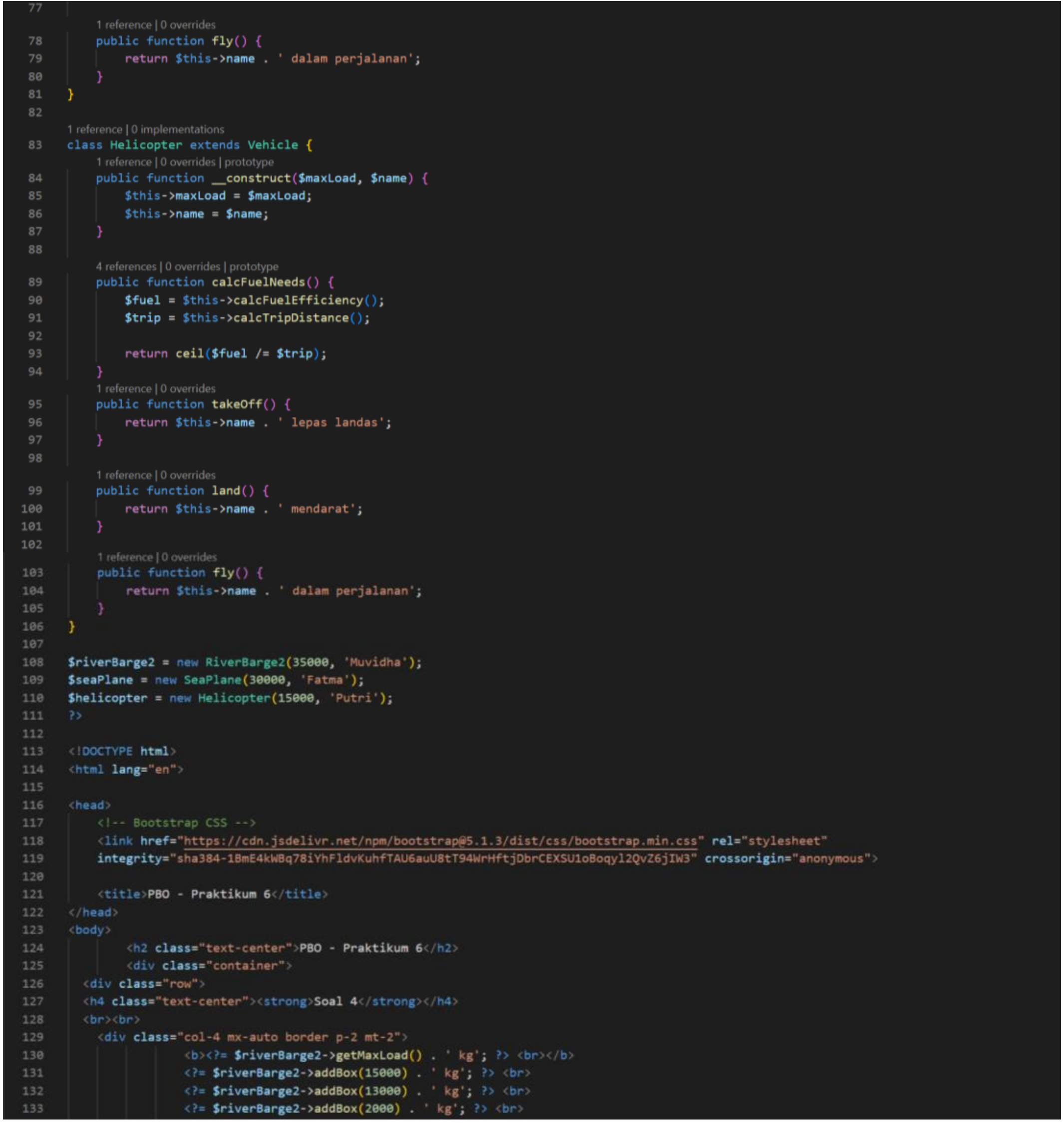


C: > xampp > htdocs > UTS PBO > Praktikum 6 > 011_Praktikum06_4.php > ...

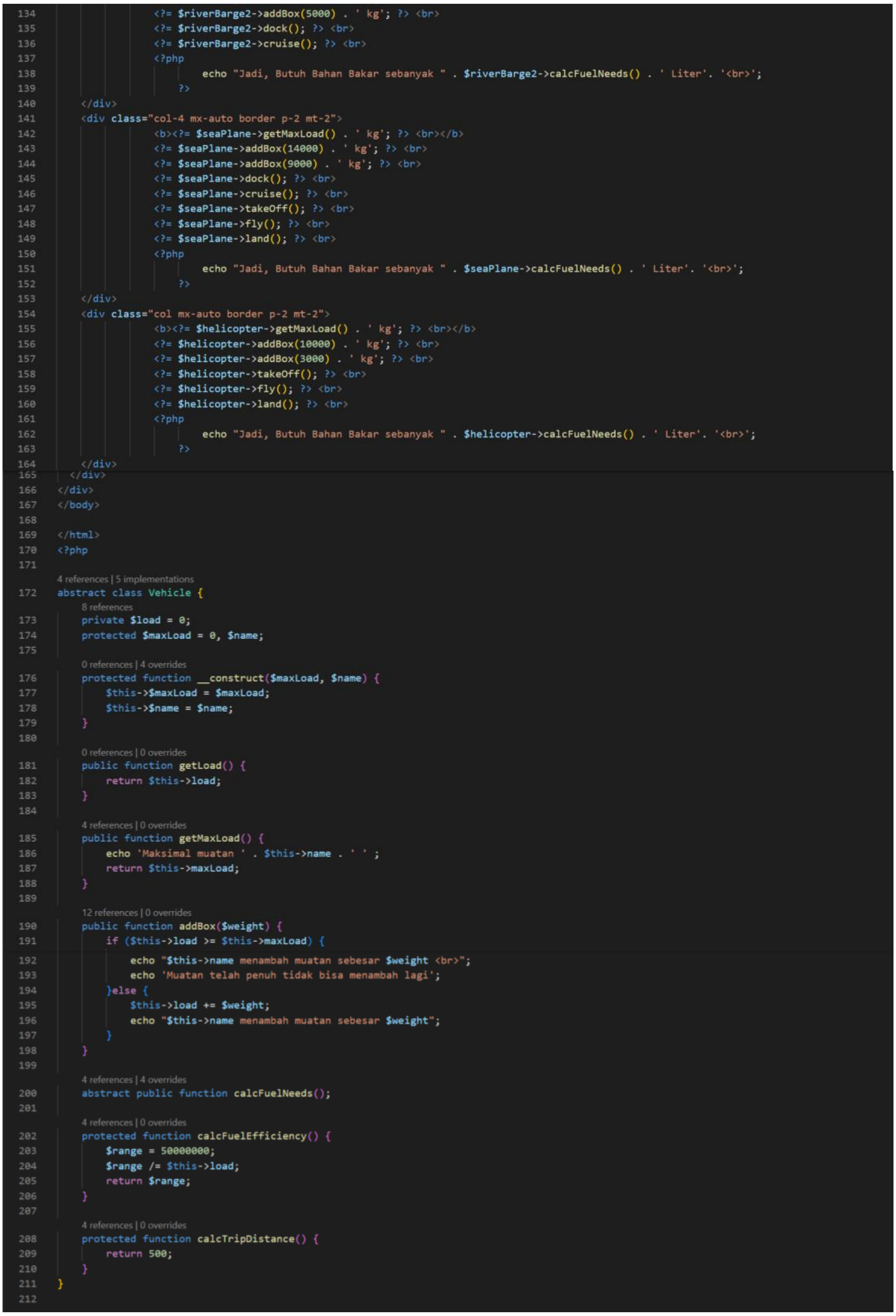
```
<!-- Muvidha Fatmawati Putri - 21091397011 -->

<?php

3 references | 3 implementations
interface Flyer {
    2 references | 3 overrides
    public function takeOff();
    2 references | 3 overrides
    public function land();
    2 references | 3 overrides
    public function fly();
}

2 references | 2 implementations
interface Sailer {
    2 references | 2 overrides
    public function dock();
    2 references | 2 overrides
    public function cruise();
}
1 reference | 0 implementations
class RiverBarge2 extends Vehicle implements Sailer {
    1 reference | 0 overrides | prototype
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    4 references | 0 overrides | prototype
    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

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

    2 references | 0 overrides
    public function dock() {
        return $this->name . ' berada di dermaga';
    }

    2 references | 0 overrides
    public function cruise() {
        return $this->name . ' sedang berlayar';
    }
}

1 reference | 0 implementations
class Airplane2 implements Flyer {
    2 references | 0 overrides
    public function takeOff() {
        return 'Pesawat lepas landas';
    }
    2 references | 0 overrides
    public function land() {
        return 'Pesawat mendarat';
    }
    2 references | 0 overrides
    public function fly() {
        return 'Pesawat dalam perjalanan';
    }
}

1 reference | 0 implementations
class SeaPlane extends Vehicle implements Sailer {
```



```
    1 reference | 0 overrides | prototype
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    4 references | 0 overrides | prototype
    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

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

    2 references | 0 overrides
    public function dock() {
        return $this->name . ' berada di dermaga';
    }

    2 references | 0 overrides
    public function cruise() {
        return $this->name . ' sedang berlayar';
    }

    1 reference | 0 overrides
    public function takeOff() {
        return $this->name . ' lepas landas';
    }

    1 reference | 0 overrides
    public function land() {
        return $this->name . ' mendarat';
    }
```

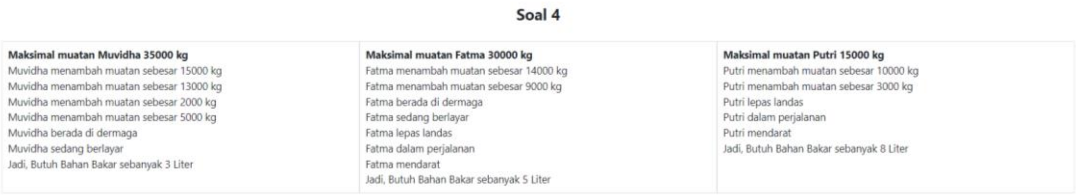


```

    1 reference | 0 overrides
    public function fly() {
        return $this->name . ' dalam perjalanan';
    }
}

1 reference | 0 implementations
class Helicopter extends Vehicle {
    1 reference | 0 overrides | prototype
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    4 references | 0 overrides | prototype
    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
    1 reference | 0 overrides
    public function takeOff() {
        return $this->name . ' lepas landas';
    }

    1 reference | 0 overrides
    public function land() {
        return $this->name . ' mendarat';
    }

    1 reference | 0 overrides
    public function fly() {
        return $this->name . ' dalam perjalanan';
    }
}

$riverBarge2 = new RiverBarge2(35000, 'Muvidha');
$seaPlane = new SeaPlane(30000, 'Fatma');
$helicopter = new Helicopter(15000, 'Putri');
?>

<!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 4</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(13000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(2000) . ' kg'; ?> <br>
```



```
                <?= $riverBarge2->addBox(5000) . ' 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(14000) . ' kg'; ?> <br>
                <?= $seaPlane->addBox(9000) . ' 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(10000) . ' kg'; ?> <br>
                <?= $helicopter->addBox(3000) . ' 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

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 4

Maksimal muatan Muvidha 35000 kg	**Maksimal muatan Fatma 30000 kg**	**Maksimal muatan Putri 15000 kg**
Muvidha menambah muatan sebesar 15000 kg Muvidha menambah muatan sebesar 13000 kg Muvidha menambah muatan sebesar 2000 kg Muvidha menambah muatan sebesar 5000 kg Muvidha berada di dermaga Muvidha sedang berlayar Jadi, Butuh Bahan Bakar sebanyak 3 Liter	Fatma menambah muatan sebesar 14000 kg Fatma menambah muatan sebesar 9000 kg Fatma berada di dermaga Fatma sedang berlayar Fatma lepas landas Fatma dalam perjalanan Fatma mendarat Jadi, Butuh Bahan Bakar sebanyak 5 Liter	Putri menambah muatan sebesar 10000 kg Putri menambah muatan sebesar 3000 kg Putri lepas landas Putri dalam perjalanan Putri mendarat Jadi, Butuh Bahan Bakar sebanyak 8 Liter

- Penjelasan :

  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.