Laporan UTS

Pemrograman Berorientasi Obyek

Oleh:

Putri Dwinatryska A.R.F (21091397075)

PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA (UNESA)

Praktikum 5

1. Suatu program terdiri dari class Pegawai sebagai parent class, class Manajer dan class Kurir sebagai subclass. Buatlah suatu program yang menerapkan konsep polymorphic argument (Polymorphic arguments adalah tipe suatu parameter yang menerima suatu nilai yang bertipe subclass-nya).

```php
<!-- Putri Dwinatryska
    21091397075 -->

    <!DOCTYPE html>
<html lang="id">
<head>
    <link rel="stylesheet" type="text/css" href="no 1.css">
    <title>Praktikum 5</title>
</head>
<body>
    <div class="container">
        <h2><u>Soal No.1</u></h2>

        <?php
    class Pegawai
    {
     public $nama;
     public function __construct($nama)
     {
     $this->nama = $nama;
     }
     public function getNama()
     {
     return $this->nama;
     }
    }
     class Manajer extends Pegawai
     {
     public $tunjangan;
```

prak 5 (1).php > html > body > div.container > Soal1 > Proses

```php
     public function __construct($nama, $tunjangan)
     {
      parent::__construct($nama);
      $this->tunjangan = $tunjangan;
     }
     public function getTunjangan()
     {
      return $this->tunjangan;
     }
    }
    class Kurir extends Pegawai
    {
     public $gaji;

     public function __construct($nama, $gaji)
     {
      parent::__construct($nama);
      $this->gaji = $gaji;
     }
     public function getGaji()
     {
      return $this->gaji;
     }
    }
    class Soal1
    {
     public static
     function Proses($peg)
     {
      if ($peg instanceof Manajer)
```

```
prak 5 (1).php > html > body > div.container
54      class Soal1
55      {
56       public static
57       function Proses($peg)
58       {
59        if ($peg instanceof Manajer)
60        {
61         $man = $peg;
62         echo "Nama manajer: ".$man->nama, "\n";
63         echo "<br>Tunjangan: Rp. ".strval($man->tunjangan), "\n";
64        }
65        else if ($peg instanceof Kurir)
66        {
67         $kur =  $peg;
68         echo "Nama kurir: ".$kur->nama, "\n";
69         echo "<br>Gaji: Rp. ".strval($kur->gaji), "\n";
70        }
71       }
72       public static
73       function main($args)
74       {
75        $peg1 = new Manajer("Dwi", 5000000);
76        Soal1::Proses($peg1);
77            echo "<br>", "<br>";
78        $peg2 = new Kurir("Sri", 200000);
79        Soal1::Proses($peg2);
80       }
81      }
82      Soal1::main(array());
83  ?>
```

- OUTPUT

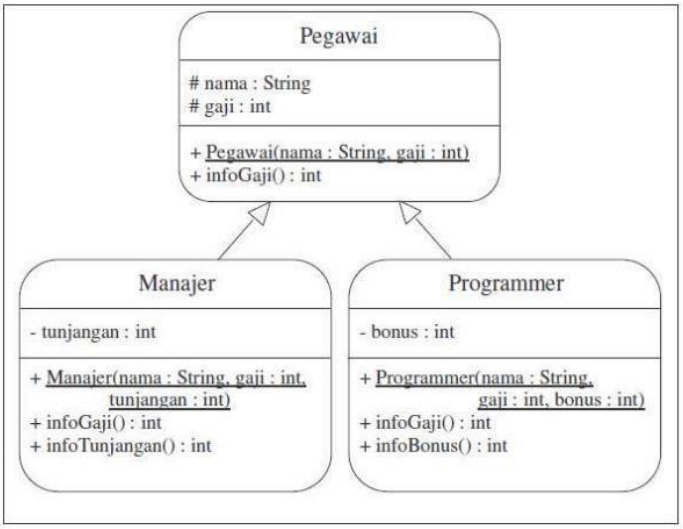
Nama manajer: Dwi
Tunjangan: Rp. 5000000

Nama kurir: Sri
Gaji: Rp. 200000

- Analisis

Method info dibuat static bertujuan agar pemanggilannya tidak perlu inisiasi. Sehingga dapat langsung dimasukan menjadi parameter pada method info di class info.

2. Buat program berdasarkan UML berikut.



- Code

prak 5(2).php > ...

```
<!-- Putri Dwinatryska
    21091397075 -->

<!DOCTYPE html>
<html lang="id">
<head>
    <link rel="stylesheet" type="text/css" href="no 2.css">
    <title>Praktikum 5</title>
</head>
<body>
    <div class="container">
        <h2><u>Soal No.2</u></h2>

        <?php
    class Pegawai
    {
     public  $nama;
     public $gaji;
     public function __construct($nama, $gaji)
     {
      $this->nama = $nama;
      $this->gaji = $gaji;
     }
     public function infoGaji()
     {
      return $this->gaji;
     }
    }
     class Manajer extends Pegawai
     {
```

```
prak 5(2).php > html > body > div.container > Programmer > infoBonus
31      private $tunjangan;
32      public function __construct($nama, $gaji, $tunjangan)
33      {
34       parent::__construct($nama, $gaji);
35       $this->tunjangan = $tunjangan;
36      }
37      public function infoGaji()
38      {
39       return $this->gaji;
40      }
41      public function infoTunjangan()
42      {
43       return $this->tunjangan;
44      }
45     }
46     class Programmer extends Pegawai
47     {
48      private $bonus;
49      public function __construct($nama, $gaji, $bonus)
50      {
51       parent::__construct($nama, $gaji);
52       $this->bonus = $bonus;
53      }
54      public function infoGaji()
55      {
56       return $this->gaji;
57      }
58      public function infoBonus()
59      {
60       return $this->bonus;
61     }
62     }
63     class Bayaran
64     {
65      public function hitungBayaran($peg)
66      {
67       $uang = $peg->infoGaji();
68
69       return $uang;
70      }
71      public static function main($args)
72      {
73       $man = new Manajer("Dwi", 7000000, 3000000);
74       $prog = new Programmer("Sri", 2000000, 1100000);
75       $hr = new Bayaran();
76       echo "<br> Nama Manajer : ".$man->nama."<br> Gaji : Rp. ".strval($hr->hitungBayaran($man)) , "\n";
77           echo "<br>";
78       echo "<br>Nama Programmer : ".$prog->nama. "<br> Gaji : Rp. ".strval($hr->hitungBayaran($prog)) , "\
79
80      }
81     }
82     Bayaran::main(array());
83    ?>
84      </div>
85   </body>
86   </html>
```

- Output

Nama Manajer : Dwi
Gaji : Rp. 7000000

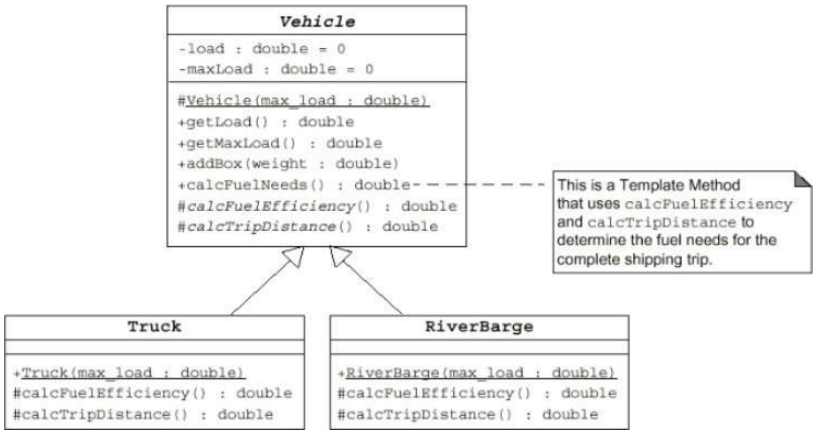
Nama Programmer : Sri
Gaji : Rp. 2000000

- Analisi

Program tersebut menerapkan inheritance dengan konsep overriding. Pemanggilan construct pada masing-masing class turunan hanya akan menginisiasi properti yang dimiliki dengan visibilitas private serta properti lain yang diturunkan akan langsung diinisiasi dengan construct dari parentnya.

Praktikum 6

1. Buat program berdasarkan UML berikut.



- Code

```
prak6 no1.php > html > body > div.container > div.row > div.col-5.mx-auto.border.p-3.mt-2
4       <?php
5       require_once 'prak6 no1.php';
6   ?>
7
8   <!DOCTYPE html>
9   <html lang="id">
10
11  <head>
12      <!-- Bootstrap CSS -->
13      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
14      integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anony
15
16      <title>Praktikum 6</title>
17  </head>
18
19  <body>
20      <div class="container">
21          <br>
22          <div class="row">
23              <div class="col-5 mx-auto border p-3 mt-2">
24                  <h4 class="text-center"><strong><u>Soal No.1</u></strong></h4>
25                  <br><br>
26                  <b><?= $truck->getMaxLoad() . ' kg'; ?> <br></b>
27                  <br>
28                  <?= $truck->addBox(2000) . ' kg'; ?> <br>
29                  <?= $truck->addBox(7000) . ' kg'; ?> <br>
30                  <?= $truck->addBox(9000) . ' kg'; ?> <br>
31
32                  <?php
33                      echo "Jadi, Butuh Bahan Bakar sebanyak " . $truck->calcFuelNeeds() . ' Liter'. '<b
```

```
34                    ?>
35                <br>
36                <hr>
37                <br>
38                <b><?= $riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
39                <br>
40                <?= $riverBarge->addBox(5000) . ' kg'; ?> <br>
41                <?= $riverBarge->addBox(7000) . ' kg'; ?> <br>
42                <?= $riverBarge->addBox(8000) . ' kg'; ?> <br>
43
44                    <?php
45                        echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge->calcFuelNeeds() . ' Liter'
46                    ?>
47            </div>
48        </div>
49    </div>
50</body>
51
52</html>
```

- Output

Maksimal muatan Truk 18000 kg

Truk menambah muatan sebesar 2000 kg
Truk menambah muatan sebesar 7000 kg
Truk menambah muatan sebesar 9000 kg
Jadi, Butuh Bahan Bakar sebanyak 6 Liter

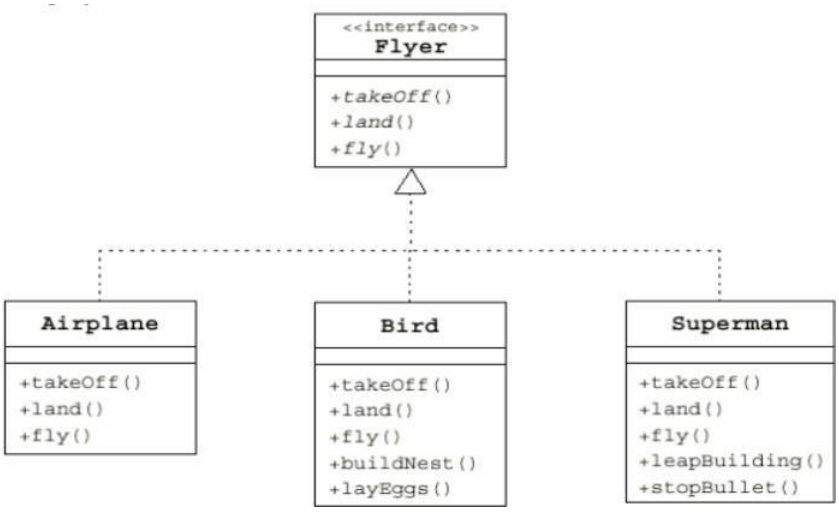
Maksimal muatan Tongkang Sungai 20000 kg

Tongkang Sungai menambah muatan sebesar 5000 kg
Tongkang Sungai menambah muatan sebesar 7000 kg
Tongkang Sungai menambah muatan sebesar 8000 kg
Jadi, Butuh Bahan Bakar sebanyak 5 Liter

- Analisis

Program ini 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



- Code

prak6 no2.php > ...

```php
<?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;
```

- Output

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

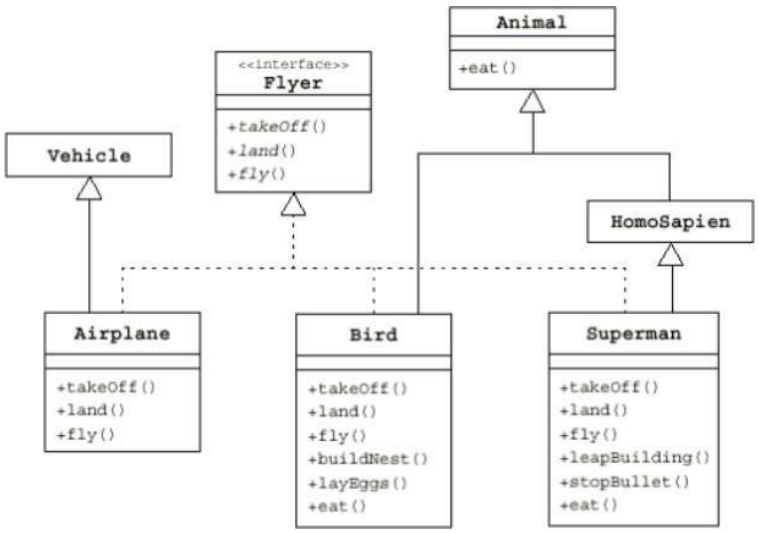
Bird
Burung membuat sarang
Burung mencari makan
Burung terbang
Burung kembali pulang
Burung bertelur

Airplane
Pesawat lepas landas..
Pesawat dalam perjalanan
Pesawat mendarat

- Analisis

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

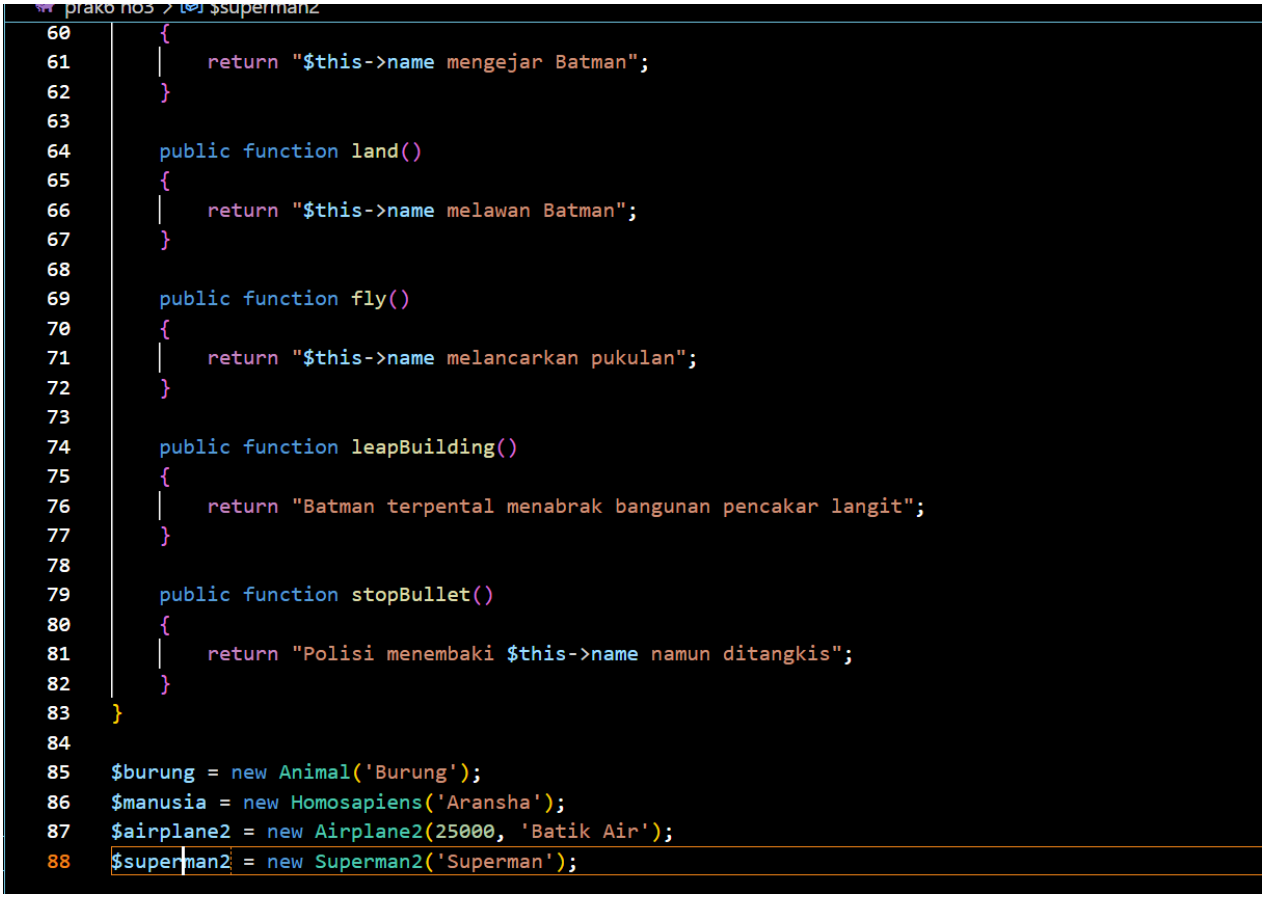
3. Buh



- Code

```php
<?php

require_once 'abstrac 3.php';
require_once 'interface 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;
    }
```

```php
    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()
    {
```



```
60      {
61          return "$this->name mengejar Batman";
62      }
63
64      public function land()
65      {
66          return "$this->name melawan Batman";
67      }
68
69      public function fly()
70      {
71          return "$this->name melancarkan pukulan";
72      }
73
74      public function leapBuilding()
75      {
76          return "Batman terpental menabrak bangunan pencakar langit";
77      }
78
79      public function stopBullet()
80      {
81          return "Polisi menembaki $this->name namun ditangkis";
82      }
83  }
84
85  $burung = new Animal('Burung');
86  $manusia = new Homosapiens('Aransha');
87  $airplane2 = new Airplane2(25000, 'Batik Air');
88  $superman2 = new Superman2('Superman');
```

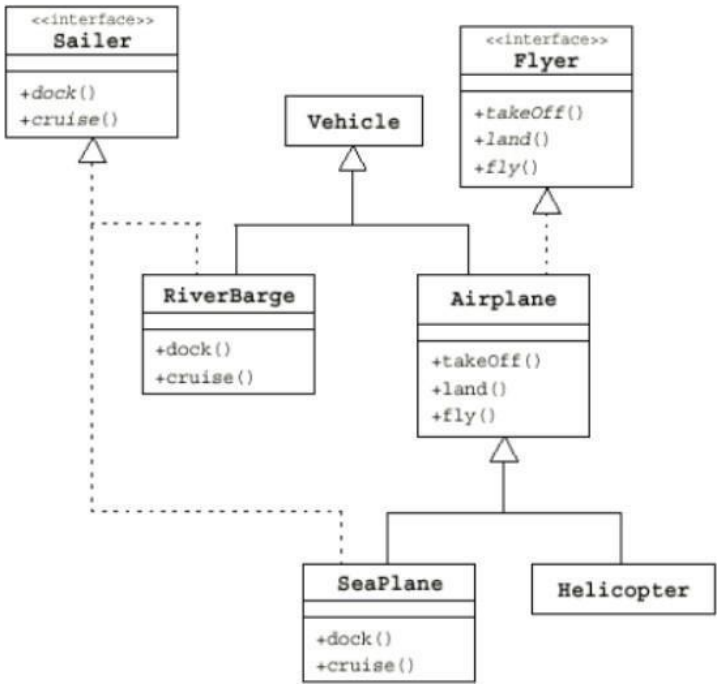
- Output

```
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. Hgh



➢ Code

prak6 no4 > SeaPlane

```php
<?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() {
```

```
90      public function land() {
91          return $this->name . ' mendarat';
92      }
93
94      public function fly() {
95          return $this->name . ' dalam perjalanan';
96      }
97  }
98
99  $riverBarge2 = new RiverBarge2(40000, 'Atomic');
100 $seaPlane = new SeaPlane(30000, 'Titanic');
101 $helicopter = new Helicopter(15000, 'Brocklyn');
```

- Output

Maksimal muatan Atomic 40000 kg
Atomic menambah muatan sebesar 15000 kg
Atomic menambah muatan sebesar 10000 kg
Atomic menambah muatan sebesar 8000 kg
Atomic menambah muatan sebesar 2000 kg
Atomic berada di dermaga
Atomic sedang berlayar
Jadi, Butuh Bahan Bakar sebanyak 3 Liter

Maksimal muatan Titanic 30000 kg
Titanic menambah muatan sebesar 15000 kg
Titanic menambah muatan sebesar 7000 kg
Titanic berada di dermaga
Titanic sedang berlayar
Titanic lepas landas
Titanic dalam perjalanan
Titanic mendarat
Jadi, Butuh Bahan Bakar sebanyak 5 Liter

Maksimal muatan Brocklyn 15000 kg
Brocklyn menambah muatan sebesar 5000 kg
Brocklyn menambah muatan sebesar 7000 kg
Brocklyn lepas landas
Brocklyn dalam perjalanan
Brocklyn mendarat
Jadi, Butuh Bahan Bakar sebanyak 9 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.