

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

```
1 <!-- Affandika --->
2 <!-- 21091397030 --->
3
4 <?php
5
6 require_once 'Prak 6 Nomor 1 abstrak.php';
7
8 class Truck extends Vehicle {
9     public function __construct($maxLoad, $name)
10     {
11         $this->maxLoad = $maxLoad;
12         $this->name = $name;
13     }
14
15     public function calcFuelNeeds()
16     {
17         $fuel = $this->calcFuelEfficiency();
18         $trip = $this->calcTripDistance();
19
20         return ceil($fuel /= $trip);
21     }
22 }
23
24 class RiverBarge extends Vehicle {
25     public function __construct($maxLoad, $name)
26     {
27         $this->maxLoad = $maxLoad;
28         $this->name = $name;
29     }
30
31     public function calcFuelNeeds()
32     {
33         $fuel = $this->calcFuelEfficiency();
34         $trip = $this->calcTripDistance();
35
36         return ceil($fuel /= $trip);
37     }
38 }
39 $truck = new Truck(20000, 'Truk');
40 $riverBarge = new RiverBarge(17000, 'Perahu');
```

Abstract

```
1 <!-- Affandika --->
2 <!-- 21091397030 --->
3
4 <?php
5
6 abstract class Vehicle {
7     private $load = 0;
8     protected $maxLoad = 0, $name;
9
10     protected function __construct($maxLoad, $name) {
11         $this->$maxLoad = $maxLoad;
12         $this->$name = $name;
13     }
14
15     public function getLoad() {
16         return $this->load;
17     }
18
19     public function getMaxLoad() {
20         echo 'Maksimal muatan ' . $this->name . ' ' ;
21         return $this->maxLoad;
22     }
23
24     public function addBox($weight) {
25         if ($this->load >= $this->maxLoad) {
26             echo "$this->name ditambahkan muatan sebesar $weight <br>";
27         }
28     }
29
30     public function addBox($weight) {
31         if ($this->load >= $this->maxLoad) {
32             echo "$this->name ditambahkan muatan sebesar $weight <br>";
33             echo 'Muatan telah penuh tidak bisa ditambahkan muatan lagi';
34         } else {
35             $this->load += $weight;
36             echo "$this->name ditambahkan muatan sebesar $weight";
37         }
38     }
39
40     abstract public function calcFuelNeeds();
41
42     protected function calcFuelEfficiency() {
43         $range = 60000000;
44         $range /= $this->load;
45         return $range;
46     }
47
48     protected function calcTripDistance() {
49         return 500;
50     }
51 }
```

Index

```
1 <!-- Affandika -->
2 <!-- 21091397030 -->
3
4 <?php
5     require_once 'Prak 6 Nomor 1.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="sha384-1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrfhtjDbrCEXSU1oBoqy12QvZ6jIW3" crossorigin="anonymous">
15
16     <title>Praktikum 6</title>
17 </head>
18
19 <body>
20     <div class="container">
21         <br>
22         <h2 class="text-center">Praktikum 6 PBO</h2>
23         <div class="row">
24             <div class="col-5 mx-auto border p-3 mt-2">
25                 <h4 class="text-center"><strong>Nomor 1</strong></h4>
26                 <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 X
Prak 6 Nomor 1 index.php > ...
27 <b><?= $truck->getMaxLoad() . ' kg'; ?> <br></b>
28 <br>
29 <?= $truck->addBox(2000) . ' kg'; ?> <br>
30 <?= $truck->addBox(8000) . ' kg'; ?> <br>
31 <?= $truck->addBox(10000) . ' kg'; ?> <br>
32
33     <?php
34         echo "Banyak bahan bakar yang dibutuhkan adalah " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
35     ?>
36 <br>
37 -----
38 <br>
39 <br>
40 <b><?= $riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
41 <br>
42 <?= $riverBarge->addBox(2000) . ' kg'; ?> <br>
43 <?= $riverBarge->addBox(5000) . ' kg'; ?> <br>
44 <?= $riverBarge->addBox(10000) . ' kg'; ?> <br>
45
46     <?php
47         echo "Banyak bahan bakar yang dibutuhkan adalah " . $riverBarge->calcFuelNeeds() . ' Liter';
48     ?>
49 </div>
50 </div>
51 </div>
52 </body>
```

Output:

Praktikum 6 PBO

Nomor 1

Maksimal muatan Truk 20000 kg

Truk ditambahkan muatan sebesar 2000 kg
Truk ditambahkan muatan sebesar 8000 kg
Truk ditambahkan muatan sebesar 10000 kg
Banyak bahan bakar yang dibutuhkan adalah 6 Liter

Maksimal muatan Perahu 17000 kg

Perahu ditambahkan muatan sebesar 2000 kg
Perahu ditambahkan muatan sebesar 5000 kg
Perahu ditambahkan muatan sebesar 10000 kg
Banyak bahan bakar yang dibutuhkan adalah 8 Liter

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

```
1 <!-- Affandika -->
2 <!-- 21091397030 -->
3
4 <?php
5
6 require_once 'Prak 6 Nomor 2 Interface.php';
7
8 class Airplane implements Flyer {
9     public function takeOff() {
10         return 'Pesawat lepas landas...';
11     }
12
13     public function land() {
14         return 'Pesawat mendarat';
15     }
16
17     public function fly() {
18         return 'Pesawat dalam perjalanan';
19     }
20 }
21
22 class Bird implements Flyer {
23     public function takeOff() {
24         return 'Burung mencari makan';
25     }
26
27     public function land() {
28         return 'Burung kembali pulang';
29     }
30
31     public function fly() {
32         return 'Burung terbang';
33     }
34
35     public function buildNest() {
36         return 'Burung membuat sarang';
37     }
38
39     public function layEggs() {
40         return 'Burung bertelur';
41     }
42 }
43
44 class Superman implements Flyer {
45     public function takeOff() {
46         return 'Superman mengejar Steppenwolf';
47     }
48
49     public function land() {
50         return 'Superman melawan Steppenwolf';
51     }
52 }
```

```
Go Run Terminal Help      Prak 6 Nomor 2.php - tugas pbo - Visual Studio Code
Prak 6 Nomor 2.php X      Prak 6 Nomor 2 Interface.php      Prak 6 Nomor 2 index.php
Prak 6 Nomor 2.php > PHP Intelephense > Superman > leapBuilding
44 class Superman implements Flyer {
45     public function takeOff() {
46         return 'Superman mengejar Steppenwolf';
47     }
48
49     public function land() {
50         return 'Superman melawan Steppenwolf';
51     }
52
53     public function fly() {
54         return 'Superman menuju matahari';
55     }
56
57     public function leapBuilding() {
58         return 'Steppenwolf terpesong menabrak bangunan pencakar langit';
59     }
60
61     public function stopBullet() {
62         return 'Polisi menembaki superman namun tidak tembus';
63     }
64 }
65
66 $airplane = new Airplane;
67 $bird = new Bird;
68 $superman = new Superman;
```

Interface

```
1 <!-- Affandika --->
2 <!-- 21091397030 --->
3
4 <?php
5
6 interface Flyer {
7     public function takeOff();
8     public function land();
9     public function fly();
10 }
11
12 interface Sailer {
13     public function dock();
14     public function cruise();
15 }
```

Index

```
1 <!-- Affandika --->
2 <!-- 21091397030 --->
3
4 <?php
5     require_once 'Prak 6 Nomor 2.php';
6 ?>
7
8 <!DOCTYPE html>
9 <html lang="en">
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="sha384-1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrfhtjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
15
16     <title>Praktikum 6</title>
17 </head>
18
19 <body>
20     <div class="container">
21         <br>
22         <h2 class="text-center">Praktikum 6 PBO</h2>
23         <div class="row">
24             <div class="col-5 mx-auto border p-3 mt-2">
25                 <h4 class="text-center"><strong><u>Nomor 2</u></strong></h4>
26                 <br><br>
```

```

26         <br><br>
27         <b><?php
28             echo "Superman";
29             ?></b> <br>
30         <?= $superman->land(); ?> <br>
31         <?= $superman->takeOff(); ?> <br>
32         <?= $superman->fly(); ?> <br>
33         <?= $superman->leapBuilding(); ?> <br>
34         <?= $superman->stopBullet(); ?> <br>
35         <br>
36         <b><?php
37             echo "Bird";
38             ?></b> <br>
39         <?= $bird->buildNest(); ?> <br>
40         <?= $bird->takeOff(); ?> <br>
41         <?= $bird->fly(); ?> <br>
42         <?= $bird->land(); ?> <br>
43         <?= $bird->layEggs(); ?> <br>
44         <br>
45         <b><?php
46             echo "Airplane";
47             ?></b> <br>
48         <?= $airplane->takeOff(); ?> <br>
49         <?= $airplane->fly(); ?> <br>
50         <?= $airplane->land(); ?> <br>
51         </div>

```

Output:

Praktikum 6 PBO

Nomor 2

Superman

Superman melawan Steppenwolf
 Superman mengejar Steppenwolf
 Superman menuju matahari
 Steppenwolf terpanah 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 Polymorphism dengan penggunaan Interface Flyer sehingga semua class yang Implements dari interface Flyer harus memiliki method takeoff, land, dan fly.

3. Source Code

PHP

```
Prak 6 Nomor 3.php > PHP Intelephense > Animal
1 <!-- Affandika -->
2 <!-- 21091397030 -->
3
4 <?php
5
6 require_once 'Prak 6 Nomor 3 Abstrac.php';
7 require_once 'Prak 6 Nomor 3 Interface.php';
8
9 class Animal
10 {
11     protected $name;
12
13     public function __construct($name)
14     {
15         $this->name = $name;
16     }
17
18     public function eat()
19     {
20         return $this->name . ' sedang makan';
21     }
22 }
23
24 class Homosapiens extends Animal {}
25
26 class Airplane2 extends Vehicle implements Flyer
27 {
28     public function __construct($maxLoad, $name)
29     {
30         $this->maxLoad = $maxLoad;
31         $this->name = $name;
32     }
33
34     public function takeOff()
35     {
36         return $this->name lepas landas";
37     }
38
39     public function land()
40     {
41         return $this->name mendarat";
42     }
43
44     public function fly()
45     {
46         return $this->name dalam perjalanan";
47     }
48 }
49
50 public function calcFuelNeeds()
51 {
52     $fuel = $this->calcFuelEfficiency();
53     $trip = $this->calcTripDistance();
54
55     return ceil($fuel / $trip);
56 }
57
58 }
59
60 class Superman2 extends Homosapiens implements Flyer
61 {
62     public function takeOff()
63     {
64         return $this->name mengejar Steppenwolf";
65     }
66
67     public function land()
68     {
69         return $this->name melawan Steppenwolf";
70     }
71
72     public function fly()
73
74     public function fly()
75     {
76         return $this->name melakukan pukulan";
77     }
78
79     public function leapBuilding()
80     {
81         return "Steppenwolf terpentak menabrak bangunan pencakar langit";
82     }
83
84     public function stopBullet()
85     {
86         return "Polisi menembaki $this->name namun tidak tembus";
87     }
88 }
89
90 $burung = new Animal('Burung');
91 $manusia = new Homosapiens('Aransha');
92 $airplane2 = new Airplane2(27000, 'Lion Air');
93 $superman2 = new Superman2('Superman');
```

Interface

```
1 <!-- Affandika -->
2 <!-- 21091397030 -->
3
4 <?php
5
6 interface Flyer {
7     public function takeOff();
8     public function land();
9     public function fly();
10 }
11
12 interface Sailer {
13     public function dock();
14     public function cruise();
15 }
```

Abstract

```
Prak 6 Nomor 3 Abstract.php > ...
1 <!-- Affandika --->
2 <!-- 21091397030 --->
3
4 <?php
5
6 abstract class Vehicle {
7     private $load = 0;
8     protected $maxLoad = 0, $name;
9
10    protected function __construct($maxLoad, $name) {
11        $this->$maxLoad = $maxLoad;
12        $this->$name = $name;
13    }
14
15    public function getLoad() {
16        return $this->load;
17    }
18
19    public function getMaxLoad() {
20        echo 'Maksimal muatan ' . $this->name . ' ' ;
21        return $this->maxLoad;
22    }
23
24    public function addBox($weight) {
25        if ($this->load >= $this->maxLoad) {
26            echo "$this->name ditambahkan muatan sebesar $weight <br>";
27
28            public function addBox($weight) {
29                if ($this->load >= $this->maxLoad) {
30                    echo "$this->name ditambahkan muatan sebesar $weight <br>";
31                    echo 'Muatan telah penuh tidak bisa menambah lagi';
32                }else {
33                    $this->load += $weight;
34                    echo "$this->name ditambahkan muatan sebesar $weight";
35                }
36            }
37
38            abstract public function calcFuelNeeds();
39
40            protected function calcFuelEfficiency() {
41                $range = 70000000;
42                $range /= $this->load;
43                return $range;
44            }
45
46            protected function calcTripDistance() {
47                return 500;
48            }
49        }
50    }
```

Index

```
1 <!-- Affandika --->
2 <!-- 21091397030 --->
3
4 <?php
5     require_once 'Prak 6 Nomor 3.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="sha384-18mE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrfHfjDbrCEXSU1oBoqy12QvZ6jIW3" crossorigin="anonymous">
15
16     <title>Praktikum 6</title>
17 </head>
18 <body>
19     <div class="container">
20         <h2 class="text-center">Praktikum 6 PBO</h2>
21         <div class="row">
22             <div class="col-5 mx-auto border p-3 mt-2">
23                 <h4 class="text-center"><strong><u>Nomor 3</u></strong></h4>
24                 <br><br>
25                 <?= $burung->eat(); ?> <br>
26                 <?= $manusia->eat(); ?> <br>

```



```

27 <br>
28
29 <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
30 <?= $airplane2->addBox(5000) . ' kg'; ?> <br>
31 <?= $airplane2->addBox(3000) . ' kg'; ?> <br>
32 <?= $airplane2->addBox(8000) . ' kg'; ?> <br>
33 <?= $airplane2->addBox(9000) . ' kg'; ?> <br>
34 <?= $airplane2->takeOff(); ?> <br>
35 <?= $airplane2->fly(); ?> <br>
36 <?= $airplane2->land(); ?> <br>
37
38 <?php
39     echo "Banyak bahan bakar yang dibutuhkan adalah " . $airplane2->calcFuelNeeds() . ' Liter'. '<br>';
40     ?>
41 <br>
42 <?= $superman2->eat(); ?> <br>
43 <?= $superman2->land(); ?> <br>
44 <?= $superman2->takeOff(); ?> <br>
45 <?= $superman2->fly(); ?> <br>
46 <?= $superman2->leapBuilding(); ?> <br>
47 <?= $superman2->stopBullet(); ?> <br>
48 </div>
49 </div>
50 </div>
51 </body>
52 </html>

```

Output:

Praktikum 6 PBO

Nomor 3

Burung sedang makan
Aransha sedang makan

Maksimal muatan Lion Air 27000 kg

Lion Air ditambahkan muatan sebesar 5000 kg
Lion Air ditambahkan muatan sebesar 3000 kg
Lion Air ditambahkan muatan sebesar 8000 kg
Lion Air ditambahkan muatan sebesar 9000 kg
Lion Air lepas landas
Lion Air dalam perjalanan
Lion Air mendarat
Banyak bahan bakar yang dibutuhkan adalah 6 Liter

Superman sedang makan
Superman melawan Steppenwolf
Superman mengejar Steppenwolf
Superman melakukan pukulan
Steppenwolf terpental menabrak bangunan pencakar langit
Polisi menembaki Superman namun tidak tembus

Analisis:

Terdapat interface Flyer dan abstract 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

```
1 <!-- Affandika -->
2 <!-- 21091397030 -->
3
4 <?php
5
6 require_once 'Prak 6 Nomor 4 Abstract.php';
7 require_once 'Prak 6 Nomor 4 Interface.php';
8
9 class RiverBarge2 extends Vehicle implements Sailer {
10     public function __construct($maxLoad, $name) {
11         $this->maxLoad = $maxLoad;
12         $this->name = $name;
13     }
14
15     public function calcFuelNeeds() {
16         $fuel = $this->calcFuelEfficiency();
17         $trip = $this->calcTripDistance();
18
19         return ceil($fuel / $trip);
20     }
21
22     public function dock() {
23         return $this->name . ' berada di dermaga';
24     }
25
26     public function cruise() {
27         return $this->name . ' sedang berlayar';
28     }
29 }
30
31 class Airplane2 implements Flyer {
32     public function takeOff() {
33         return 'Pesawat lepas landas';
34     }
35     public function land() {
36         return 'Pesawat mendarat';
37     }
38     public function fly() {
39         return 'Pesawat dalam perjalanan';
40     }
41 }
42
43 class SeaPlane extends Vehicle implements Sailer {
44     public function __construct($maxLoad, $name) {
45         $this->maxLoad = $maxLoad;
46         $this->name = $name;
47     }
48
49     public function calcFuelNeeds() {
50         $fuel = $this->calcFuelEfficiency();
51         $trip = $this->calcTripDistance();
52
53         return ceil($fuel / $trip);
54     }
55
56     public function dock() {
57         return $this->name . ' berada di dermaga';
58     }
59
60     public function cruise() {
61         return $this->name . ' sedang berlayar';
62     }
63
64     public function takeOff() {
65         return $this->name . ' lepas landas';
66     }
67
68     public function land() {
69         return $this->name . ' mendarat';
70     }
71
72     public function fly() {
73         return $this->name . ' dalam perjalanan';
74     }
75
76     public function calcFuelNeeds() {
77         $fuel = $this->calcFuelEfficiency();
78         $trip = $this->calcTripDistance();
79
80         return ceil($fuel / $trip);
81     }
82
83     public function dock() {
84         return $this->name . ' berada di dermaga';
85     }
86
87     public function cruise() {
88         return $this->name . ' sedang berlayar';
89     }
90
91     public function takeOff() {
92         return $this->name . ' lepas landas';
93     }
94
95     public function land() {
96         return $this->name . ' mendarat';
97     }
98
99     public function fly() {
100         return $this->name . ' dalam perjalanan';
101     }
102 }
103
104 $riverBarge2 = new RiverBarge2(31000, 'Atomic');
105 $seaPlane = new SeaPlane(25000, 'Titanic');
106 $helicopter = new Helicopter(13000, 'Brocklyn');
```

Interface

```
1 <!-- Affandika --->
2 <!-- 21091397030 --->
3
4 <?php
5
6 interface Flyer {
7     public function takeOff();
8     public function land();
9     public function fly();
10 }
11
12 interface Sailer {
13     public function dock();
14     public function cruise();
15 }
```

Abstract

```
1 <!-- Affandika --->
2 <!-- 21091397030 --->
3
4 <?php
5
6 abstract class Vehicle {
7     private $load = 0;
8     protected $maxLoad = 0, $name;
9
10     protected function __construct($maxLoad, $name) {
11         $this->$maxLoad = $maxLoad;
12         $this->$name = $name;
13     }
14
15     public function getload() {
16         return $this->load;
17     }
18
19     public function getMaxLoad() {
20         echo 'Maksimal muatan ' . $this->name . ' ' ;
21         return $this->maxLoad;
22     }
23
24     public function addBox($weight) {
25         if ($this->load >= $this->maxLoad) {
26             echo "$this->name menambah muatan sebesar $weight <br>";
27
28         public function addBox($weight) {
29             if ($this->load >= $this->maxLoad) {
30                 echo "$this->name menambah muatan sebesar $weight <br>";
31                 echo 'Muatan telah penuh tidak bisa menambah lagi';
32             }else {
33                 $this->load += $weight;
34                 echo "$this->name menambah muatan sebesar $weight";
35             }
36         }
37
38         abstract public function calcFuelNeeds();
39
40         protected function calcFuelEfficiency() {
41             $range = 70000000;
42             $range /= $this->load;
43             return $range;
44         }
45
46         protected function calcTripDistance() {
47             return 500;
48         }
49     }
50 }
```

Index

```

1 <!-- Affandika -->
2 <!-- 21091397030 -->
3
4 <?php
5     require_once 'Prak 6 Nomor 4.php';
6 ?>
7
8 <!DOCTYPE html>
9 <html lang="en">
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="sha384-18mE4kWBq78iYhF1dVkuhFTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
15         crossorigin="anonymous">
16
17     <title>PBO - Praktikum 6</title>
18 </head>
19 <body>
20     <h2 class="text-center">Praktikum 6 PBO</h2>
21     <div class="container">
22         <div class="row">
23             <h4 class="text-center"><strong>Nomor 4</strong></h4>
24             <br><br>
25             <div class="col-4 mx-auto border p-2 mt-2">
26                 <b><?=$riverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
27                 <?=$riverBarge2->addBox(10000) . ' kg'; ?> <br>
28                 <b><?=$riverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
29                 <?=$riverBarge2->addBox(10000) . ' kg'; ?> <br>
30                 <?=$riverBarge2->addBox(15000) . ' kg'; ?> <br>
31                 <?=$riverBarge2->addBox(2000) . ' kg'; ?> <br>
32                 <?=$riverBarge2->addBox(4000) . ' kg'; ?> <br>
33                 <?=$riverBarge2->dock(); ?> <br>
34                 <?=$riverBarge2->cruise(); ?> <br>
35                 <?php
36                     echo "Banyak bahan bakar yang dibutuhkan " . $riverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
37                 ?>
38             </div>
39             <div class="col-4 mx-auto border p-2 mt-2">
40                 <b><?=$seaPlane->getMaxLoad() . ' kg'; ?> <br></b>
41                 <?=$seaPlane->addBox(16000) . ' kg'; ?> <br>
42                 <?=$seaPlane->addBox(9000) . ' kg'; ?> <br>
43                 <?=$seaPlane->dock(); ?> <br>
44                 <?=$seaPlane->cruise(); ?> <br>
45                 <?=$seaPlane->takeOff(); ?> <br>
46                 <?=$seaPlane->fly(); ?> <br>
47                 <?=$seaPlane->land(); ?> <br>
48                 <?php
49                     echo "Banyak bahan bakar yang dibutuhkan " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
50                 ?>
51             </div>
52             <div class="col mx-auto border p-2 mt-2">
53                 <?=$seaPlane->fly(); ?> <br>
54                 <?=$seaPlane->land(); ?> <br>
55                 <?php
56                     echo "Banyak bahan bakar yang dibutuhkan " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
57                 ?>
58             </div>
59             <div class="col mx-auto border p-2 mt-2">
60                 <b><?=$helicopter->getMaxLoad() . ' kg'; ?> <br></b>
61                 <?=$helicopter->addBox(7000) . ' kg'; ?> <br>
62                 <?=$helicopter->addBox(6000) . ' kg'; ?> <br>
63                 <?=$helicopter->takeOff(); ?> <br>
64                 <?=$helicopter->fly(); ?> <br>
65                 <?=$helicopter->land(); ?> <br>
66                 <?php
67                     echo "Banyak bahan bakar yang dibutuhkan " . $helicopter->calcFuelNeeds() . ' Liter'. '<br>';
68                 ?>
69             </div>
70         </div>
71     </div>
72 </body>
73
74 </html>

```

Output:

Praktikum 6 PBO
Nomor 4

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

Analisis:

Implementasi polymorphism dengan interface dan abstract class ditunjukkan 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