

**LAPORAN PEMROGRAMAN BERORIENTASI OBJEK  
UJIAN TENGAH SEMESTER**



Disusun oleh :

Diah Ayuning Tyas (21091397013)  
A2021 MI

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA  
FAKULTAS VOKASI  
UNIVERSITAS NEGERI SURABAYA**

**2022**

## 1. Buat program berdasarkan UML berikut

- Kodingan :

### 1. Abstrak

```
C: > Users > user > Downloads > 013_Diah Ayuning Tyas_No.1 Abstract_Praktikum 6.php > ...
1  <!-- Diah Ayuning Tyas - 21091397013 -->
2
3  <?php
4
5  1 reference | 1 implementation
   abstract class Vehicle {
6      4 references
7      private $load = 0;
8      protected $maxLoad = 0, $name;
9
10     0 references | 1 override
       protected function __construct($maxLoad, $name) {
11         $this->$maxLoad = $maxLoad;
12         $this->$name = $name;
13     }
14
15     0 references | 0 overrides
       public function getLoad() {
16         return $this->load;
17     }
18
19     0 references | 0 overrides
       public function getMaxLoad() {
20         echo 'Maksimal muatan ' . $this->name . ' ' ;
21         return $this->maxLoad;
22     }
23
24     0 references | 0 overrides
       public function addBox($weight) {
25         if ($this->load >= $this->maxLoad) {
26             echo "$this->name menambah muatan sebesar $weight <br>";
27             echo 'Muatan telah penuh tidak bisa menambah lagi';
28         }else {
29             $this->load += $weight;
30             echo "$this->name menambah muatan sebesar $weight";
31         }
32     }
33
34     0 references | 1 override
       abstract public function calcFuelNeeds();
35
36     1 reference | 0 overrides
       protected function calcFuelEfficiency() {
37         $range = 50000000;
38         $range /= $this->load;
39         return $range;
40     }
41
42     1 reference | 0 overrides
       protected function calcTripDistance() {
43         return 500;
44     }
45 }
```

### 2. Indexs

```

C:\> Users\user> Downloads> 013_Diah Ayuning Tyas_No.1 Index_Praktikum 6.php
1  <!-- Diah Ayuning Tyas - 21091397013 -->
2
3  <?php
4      require_once 'no1.php';
5  ?>
6
7  <!DOCTYPE html>
8  <html lang="id">
9
10     <head>
11         <!-- Bootstrap CSS -->
12         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
13             integrity="sha384-18mE4kWBq78iYhFldvKuHfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
14
15         <title>PBO - Praktikum 6</title>
16     </head>
17
18     <body>
19         <div class="container">
20             <br>
21             <h2 class="text-center">PBO - Praktikum 6</h2>
22             <div class="row">
23                 <div class="col-5 mx-auto border p-3 mt-2">
24                     <h4 class="text-center"><strong>Soal 1</strong></h4>
25                     <br><br>
26                     <b><?=$truck->getMaxLoad() . ' kg'; ?> <br></b>
27                     <br>
28                     <?=$truck->addBox(4000) . ' kg'; ?> <br>
29                     <?=$truck->addBox(2000) . ' kg'; ?> <br>
30                     <?=$truck->addBox(8000) . ' kg'; ?> <br>
31
32                     <?php
33                         echo "Jadi, Butuh Bahan Bakar sebanyak " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
34                     ?>
35                     <br>
36                     -----
37                     <br>
38                     <br>
39                     <b><?=$riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
40                     <br>
41                     <?=$riverBarge->addBox(2000) . ' kg'; ?> <br>
42                     <?=$riverBarge->addBox(4000) . ' kg'; ?> <br>
43                     <?=$riverBarge->addBox(10000) . ' kg'; ?> <br>
44
45                     <?php
46                         echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge->calcFuelNeeds() . ' Liter';
47                     ?>
48                 </div>
49             </div>
50         </div>
51     </body>
52 </html>

```

### 3. PHP

```

1  <!-- Diah Ayuning Tyas - 21091397013 -->
2
3  <?php
4
5  require_once '../No1/abstract.php';
6
7  1 reference | 0 implementations
8  class Truck extends Vehicle {
9      1 reference | 0 overrides | prototype
10     public function __construct($maxLoad, $name)
11     {
12         $this->maxLoad = $maxLoad;
13         $this->name = $name;
14     }
15
16     5 references | 0 overrides | prototype
17     public function calcFuelNeeds()
18     {
19         $fuel = $this->calcFuelEfficiency();
20         $trip = $this->calcTripDistance();
21         return ceil($fuel / $trip);
22     }
23 }
24
25 1 reference | 0 implementations
26 class RiverBarge extends Vehicle {
27     1 reference | 0 overrides | prototype
28     public function __construct($maxLoad, $name)
29     {
30         $this->maxLoad = $maxLoad;
31         $this->name = $name;
32     }
33
34     5 references | 0 overrides | prototype
35     public function calcFuelNeeds()
36     {
37         $fuel = $this->calcFuelEfficiency();
38         $trip = $this->calcTripDistance();
39         return ceil($fuel / $trip);
40     }
41 }
42
43 $truck = new Truck(15000, 'Truk');
44 $riverBarge = new RiverBarge(20000, 'Perahu');

```

- Output :



- 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 dan calcTripDistance

## 2. Buat program berdasarkan UML berikut

- Kodingan :

### 1. index

```
C:\Users\user > Downloads > 013_Diah Ayuning Tyas_No.2 Index_Praktikum 6.php
1  <!-- Diah Ayuning Tyas - 21091397013 -->
2
3  <?php
4      require_once 'no2.php';
5  ?>
6
7  <!DOCTYPE html>
8  <html lang="en">
9
10 <head>
11     <!-- Bootstrap CSS -->
12     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
13         integrity="sha384-1BmE4KWBq78iYhF1dvKuhfTAU6auU8tT94wrHffTjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
14
15     <title>PBO - Praktikum 6</title>
16 </head>
17
18 <body>
19     <div class="container">
20         <br>
21         <h2 class="text-center">PBO - Praktikum 6</h2>
22         <div class="row">
23             <div class="col-5 mx-auto border p-3 mt-2">
24                 <h4 class="text-center"><strong>Soal 2</strong></h4>
25                 <br><br>
26                 <b><?php
27                     echo "Superman";
28                 ?></b><br>
29                 <?= $superman->land(); ?><br>
30                 <?= $superman->takeOff(); ?><br>
31                 <?= $superman->fly(); ?><br>
32                 <?= $superman->leapBuilding(); ?><br>
33                 <?= $superman->stopBullet(); ?><br>
34                 <br>
35                 <b><?php
36                     echo "Bird";
37                 ?></b><br>
38                 <?= $bird->buildNest(); ?><br>
39                 <?= $bird->takeOff(); ?><br>
40                 <?= $bird->fly(); ?><br>
41                 <?= $bird->land(); ?><br>
42                 <?= $bird->layEggs(); ?><br>
43                 <br>
44                 <b><?php
45                     echo "Airplane";
46                 ?></b><br>
47                 <?= $airplane->takeOff(); ?><br>
48                 <?= $airplane->fly(); ?><br>
49                 <?= $airplane->land(); ?><br>
50             </div>
51         </div>
52     </div>
53 </body>
54
55 </html>
```

### 2. PHP

C:\> Users > user > Downloads > 013\_Diah Ayuning Tyas\_No.2 PHP\_Praktikum 6.php > ...

```
1 <!-- Diah Ayuning Tyas - 21091397013 -->
```

```
2
```

```
3
```

```
4 <?php
```

```
5
```

```
6 require_once '../No2/interface.php';
```

```
7
```

```
1 reference | 0 implementations
```

```
8 class Airplane implements Flyer {
```

```
5 references | 0 overrides
```

```
9     public function takeOff() {
```

```
10         return 'Pesawat lepas landas..';
```

```
11     }
```

```
12
```

```
5 references | 0 overrides
```

```
13     public function land() {
```

```
14         return 'Pesawat mendarat';
```

```
15     }
```

```
16
```

```
5 references | 0 overrides
```

```
17     public function fly() {
```

```
18         return 'Pesawat dalam perjalanan';
```

```
19     }
```

```
20 }
```

```
21
```

```
1 reference | 0 implementations
```

```
22 class Bird implements Flyer {
```

```
5 references | 0 overrides
```

```
23     public function takeOff() {
```

```
24         return 'Burung mencari makan';
```

```
25     }
```

```
26
```

```
5 references | 0 overrides
```

```
27     public function land() {
```

```
28         return 'Burung kembali pulang';
```

```
29     }
```

```
30
```

```
3 references | 0 overrides
```

```
31     public function fly() {
```

```
32         return 'Burung terbang';
```

```
33     }
```

```
34
```

```
1 reference | 0 overrides
```

```
35     public function buildNest() {
```

```
36         return 'Burung membuat sarang';
```

```
37     }
```

```
38
```

```
1 reference | 0 overrides
```

```
39     public function layEggs() {
```

```
40         return 'Burung bertelur';
```

```
41     }
```

```
42 }
```

```
43
```

```
1 reference | 0 implementations
```

```
44 class Superman implements Flyer {
```

```
3 references | 0 overrides
```

```
45     public function takeOff() {
```

```
46         return 'Superman mengejar Batman';
```

```
47     }
```

```
48
```

```
3 references | 0 overrides
```

```
49     public function land() {
```

```
50         return 'Superman melawan Batman';
```

```
51     }
```

```
52
```

```
3 references | 0 overrides
```

```
53     public function fly() {
```

```
54         return 'Superman melancarkan pukulan';
```

```
55     }
```

```
56
```

```
1 reference | 0 overrides
```

```
57     public function leapBuilding() {
```

```
58         return 'Batman terpentak menabrak bangunan pencakar langit';
```

```
59     }
```

```
60
```

```
1 reference | 0 overrides
```

```
61     public function stopBullet() {
```

```
62         return 'Polisi menembaki superman namun ditangkis';
```

```
63     }
```

```
64 }
```

```
65
```

```
66 $airplane = new Airplane;
```

```
67 $bird = new Bird;
```

```
68 $superman = new Superman;
```

- Output :

### Soal 2

#### Superman

Superman melawan Batman  
Superman mengejar Batman  
Superman melancarkan pukulan  
Batman terpesant 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 Polymorphism dengan penggunaan Interface Flyer sehingga semua class yang Implements dari interface Flyer harus memiliki method takeoff, land, dan fly.

### 3. Buat program berdasarkan UML berikut

- Kodingan :

#### 1. Interface

```
C: > Users > user > Downloads > 013_Diah Ayuning Tyas_No.3 Interface_Praktikum 6.php > ...
1  <!-- Diah Ayuning Tyas - 21091397013 -->
2  <?php
3
4  5 references | 5 implementations
   interface Flyer {
5      7 references | 5 overrides
       public function takeOff();
6      7 references | 5 overrides
       public function land();
7      7 references | 5 overrides
       public function fly();
8  }
9
10 0 references | 0 implementations
   interface Sailer {
11      2 references | 0 overrides
       public function dock();
12      2 references | 0 overrides
       public function cruise();
13  }
```

#### 2. Indexs

```
C: > Users > user > Downloads > 013_Diah Ayuning Tyas_No.3 Index_Praktikum 6.php
1  <!-- Diah Ayuning Tyas - 21091397013 -->
2
3  <?php
4      require_once 'no3.php';
5  >
6
7  <!DOCTYPE html>
8  <html lang="id">
9
10 <head>
11     <!-- Bootstrap CSS -->
12     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
13         integrity="sha384-18mE4kWBq78iYhF1dvKuhF7AU6auU8tT94WrHffjDbrCEXSU1oBoqy12Qv26jIW3" crossorigin="anonymous">
14
15     <title>PBO - Praktikum 6</title>
16 </head>
17 <body>
18     <div class="container">
19         <h2 class="text-center">PBO - Praktikum 6</h2>
20         <div class="row">
21             <div class="col-5 mx-auto border p-3 mt-2">
22                 <h4 class="text-center"><strong>Soal 3</strong></h4>
23                 <br><br>
24                 <?= $singa->eat(); ?> <br>
25                 <?= $manusia->eat(); ?> <br>
26                 <br>
27                 <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
28                 <?= $airplane2->addBox(5000) . ' kg'; ?> <br>
29                 <?= $airplane2->addBox(7000) . ' kg'; ?> <br>
30                 <?= $airplane2->addBox(3000) . ' kg'; ?> <br>
31                 <?= $airplane2->addBox(4000) . ' kg'; ?> <br>
32                 <?= $airplane2->takeOff(); ?> <br>
33                 <?= $airplane2->fly(); ?> <br>
34                 <?= $airplane2->land(); ?> <br>
35                 <br>
36                 <?php
37                     echo "Jadi, Butuh Bahan Bakar sebanyak " . $airplane2->calcFuelNeeds() . ' Liter'. '<br>';
38                 ?>
39                 <br>
40                 <?= $superman2->eat(); ?> <br>
41                 <?= $superman2->land(); ?> <br>
42                 <?= $superman2->takeOff(); ?> <br>
43                 <?= $superman2->fly(); ?> <br>
44                 <?= $superman2->leapBuilding(); ?> <br>
45                 <?= $superman2->stopBullet(); ?> <br>
46             </div>
47         </div>
48     </div>
49 </body>
50 </html>
```

#### 3. PHP



C:\Users\user > Downloads > 013\_Diah Ayuning Tyas\_No.3 PHP\_Praktikum 6.php > ...

```
1  |<!-- Diah Ayuning Tyas - 21091397013 -->
2  |<?php
3  |
4  |require_once '../No3/abstract.php';
5  |require_once '../No3/interface.php';
6  |
7  |2 references | 2 implementations
8  |class Animal
9  |{
10 |    6 references
11 |    protected $name;
12 |
13 |    1 reference | 0 overrides
14 |    public function __construct($name)
15 |    {
16 |        $this->name = $name;
17 |    }
18 |
19 |    0 references | 0 overrides
20 |    public function eat()
21 |    {
22 |        return $this->name . ' sedang makan';
23 |    }
24 |}
25 |
26 |2 references | 1 implementation
27 |class Homosapiens extends Animal {}
28 |
29 |1 reference | 0 implementations
30 |class Airplane2 extends Vehicle implements Flyer
31 |{
32 |    1 reference | 0 overrides | prototype
33 |    public function __construct($maxLoad, $name)
```

```
34 |    {
35 |        $this->maxLoad = $maxLoad;
36 |        $this->name = $name;
37 |    }
38 |
39 |    5 references | 0 overrides
40 |    public function takeOff()
41 |    {
42 |        return "$this->name lepas landas";
43 |    }
44 |
45 |    5 references | 0 overrides
46 |    public function land()
47 |    {
48 |        return "$this->name mendarat";
49 |    }
50 |
51 |    5 references | 0 overrides
52 |    public function fly()
53 |    {
54 |        return "$this->name dalam perjalanan";
55 |    }
56 |
57 |    5 references | 0 overrides | prototype
58 |    public function calcFuelNeeds()
59 |    {
60 |        $fuel = $this->calcFuelEfficiency();
61 |        $strip = $this->calcTripDistance();
62 |
63 |        return ceil($fuel / $strip);
64 |    }
65 |}
```

```

55     }
56 }
57
58 1 reference | 0 implementations
59 class Superman2 extends Homosapiens implements Flyer
60 {
61     5 references | 0 overrides
62     public function takeOff()
63     {
64         return "$this->name mengejar Batman";
65     }
66
67     5 references | 0 overrides
68     public function land()
69     {
70         return "$this->name melawan Batman";
71     }
72
73     5 references | 0 overrides
74     public function fly()
75     {
76         return "$this->name melancarkan pukulan";
77     }
78
79     1 reference | 0 overrides
80     public function leapBuilding()
81     {
82         return "Batman terpesantap menabrak bangunan pencakar langit";
83     }
84
85     1 reference | 0 overrides
86     public function stopBullet()
87     {
88         return "Polisi menembaki $this->name namun ditangkis";
89     }
90 }
91
92 $singa = new Animal('Anjing');
93 $manusia = new Homosapiens('Rahman');
94 $airplane2 = new Airplane2(25000, 'Grande Air');
95 $superman2 = new Superman2('Superman');

```

- Output :

**Soal 3**

Anjing sedang makan  
Rahman sedang makan

**Maksimal muatan Grande Air 25000 kg**  
Grande Air menambah muatan sebesar 5000 kg  
Grande Air menambah muatan sebesar 7000 kg  
Grande Air menambah muatan sebesar 3000 kg  
Grande Air menambah muatan sebesar 4000 kg  
Grande Air lepas landas  
Grande Air dalam perjalanan  
Grande Air mendarat  
Jadi, Butuh Bahan Bakar sebanyak 6 Liter

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

- Penjelasan :

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. Buat program berdasarkan UML berikut

- Kodingan :

##### 1. Interface

```
C: > Users > user > Downloads > 013_Diah Ayuning Tiyas_No.4 Interface_Praktikum 6.php > ...
1  |<!-- Diah Ayuning Tiyas - 21091397013 -->
2
3  <?php
4
5  5 references | 5 implementations
6  interface Flyer {
7      7 references | 5 overrides
8      public function takeOff();
9      7 references | 5 overrides
10     public function land();
11     7 references | 5 overrides
12     public function fly();
13 }
14
15 0 references | 0 implementations
16 interface Sailer {
17     2 references | 0 overrides
18     public function dock();
19     2 references | 0 overrides
20     public function cruise();
21 }
22 }
```

##### 2. Indexs

```
C: > Users > user > Downloads > 013_Diah Ayuning Tiyas_No.4 Index_Praktikum 6.php
1  <!-- Diah Ayuning Tiyas - 21091397013 -->
2  <?php
3      require_once 'no4.php';
4  >
5
6  <!DOCTYPE html>
7  <html lang="en">
8
9  <head>
10     <!-- Bootstrap CSS -->
11     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
12         integrity="sha384-1BmE4kWBq78iYhFdvKuhfTAU6auU8tT94MrHftjDbrCEXSU1oBoqy12QvZ6jIW3" crossorigin="anonymous">
13
14     <title>PBO - Praktikum 6</title>
15 </head>
16 <body>
17     <h2 class="text-center">PBO - Praktikum 6</h2>
18     <div class="container">
19         <div class="row">
20             <h4 class="text-center"><strong>Soal 4</strong></h4>
21             <br><br>
22             <div class="col-4 mx-auto border p-2 mt-2">
23                 <b><?=$RiverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
24                 <?=$RiverBarge2->addBox(15000) . ' kg'; ?> <br>
25                 <?=$RiverBarge2->addBox(13000) . ' kg'; ?> <br>
26                 <?=$RiverBarge2->addBox(2000) . ' kg'; ?> <br>
27                 <?=$RiverBarge2->addBox(5000) . ' kg'; ?> <br>
28                 <?=$RiverBarge2->dock(); ?> <br>
29                 <?=$RiverBarge2->cruise(); ?> <br>
30                 <?php
31                     echo "Jadi, Butuh Bahan Bakar sebanyak " . $RiverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
32                 ?>
33             </div>
34             <div class="col-4 mx-auto border p-2 mt-2">
35                 <b><?=$seaPlane->getMaxLoad() . ' kg'; ?> <br></b>
36                 <?=$seaPlane->addBox(14000) . ' kg'; ?> <br>
37                 <?=$seaPlane->addBox(9000) . ' kg'; ?> <br>
38                 <?=$seaPlane->dock(); ?> <br>
39                 <?=$seaPlane->cruise(); ?> <br>
40                 <?=$seaPlane->takeOff(); ?> <br>
41                 <?=$seaPlane->fly(); ?> <br>
42                 <?=$seaPlane->land(); ?> <br>
43                 <?php
44                     echo "Jadi, Butuh Bahan Bakar sebanyak " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
45                 ?>
46             </div>
47             <div class="col mx-auto border p-2 mt-2">
48                 <b><?=$helicopter->getMaxLoad() . ' kg'; ?> <br></b>
49                 <?=$helicopter->addBox(10000) . ' kg'; ?> <br>
50                 <?=$helicopter->addBox(3000) . ' kg'; ?> <br>
51                 <?=$helicopter->takeOff(); ?> <br>
52                 <?=$helicopter->fly(); ?> <br>
53                 <?=$helicopter->land(); ?> <br>
54                 <?php
55                     echo "Jadi, Butuh Bahan Bakar sebanyak " . $helicopter->calcFuelNeeds() . ' Liter'. '<br>';
56                 ?>
57             </div>
58         </div>
59     </div>
60 </body>
61 </html>
```

### 3. PHP

```
1 <!-- Diah Ayuning Tyas - 21091397013 -->
2
3 <?php
4
5 require_once '../No4/abstract.php';
6 require_once '../No4/interface.php';
7
8 1 reference | 0 implementations
9 class RiverBarge2 extends Vehicle implements Sailer {
10     1 reference | 0 overrides | prototype
11     public function __construct($maxLoad, $name) {
12         $this->maxLoad = $maxLoad;
13         $this->name = $name;
14     }
15
16     6 references | 0 overrides | prototype
17     public function calcFuelNeeds() {
18         $fuel = $this->calcFuelEfficiency();
19         $strip = $this->calcTripDistance();
20
21         return ceil($fuel / $strip);
22     }
23
24     2 references | 0 overrides
25     public function dock() {
26         return $this->name . ' berada di dermaga';
27     }
28
29     2 references | 0 overrides
30     public function cruise() {
31         return $this->name . ' sedang berlayar';
32     }
33 }
34
35 1 reference | 0 implementations
36 class Airplane2 implements Flyer {
37     5 references | 0 overrides
38     public function takeOff() {
39         return 'Pesawat lepas landas';
40     }
41
42     5 references | 0 overrides
43     public function land() {
44         return 'Pesawat mendarat';
45     }
46
47     5 references | 0 overrides
48     public function fly() {
49         return 'Pesawat dalam perjalanan';
50     }
51 }
52
53 1 reference | 0 implementations
54 class SeaPlane extends Vehicle implements Sailer {
55     1 reference | 0 overrides | prototype
56     public function __construct($maxLoad, $name) {
57         $this->maxLoad = $maxLoad;
58         $this->name = $name;
59     }
60
61     6 references | 0 overrides | prototype
62     public function calcFuelNeeds() {
63         $fuel = $this->calcFuelEfficiency();
64         $strip = $this->calcTripDistance();
65
66         return ceil($fuel / $strip);
67     }
68 }
```

```

54
55     2 references | 0 overrides
56     public function dock() {
57         return $this->name . ' berada di dermaga';
58     }
59
60     2 references | 0 overrides
61     public function cruise() {
62         return $this->name . ' sedang berlayar';
63     }
64
65     1 reference | 0 overrides
66     public function takeOff() {
67         return $this->name . ' lepas landas';
68     }
69
70     1 reference | 0 overrides
71     public function land() {
72         return $this->name . ' mendarat';
73     }
74
75     1 reference | 0 overrides
76     public function fly() {
77         return $this->name . ' dalam perjalanan';
78     }
79 }
80
81     1 reference | 0 implementations
82 class Helicopter extends Vehicle {
83     1 reference | 0 overrides | prototype
84     public function __construct($maxLoad, $name) {
85         $this->maxLoad = $maxLoad;
86         $this->name = $name;
87     }
88
89     6 references | 0 overrides | prototype
90     public function calcFuelNeeds() {
91         $fuel = $this->calcFuelEfficiency();
92         $trip = $this->calcTripDistance();
93
94         return ceil($fuel / $trip);
95     }
96
97     1 reference | 0 overrides
98     public function takeOff() {
99         return $this->name . ' lepas landas';
100     }
101
102     1 reference | 0 overrides
103     public function land() {
104         return $this->name . ' mendarat';
105     }
106
107     1 reference | 0 overrides
108     public function fly() {
109         return $this->name . ' dalam perjalanan';
110     }
111 }
112
113 $riverBarge2 = new RiverBarge2(35000, 'Brantas');
114 $seaPlane = new SeaPlane(30000, 'Thrive');
115 $helicopter = new Helicopter(15000, 'Your');

```

- Output :

#### Soal 4

<b>Maksimal muatan Brantas 35000 kg</b> Brantas menambah muatan sebesar 15000 kg Brantas menambah muatan sebesar 13000 kg Brantas menambah muatan sebesar 2000 kg Brantas menambah muatan sebesar 5000 kg Brantas berada di dermaga Brantas sedang kapal Jadi, Butuh Bahan Bakar sebanyak 3 Liter	<b>Maksimal muatan Thrive 30000 kg</b> Thrive menambah muatan sebesar 14000 kg Thrive menambah muatan sebesar 9000 kg Thrive berada di dermaga Thrive sedang kapal Thrive lepas landas landas Thrive dalam perjalanan Thrive mendarat Jadi, Butuh Bahan Bakar sebanyak 5 Liter	<b>Maksimal muatan muatan Anda 15000 kg</b> Anda menambah muatan sebesar 10000 kg Anda menambah muatan sebesar 3000 kg Anda lepas landas Anda mendarat Jadi, Butuh Bahan Bakar sebanyak 8 Liter
--	---	--

- Penjelasan :

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.