

## **LAPORAN PRAKTIKUM 6**

**Mata Kuliah Pemograman Berorientasi Obyek**



**Disusun Oleh :**

Kania Meliana Fityanti (21091397028)

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA**

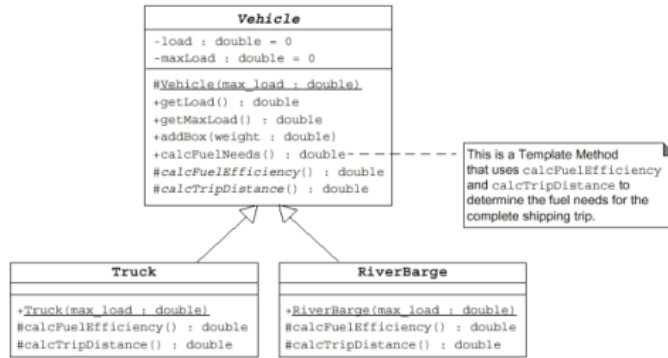
**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2022**

# 1. Buat program berdasarkan UML berikut

1. Buat program berdasarkan UML berikut



## Source Code

### PHP

```
1 <!-- Nama : Kania Meliana Fityanti
2 NIM : 21091397028
3 Kelas : 2021 B-->
4
5 <?php
6
7 require_once 'Nomor1 Abstract.php';
8
9 class Truck extends Vehicle {
10     public function __construct($maxLoad, $name)
11     {
12         $this->maxLoad = $maxLoad;
13         $this->name = $name;
14     }
15
16     public function calcFuelNeeds()
17     {
18         $fuel = $this->calcFuelEfficiency();
19         $trip = $this->calcTripDistance();
20
21         return ceil($fuel / $trip);
22     }
23 }
24
```

```
25 class RiverBarge extends Vehicle {
26     public function __construct($maxLoad, $name)
27     {
28         $this->maxLoad = $maxLoad;
29         $this->name = $name;
30     }
31
32     public function calcFuelNeeds()
33     {
34         $fuel = $this->calcFuelEfficiency();
35         $trip = $this->calcTripDistance();
36
37         return ceil($fuel / $trip);
38     }
39 }
40 $truck = new Truck(16000, 'Truk');
41 $riverBarge = new RiverBarge(15000, 'Perahu');
```

### Abstract PHP

```
1 <!-- Nama : Kania Meliana Fityanti
2 NIM : 21091397028
3 Kelas : 2021 B-->
4
5 <?php
6
7 abstract class Vehicle {
8     private $load = 0;
9     protected $maxLoad = 0, $name;
10
11     protected function __construct($maxLoad, $name) {
12         $this->$maxLoad = $maxLoad;
13         $this->$name = $name;
14     }
15
16     public function getLoad() {
17         return $this->load;
18     }
19
20     public function getMaxLoad() {
21         echo 'Maksimal muatan ' . $this->name . ' ';
22         return $this->maxLoad;
23     }
24
```

```
25     public function addBox($weight) {
26         if ($this->load >= $this->maxLoad) {
27             echo "$this->name menambah muatan sebesar $weight <br>";
28             echo "Muatan telah penuh tidak bisa menambah lagi";
29         } else {
30             $this->load += $weight;
31             echo "$this->name menambah muatan sebesar $weight";
32         }
33     }
34
35     abstract public function calcFuelNeeds();
36
37     protected function calcFuelEfficiency() {
38         $range = 50000000;
39         $range /= $this->load;
40         return $range;
41     }
42
43     protected function calcTripDistance() {
44         return 500;
45     }
46 }
```

## Index PHP

```
1 <!-- Nama : Kania Meliana Fityanti
2 NIM : 21091397028
3 Kelas : 2021 B-->
4
5 <?php
6 require_once 'Nomor1.php';
7 ?>
8
9 <!DOCTYPE html>
10 <html lang="id">
11
12 <head>
13 <!-- Bootstrap CSS -->
14 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
15 integrity="sha384-1BmE4kWBq781YhF1dvKuhfTAU6au08tT94WrfhtjDbrCEXSUIoBoqy12Qv26jIW3" crossorigin="ano
16
17 <title>PBO Praktikum 6</title>
18 </head>
19
20 <body>
21 <div class="container">
22 <br>
23 <h2 class="text-center">PBO - Praktikum 6</h2>
24 <div class="row">
25 <div class="col-5 mx-auto border p-3 mt-2">
26 <h4 class="text-center"><strong>Soal 1</strong></h4>
27 <br><br>
28 <b><?= $truck->getMaxLoad() . ' kg'; ?> <br></b>
29 <br>
30 <?= $truck->addBox(1000) . ' kg'; ?> <br>
31 <?= $truck->addBox(7000) . ' kg'; ?> <br>
32 <?= $truck->addBox(8000) . ' kg'; ?> <br>
33
34 <?php
35 echo "Jadi, Butuh Bahan Bakar sebanyak " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
36 ?>
37 <br>
38 -----
39 <br>
40 <br>
41 <b><?= $riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
42 <br>
43 <?= $riverBarge->addBox(2000) . ' kg'; ?> <br>
44 <?= $riverBarge->addBox(4000) . ' kg'; ?> <br>
45 <?= $riverBarge->addBox(9000) . ' kg'; ?> <br>
46
47 <?php
48 echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge->calcFuelNeeds() . ' Liter';
49 ?>
50 </div>
51 </div>
52 </div>
53 </body>
54
55 </html>
```

## Output

### PBO - Praktikum 6

#### Soal 1

##### Maksimal muatan Truk 16000 kg

Truk menambah muatan sebesar 1000 kg  
Truk menambah muatan sebesar 7000 kg  
Truk menambah muatan sebesar 8000 kg  
Jadi, Butuh Bahan Bakar sebanyak 7 Liter

##### Maksimal muatan Perahu 15000 kg

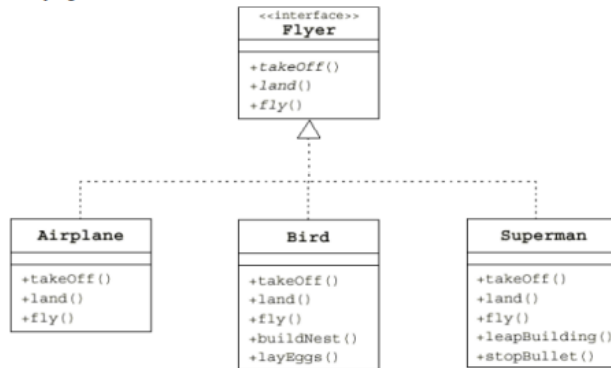
Perahu menambah muatan sebesar 2000 kg  
Perahu menambah muatan sebesar 4000 kg  
Perahu menambah muatan sebesar 9000 kg  
Jadi, Butuh Bahan Bakar sebanyak 7 Liter

## Analisa

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

2. Buat program berdasarkan UML berikut



## Source Code

### PHP

```
1 <!-- Kania Melliana Fityanti_21091397028 -->
2
3 <?php
4
5 require_once 'Nomor2 interface.php';
6
7 class Airplane implements Flyer {
8     public function takeOff() {
9         return 'Pesawat lepas landas..';
10    }
11
12    public function land() {
13        return 'Pesawat mendarat';
14    }
15
16    public function fly() {
17        return 'Pesawat dalam perjalanan';
18    }
19 }
20
21 class Bird implements Flyer {
22     public function takeOff() {
23         return 'Burung mencari makan';
24    }
25
26    public function land() {
27        return 'Burung kembali pulang';
28    }
29
30    public function fly() {
31        return 'Burung terbang';
32    }
33 }
```

```
34
35 public function buildNest() {
36     return 'Burung membuat sarang';
37 }
38
39 public function layEggs() {
40     return 'Burung bertelur';
41 }
42
43 class Superman implements Flyer {
44     public function takeOff() {
45         return 'Superman mengejar Batman';
46     }
47
48     public function land() {
49         return 'Superman melawan Batman';
50     }
51
52     public function fly() {
53         return 'Superman melancarkan pukulan';
54     }
55
56     public function leapBuilding() {
57         return 'Batman terpelant menabrak bangunan pencakar langit';
58     }
59
60     public function stopBullet() {
61         return 'Polisi menembaki superman namun ditangkis';
62     }
63 }
64 }
```

```
65 $airplane = new Airplane;
66 $bird = new Bird;
67 $superman = new Superman;
```

## Interface PHP

```
1 <!-- Kania Meliana Fityanti_21091397028 -->
2
3 <?php
4
5 interface Flyer {
6     public function takeOff();
7     public function land();
8     public function fly();
9 }
10
11 interface Sailer {
12     public function dock();
13     public function cruise();
14 }
```

## Index PHP

```
1 <!-- Kania Meliana Fityanti_21091397028 -->
2
3 <?php
4     require_once 'Nomor2.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="sha384-18me4kWBq781YhF1dvKuhfTAU6auU8tT94WnrHfjtjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
14
15     <title>Praktikum 6</title>
16 </head>
17
18 <body>
19     <div class="container">
20         <br>
21         <div class="row">
22             <div class="col-5 mx-auto border p-3 mt-2">
23                 <h4 class="text-center"><strong><u>Soal 2</u></strong></h4>
24                 <br><br>
25                 <b><?php
26                     echo "Superman";
27                     ?></b> <br>
28                     <?= $superman->land(); ?> <br>
29                     <?= $superman->takeOff(); ?> <br>
30                     <?= $superman->fly(); ?> <br>
31                     <?= $superman->leapBuilding(); ?> <br>
32                     <?= $superman->stopBullet(); ?> <br>
33                 </b>
34             </div>
35             <div class="col-5 mx-auto border p-3 mt-2">
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                 </b>
45             </div>
46             <div class="col-5 mx-auto border p-3 mt-2">
47                 <b><?php
48                     echo "Airplane";
49                     ?></b> <br>
50                     <?= $airplane->takeOff(); ?> <br>
51                     <?= $airplane->fly(); ?> <br>
52                     <?= $airplane->land(); ?> <br>
53                 </b>
54             </div>
55         </div>
56     </div>
57 </body>
58 </html>
```

## Output

### Soal 2

#### Superman

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

#### Bird

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

#### Airplane

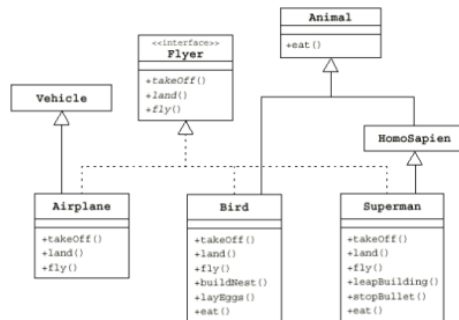
Pesawat lepas landas..  
Pesawat dalam perjalanan  
Pesawat mendarat

## Analisa

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

3. Buat program berdasarkan UML berikut



## Source Code

### Interface PHP

```
1 <!-- Kania Meliana Fityanti_21091397028 -->
2
3 <?php
4
5 interface Flyer {
6     public function takeOff();
7     public function land();
8     public function fly();
9 }
10
11 interface Sailer {
12     public function dock();
13     public function cruise();
14 }
```

```

1 <!-- Kania Meliana Fityanti_21091397028 -->
2
3 <?php
4
5 require_once 'Nomor3 abstract.php';
6 require_once 'Nomor3 interface.php';
7
8 class Animal
9 {
10     protected $name;
11
12     public function __construct($name)
13     {
14         $this->name = $name;
15     }
16
17     public function eat()
18     {
19         return $this->name . ' sedang makan';
20     }
21 }
22
23 class Homosapiens extends Animal {}
24
25 class Airplane2 extends Vehicle implements Flyer
26 {
27     public function __construct($maxLoad, $name)
28     {
29         $this->maxLoad = $maxLoad;
30         $this->name = $name;
31     }
32
33     public function takeOff()

```

```

33     public function takeOff()
34     {
35         return "$this->name lepas landas";
36     }
37
38     public function land()
39     {
40         return "$this->name mendarat";
41     }
42
43     public function fly()
44     {
45         return "$this->name dalam perjalanan";
46     }
47
48     public function calcFuelNeeds()
49     {
50         $fuel = $this->calcFuelEfficiency();
51         $strip = $this->calcTripDistance();
52
53
54         return ceil($fuel / $strip);
55     }
56 }
57
58
59 class Superman2 extends Homosapiens implements Flyer
60 {
61     public function takeOff()
62     {
63         return "$this->name mengejar Batman";
64     }

```

```

65
66     public function land()
67     {
68         return "$this->name melawan Batman";
69     }
70
71     public function fly()
72     {
73         return "$this->name melancarkan pukulan";
74     }
75
76     public function leapBuilding()
77     {
78         return "Batman terpentak menabrak bangunan pencakar langit";
79     }
80
81     public function stopBullet()
82     {
83         return "Polisi menembaki $this->name namun ditangkis";
84     }
85 }
86
87 $burung = new Animal('Burung');
88 $manusia = new Homosapiens('Aransha');
89 $airplane2 = new Airplane2(25000, 'Batik Air');
90 $superman2 = new Superman2('Superman');

```

## Abstract PHP

```
1 <!-- Kania Meliana Fityanti_21091397028 -->
2
3 <?php
4
5 abstract class Vehicle {
6     private $load = 0;
7     protected $maxLoad = 0, $name;
8
9     protected function __construct($maxLoad, $name) {
10         $this->$maxLoad = $maxLoad;
11         $this->$name = $name;
12     }
13
14     public function getLoad() {
15         return $this->load;
16     }
17
18     public function getMaxLoad() {
19         echo 'Maksimal muatan ' . $this->name . ' ' ;
20         return $this->maxLoad;
21     }
22
23     public function addBox($weight) {
24         if ($this->load >= $this->maxLoad) {
25             echo "$this->name menambah muatan sebesar $weight <br>";
26             echo 'Muatan telah penuh tidak bisa menambah lagi';
27         } else {
28             $this->load += $weight;
29             echo "$this->name menambah muatan sebesar $weight";
30         }
31     }
32
33     abstract public function calcFuelNeeds();
```

```
33     abstract public function calcFuelNeeds();
34
35     protected function calcFuelEfficiency() {
36         $range = 50000000;
37         $range /= $this->load;
38         return $range;
39     }
40
41     protected function calcTripDistance() {
42         return 500;
43     }
44 }
```

## Index PHP

```
1 <!-- Kania Meliana Fityanti_21091397028 -->
2
3 <?php
4     require_once 'Nomor3.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="sha384-1BmE4kqBQ781YhF1dvKuhfTAU6auU8tT94MrHftjDbrCEXSU1oBoqy12QvZ6jIW3" crossorigin="anonymous">
14
15     <title>Praktikum 6</title>
16 </head>
17 <body>
18     <div class="container">
19         <div class="row">
20             <div class="col-5 mx-auto border p-3 mt-2">
21                 <h4 class="text-center"><strong><u>Soal 3</u></strong></h4>
22                 <br><br>
23                 <?= $burung->eat(); ?> <br>
24                 <?= $manusia->eat(); ?> <br>
25                 <br>
26                 <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
27                 <?= $airplane2->addBox(6000) . ' kg'; ?> <br>
28                 <?= $airplane2->addBox(2000) . ' kg'; ?> <br>
29                 <?= $airplane2->addBox(7000) . ' kg'; ?> <br>
30                 <?= $airplane2->addBox(5000) . ' kg'; ?> <br>
31                 <?= $airplane2->takeOff(); ?> <br>
32                 <?= $airplane2->fly(); ?> <br>
33                 <?= $airplane2->land(); ?> <br>
```

```
33                 <?= $airplane2->fly(); ?> <br>
34                 <?= $airplane2->land(); ?> <br>
35
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
51 </html>
```



## Output

### Soal 3

Burung sedang makan  
Aransha sedang makan

#### Maksimal muatan Batik Air 25000 kg

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

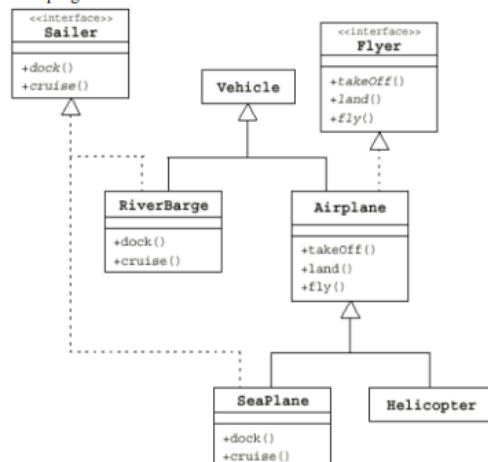
Superman sedang makan  
Superman melawan Batman  
Superman mengejar Batman  
Superman melancarkan pukulan  
Batman terpental menabrak bangunan pencakar langit  
Polisi menembaki Superman namun ditangkis

## Analisa

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

4. Buat program berdasarkan UML berikut



## Source Code

### Interface PHP

```
1  <!-- Nama      : Kania Meliana Fityanti
2      NIM       : 21091397028
3      Kelas    : 2021 B-->
4
5  <?php
6
7  interface Flyer {
8      public function takeOff();
9      public function land();
10     public function fly();
11 }
12
13 interface Sailer {
14     public function dock();
15     public function cruise();
16 }
```

### PHP

```
1  <!-- Nama      : Kania Meliana Fityanti
2      NIM       : 21091397028
3      Kelas    : 2021 B-->
4
5  <?php
6
7  require_once 'Nomor4 abstract.php';
8  require_once 'Nomor4 interface.php';
9
10 class RiverBarge2 extends Vehicle implements Sailer {
11     public function __construct($maxLoad, $name) {
12         $this->maxLoad = $maxLoad;
13         $this->name = $name;
14     }
15
16     public function calcFuelNeeds() {
17         $fuel = $this->calcFuelEfficiency();
18         $trip = $this->calcTripDistance();
19
20         return ceil($fuel / $trip);
21     }
22
23     public function dock() {
24         return $this->name . ' berada di dermaga';
25     }
26
27     public function cruise() {
28         return $this->name . ' sedang berlayar';
29     }
30 }
31
```

```
65     public function takeOff() {
66         return $this->name . ' lepas landas';
67     }
68
69     public function land() {
70         return $this->name . ' mendarat';
71     }
72
73     public function fly() {
74         return $this->name . ' dalam perjalanan';
75     }
76 }
77
78 class Helicopter extends Vehicle {
79     public function __construct($maxLoad, $name) {
80         $this->maxLoad = $maxLoad;
81         $this->name = $name;
82     }
83
84     public function calcFuelNeeds() {
85         $fuel = $this->calcFuelEfficiency();
86         $trip = $this->calcTripDistance();
87
88         return ceil($fuel / $trip);
89     }
90     public function takeOff() {
91         return $this->name . ' lepas landas';
92     }
93
94     public function land() {
95         return $this->name . ' mendarat';
96     }
97 }
```

```
32 class Airplane2 implements Flyer {
33     public function takeOff() {
34         return 'Pesawat lepas landas';
35     }
36     public function land() {
37         return 'Pesawat mendarat';
38     }
39     public function fly() {
40         return 'Pesawat dalam perjalanan';
41     }
42 }
43
44 class SeaPlane extends Vehicle implements Sailer {
45     public function __construct($maxLoad, $name) {
46         $this->maxLoad = $maxLoad;
47         $this->name = $name;
48     }
49
50     public function calcFuelNeeds() {
51         $fuel = $this->calcFuelEfficiency();
52         $trip = $this->calcTripDistance();
53
54         return ceil($fuel / $trip);
55     }
56
57     public function dock() {
58         return $this->name . ' berada di dermaga';
59     }
60
61     public function cruise() {
62         return $this->name . ' sedang berlayar';
63     }
64 }
```

```
98     public function fly() {
99         return $this->name . ' dalam perjalanan';
100     }
101 }
102
103 $riverBarge2 = new RiverBarge2(30000, 'Atomic');
104 $seaPlane = new SeaPlane(20000, 'Titanic');
105 $helicopter = new Helicopter(10000, 'Brocklyn');
```

## Abstract PHP

```
1 <!-- Nama : Kania Meliana Fityanti
2 NIM : 21091397028
3 Kelas : 2021 B-->
4
5 <?php
6
7 abstract class Vehicle {
8     private $load = 0;
9     protected $maxLoad = 0, $name;
10
11     protected function __construct($maxLoad, $name) {
12         $this->$maxLoad = $maxLoad;
13         $this->$name = $name;
14     }
15
16     public function getLoad() {
17         return $this->load;
18     }
19
20     public function getMaxLoad() {
21         echo 'Maksimal muatan ' . $this->name . ' ' ;
22         return $this->maxLoad;
23     }
24
25     public function addBox($weight) {
26         if ($this->load >= $this->maxLoad) {
27             echo "$this->name menambah muatan sebesar $weight <br>";
28             echo 'Muatan telah penuh tidak bisa menambah lagi';
29         } else {
30             $this->load += $weight;
31             echo "$this->name menambah muatan sebesar $weight";
32         }
33     }
34
35     abstract public function calcFuelNeeds();
36
37     protected function calcFuelEfficiency() {
38         $range = 50000000;
39         $range /= $this->load;
40         return $range;
41     }
42
43     protected function calcTripDistance() {
44         return 500;
45     }
46 }
```

## Index PHP

```
1 <!-- Nama : Kania Meliana Fityanti
2 NIM : 21091397028
3 Kelas : 2021 B-->
4
5 <?php
6 require_once 'Nomor4.php';
7
8
9 <!DOCTYPE html>
10 <html lang="en">
11
12 <head>
13 <!-- Bootstrap CSS -->
14 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
15 integrity="sha384-18mE4k8q781YhF1dvKuhfTAU6auU8tT94WwHfjtDbrCEXSU1oBoqyl2QvZ6jIM3" crossorigin="anonymous">
16
17 <title>PBO - Praktikum 6</title>
18 </head>
19 <body>
20 <h2 class="text-center">PBO - Praktikum 6</h2>
21 <div class="container">
22 <div class="row">
23 <h4 class="text-center"><strong>Soal 4</strong></h4>
24 <br><br>
25 <div class="col-4 mx-auto border p-2 mt-2">
26 <b><?= $riverBarge2->getMaxLoad() . ' kg'; ?></b><br>
27 <?= $riverBarge2->addBox(12000) . ' kg'; ?><br>
28 <?= $riverBarge2->addBox(14000) . ' kg'; ?><br>
29 <?= $riverBarge2->addBox(1800) . ' kg'; ?><br>
30 <?= $riverBarge2->addBox(3800) . ' kg'; ?><br>
31 <?= $riverBarge2->dock(); ?><br>
32 <?= $riverBarge2->cruise(); ?><br>
33 </div>
34 </div>
35 </body>
36 </html>
```

```

33         <?php
34             echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
35         }>
36     </div>
37     <div class="col-4 mx-auto border p-2 mt-2">
38         <b><?= $seaPlane->getMaxLoad() . ' kg'; ?> <br></b>
39         <?= $seaPlane->addBox(12000) . ' kg'; ?> <br>
40         <?= $seaPlane->addBox(8000) . ' kg'; ?> <br>
41         <?= $seaPlane->dock(); ?> <br>
42         <?= $seaPlane->cruise(); ?> <br>
43         <?= $seaPlane->takeOff(); ?> <br>
44         <?= $seaPlane->fly(); ?> <br>
45         <?= $seaPlane->land(); ?> <br>
46     <?php
47         echo "Jadi, Butuh Bahan Bakar sebanyak " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
48     }>
49 </div>
50 <div class="col mx-auto border p-2 mt-2">
51     <b><?= $helicopter->getMaxLoad() . ' kg'; ?> <br></b>
52     <?= $helicopter->addBox(8000) . ' kg'; ?> <br>
53     <?= $helicopter->addBox(2000) . ' kg'; ?> <br>
54     <?= $helicopter->takeOff(); ?> <br>
55     <?= $helicopter->fly(); ?> <br>
56     <?= $helicopter->land(); ?> <br>
57 <?php
58     echo "Jadi, Butuh Bahan Bakar sebanyak " . $helicopter->calcFuelNeeds() . ' Liter'. '<br>';
59 }>
60 </div>
61 </div>
62 </div>
63 </body>
64 </html>

```

## Output

### PBO - Praktikum 6 Soal 4

<b>Maksimal muatan Atomic 30000 kg</b> Atomic menambah muatan sebesar 12000 kg Atomic menambah muatan sebesar 14000 kg Atomic menambah muatan sebesar 1000 kg Atomic menambah muatan sebesar 3000 kg Atomic berada di dermaga Atomic sedang berlayar Jadi, Butuh Bahan Bakar sebanyak 4 Liter	<b>Maksimal muatan Titanic 20000 kg</b> Titanic menambah muatan sebesar 12000 kg Titanic menambah muatan sebesar 8000 kg Titanic berada di dermaga Titanic sedang berlayar Titanic lepas landas Titanic dalam perjalanan Titanic mendarat Jadi, Butuh Bahan Bakar sebanyak 5 Liter	<b>Maksimal muatan Brooklyn 10000 kg</b> Brooklyn menambah muatan sebesar 8000 kg Brooklyn menambah muatan sebesar 2000 kg Brooklyn lepas landas Brooklyn dalam perjalanan Brooklyn mendarat Jadi, Butuh Bahan Bakar sebanyak 10 Liter
--	--	--

## Analisa

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