

**LAPORAN PEMROGRAMAN BERORIENTASI OBJEK  
UJIAN TENGAH SEMESTER**



Disusun oleh :

Muvidha Fatmawati Putri (21091397011)  
A2021 MI

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA  
FAKULTAS VOKASI  
UNIVERSITAS NEGERI SURABAYA  
2022**

1. Buat program berdasarkan UML berikut

- Kodingan :

```
C: > xampp > htdocs > UTS PBO > Praktikum 6 > 011_Praktikum06_1.php > ...
1  <!-- Muvidha Fatmawati Putri - 21091397011 -->
2
3
4  <?php
5
6  2 references | 2 implementations
7  abstract class Vehicle {
8      4 references
9      private $load = 0;
10     protected $maxLoad = 0, $name;
11
12     0 references | 2 overrides
13     protected function __construct($maxLoad, $name) {
14         $this->$maxLoad = $maxLoad;
15         $this->$name = $name;
16     }
17
18     0 references | 0 overrides
19     public function getLoad() {
20         return $this->load;
21     }
22
23     2 references | 0 overrides
24     public function getMaxLoad() {
25         echo 'Maksimal muatan ' . $this->name . ' ' ;
26         return $this->maxLoad;
27     }
28
29     6 references | 0 overrides
30     public function addBox($weight) {
31         if ($this->load >= $this->maxLoad) {
32             echo "$this->name menambah muatan sebesar $weight <br>";
33             echo 'Muatan telah penuh tidak bisa menambah lagi';
34         }else {
35             $this->load += $weight;
36             echo "$this->name menambah muatan sebesar $weight";
37         }
38     }
39
40     2 references | 2 overrides
41     abstract public function calcFuelNeeds();
42
43     2 references | 0 overrides
44     protected function calcFuelEfficiency() {
45         $range = 1000000000;
46         $range /= $this->load;
47         return $range;
48     }
49
50     2 references | 0 overrides
51     protected function calcTripDistance() {
52         return 100000;
53     }
54 }
55
56 1 reference | 0 implementations
57 class Truck extends Vehicle {
58     1 reference | 0 overrides | prototype
59     public function __construct($maxLoad, $name)
60     {
61         $this->maxLoad = $maxLoad;
62         $this->name = $name;
63     }
64
65     2 references | 0 overrides | prototype
66     public function calcFuelNeeds()
```

```

54     {
55         $fuel = $this->calcFuelEfficiency();
56         $strip = $this->calcTripDistance();
57     }
58     return ceil($fuel / $strip);
59 }
60 }
61
62 1 reference | 0 implementations
63 class RiverBarge extends Vehicle {
64     1 reference | 0 overrides | prototype
65     public function __construct($maxLoad, $name)
66     {
67         $this->maxLoad = $maxLoad;
68         $this->name = $name;
69     }
70
71     2 references | 0 overrides | prototype
72     public function calcFuelNeeds()
73     {
74         $fuel = $this->calcFuelEfficiency();
75         $strip = $this->calcTripDistance();
76     }
77     return ceil($fuel / $strip);
78 }
79
80 $truck = new Truck(20000, 'Truk');
81 $riverBarge = new RiverBarge(35000, 'Perahu');
82
83 >>
84
85 <!DOCTYPE html>
86 <html lang="id">
87
88 <head>
89     <!-- Bootstrap CSS -->
90     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
91         integrity="sha384-18m4KWBq78iYhF1dvKuhfTAU6AuU8tT94WRFtjDbrCEXSU1oBoqy12QvZ6jIW3" crossorigin="anonymous">
92
93     <title>PBO - Praktikum 6</title>
94 </head>
95
96 <body>
97     <div class="container">
98         <br>
99         <h2 class="text-center">PBO - Praktikum 6</h2>
100         <div class="row">
101             <div class="col-5 mx-auto border p-3 mt-2">
102                 <h4 class="text-center"><strong>Soal 1</strong></h4>
103                 <br><br>
104                 <b><?=$truck->getMaxLoad() . ' kg'; ?> <br></b>
105                 <br>
106                 <?=$truck->addBox(3000) . ' kg'; ?> <br>
107                 <?=$truck->addBox(8000) . ' kg'; ?> <br>
108                 <?=$truck->addBox(9000) . ' kg'; ?> <br>
109
110                 <?php
111                     echo "Jadi, Butuh Bahan Bakar sebanyak " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
112                 ?>
113
114                 <br>
115                 -----
116                 <br>
117                 <br>
118                 <b><?=$riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
119                 <br>
120                 <?=$riverBarge->addBox(12000) . ' kg'; ?> <br>
121                 <?=$riverBarge->addBox(10000) . ' kg'; ?> <br>
122                 <?=$riverBarge->addBox(7000) . ' kg'; ?> <br>
123
124                 <?php
125                     echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge->calcFuelNeeds() . ' Liter';
126                 ?>
127
128             </div>
129         </div>
130     </div>
131 </body>
132 </html>

```

- Output :

Soal 1
<p><b>Maksimal muatan Truk 20000 kg</b></p> <p>Truk menambah muatan sebesar 3000 kg  Truk menambah muatan sebesar 8000 kg  Truk menambah muatan sebesar 9000 kg  Jadi, Butuh Bahan Bakar sebanyak 1 Liter</p>
<p><b>Maksimal muatan Perahu 35000 kg</b></p> <p>Perahu menambah muatan sebesar 12000 kg  Perahu menambah muatan sebesar 10000 kg  Perahu menambah muatan sebesar 7000 kg  Jadi, Butuh Bahan Bakar sebanyak 1 Liter</p>

- 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 :

C:\> xampp > htdocs > UTS PBO > Praktikum 6 > 011\_Praktikum06\_2.php > ...

```
1  <!-- Muvidha Fatmawati Putri - 21091397011 -->
2
3  <?php
4
5  3 references | 3 implementations
6  interface Flyer {
7      3 references | 3 overrides
8      public function takeOff();
9      3 references | 3 overrides
10     public function land();
11     3 references | 3 overrides
12     public function fly();
13 }
14
15 0 references | 0 implementations
16 interface Sailer {
17     0 references | 0 overrides
18     public function dock();
19     0 references | 0 overrides
20     public function cruise();
21 }
22
23 1 reference | 0 implementations
24 class Airplane implements Flyer {
25     3 references | 0 overrides
26     public function takeOff() {
27         return 'Pesawat lepas landas..';
28     }
29
30     3 references | 0 overrides
31     public function land() {
32         return 'Pesawat mendarat';
33     }
34
35     3 references | 0 overrides
36     public function fly() {
37         return 'Pesawat dalam perjalanan';
38     }
39 }
40
41 1 reference | 0 implementations
42 class Bird implements Flyer {
43     3 references | 0 overrides
44     public function takeOff() {
45         return 'Burung mencari makan';
46     }
47
48     3 references | 0 overrides
49     public function land() {
50         return 'Burung kembali pulang';
51     }
52
53     3 references | 0 overrides
54     public function fly() {
55         return 'Burung terbang';
56     }
57
58     1 reference | 0 overrides
59     public function buildNest() {
60         return 'Burung membuat sarang';
61     }
62
63     1 reference | 0 overrides
64     public function layEggs() {
65         return 'Burung bertelur';
66     }
67 }
```

```

50
51 1 reference | 0 implementations
class Superman implements Flyer {
52     3 references | 0 overrides
53     public function takeOff() {
54         return 'Superman mengejar Batman';
55     }
56
57     3 references | 0 overrides
58     public function land() {
59         return 'Superman melawan Batman';
60     }
61
62     3 references | 0 overrides
63     public function fly() {
64         return 'Superman melancarkan pukulan';
65     }
66
67     1 reference | 0 overrides
68     public function leapBuilding() {
69         return 'Batman terpenjantak menabrak bangunan pencakar langit';
70     }
71
72     1 reference | 0 overrides
73     public function stopBullet() {
74         return 'Polisi menembaki superman namun ditangkis';
75     }
76 }
77
78 $airplane = new Airplane;
79 $bird = new Bird;
80 $superman = new Superman;
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126

```

- Output :

## Soal 2

### Superman

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

### Bird

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

### Airplane

Pesawat lepas landas.  
Pesawat dalam perjalanan  
Pesawat mendarat

- Penjelasan :

Implementasi 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 :

```
C:\> xampp > htdocs > UTS PBO > Praktikum 6 > 011_Praktikum06.3.php > Vehicle

1  <!-- Muvidha Fatmawati Putri - 21091397011 -->
2  <?php
3
3 references | 3 implementations
4  interface Flyer {
5      public function takeOff();
6      public function land();
7      public function fly();
8  }
9
2 references | 2 implementations
10 interface Sailer {
11     public function dock();
12     public function cruise();
13 }
14
4 references | 5 implementations
15 abstract class Vehicle {
16     private $load = 0;
17     protected $maxLoad = 0, $name;
18
0 references | 4 overrides
19     protected function __construct($maxLoad, $name) {
20         $this->$maxLoad = $maxLoad;
21         $this->$name = $name;
22     }
23
0 references | 0 overrides
24     public function getLoad() {
25         return $this->load;
26     }
27
4 references | 0 overrides
28     public function getMaxLoad() {
29         echo 'Maksimal muatan ' . $this->name . ' ' ;
30         return $this->maxLoad;
31     }
32
12 references | 0 overrides
33     public function addBox($weight) {
34         if ($this->load >= $this->maxLoad) {
35             echo "$this->name menambah muatan sebesar $weight <br>";
36             echo 'Muatan telah penuh tidak bisa menambah lagi';
37         } else {
38             $this->load += $weight;
39             echo "$this->name menambah muatan sebesar $weight";
40         }
41     }
42
4 references | 4 overrides
43     abstract public function calcFuelNeeds();
44
4 references | 0 overrides
45     protected function calcFuelEfficiency() {
46         $range = 50000000;
47         $range /= $this->load;
48         return $range;
49     }
50
4 references | 0 overrides
51     protected function calcTripDistance() {
52         return 500;
53     }
54 }
55
2 references | 2 implementations
56 class Animal {
57     6 references
58     protected $name;
59
1 reference | 0 overrides
60     public function __construct($name) {
61         $this->name = $name;
62     }
63
3 references | 0 overrides
64     public function eat() {
65         return $this->name . ' sedang makan';
66     }
67 }
68
2 references | 1 implementation
69 class Homosapiens extends Animal {}
70
1 reference | 0 implementations
71
1 reference | 0 implementations
72 class Airplane2 extends Vehicle implements Flyer {
73
1 reference | 0 overrides | prototype
74     public function __construct($maxLoad, $name) {
75         $this->maxLoad = $maxLoad;
76     }
```



```

76         $this->maxLoad = $maxLoad;
77         $this->name = $name;
78     }
79
80     2 references | 0 overrides
81     public function takeOff()
82     {
83         return "$this->name lepas landas";
84     }
85
86     2 references | 0 overrides
87     public function land()
88     {
89         return "$this->name mendarat";
90     }
91
92     2 references | 0 overrides
93     public function fly()
94     {
95         return "$this->name dalam perjalanan";
96     }
97
98     4 references | 0 overrides | prototype
99     public function calcFuelNeeds()
100     {
101         $fuel = $this->calcFuelEfficiency();
102         $strip = $this->calcTripDistance();
103
104         return ceil($fuel / $strip);
105     }
106
107     1 reference | 0 implementations
108     class Superman2 extends Homosapiens implements Flyer
109     {
110         2 references | 0 overrides
111         public function takeOff()
112         {
113             return "$this->name mengejar Batman";
114         }
115
116         2 references | 0 overrides
117         public function land()
118         {
119             return "$this->name melawan Batman";
120         }
121
122         2 references | 0 overrides
123         public function fly()
124         {
125             return "$this->name melancarkan pukulan";
126         }
127
128         1 reference | 0 overrides
129         public function leapBuilding()
130         {
131             return "Batman terpentak menabrak bangunan pencakar langit";
132         }
133
134         1 reference | 0 overrides
135         public function stopBullet()
136         {
137             return "Polisi menembaki $this->name namun ditangkis";
138         }
139     }
140
141     $singa = new Animal('kucing');
142     $manusia = new Homosapiens('Candra');
143     $airplane2 = new Airplane2(100000, 'okey plane');
144     $superman2 = new Superman2('Superman');
145
146     <!DOCTYPE html>
147     <html lang="id">
148
149     <head>
150
151     <!-- Bootstrap CSS -->
152     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
153     integrity="sha384-18mE4kWBq78iYhF1dVkuHfTAU6auU8tT94MrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
154
155     <title>PBO - Praktikum 6</title>
156     </head>
157     <body>
158
159     <div class="container">
160
161     <h2 class="text-center">PBO - Praktikum 6</h2>
162     <div class="row">
163
164     <div class="col-5 mx-auto border p-3 mt-2">
165
166     <h4 class="text-center"><strong>Soal 3</strong></h4>
167     <br><br>
168     <?= $singa->eat(); ?> <br>
169     <?= $manusia->eat(); ?> <br>
170     <br>
171
172     <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
173     <?= $airplane2->addBox(2000) . ' kg'; ?> <br>

```

```

163         <?= $airplane2->addBox(4000) . ' kg'; ?> <br>
164         <?= $airplane2->addBox(2000) . ' kg'; ?> <br>
165         <?= $airplane2->addBox(9000) . ' kg'; ?> <br>
166         <?= $airplane2->takeOff(); ?> <br>
167         <?= $airplane2->fly(); ?> <br>
168         <?= $airplane2->land(); ?> <br>
169
170         <?php
171             echo "Jadi, Butuh Bahan Bakar sebanyak " . $airplane2->calcFuelNeeds() . ' Liter'. ' <br>';
172             ?>
173         <br>
174         <?= $superman2->eat(); ?> <br>
175         <?= $superman2->land(); ?> <br>
176         <?= $superman2->takeOff(); ?> <br>
177         <?= $superman2->fly(); ?> <br>
178         <?= $superman2->leapBuilding(); ?> <br>
179         <?= $superman2->stopBullet(); ?> <br>
180     </div>
181 </div>
182 </div>
183 </body>
184
185 </html>

```

- Output :

**Soal 3**

kucing sedang makan  
Candra sedang makan

**Maksimal muatan okey plane 100000 kg**  
 okey plane menambah muatan sebesar 2000 kg  
 okey plane menambah muatan sebesar 4000 kg  
 okey plane menambah muatan sebesar 2000 kg  
 okey plane menambah muatan sebesar 9000 kg  
 okey plane lepas landas  
 okey plane dalam perjalanan  
 okey plane mendarat  
 Jadi, Butuh Bahan Bakar sebanyak 6 Liter

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

- Penjelasan :

Terdapat interface Flyer dan 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

```
C:\> xampp > htdocs > UTS PBO > Praktikum 6 > 011_Praktikum06_4.php > ...
1  <!-- Muvidha Fatmawati Putri - 21091397811 -->
2
3  <?php
4
5  3 references | 3 implementations
6  interface Flyer {
7      2 references | 3 overrides
8      public function takeOff();
9      2 references | 3 overrides
10     public function land();
11     2 references | 3 overrides
12     public function fly();
13 }
14
15 2 references | 2 implementations
16 interface Sailer {
17     2 references | 2 overrides
18     public function dock();
19     2 references | 2 overrides
20     public function cruise();
21 }
22
23 1 reference | 0 implementations
24 class RiverBarge2 extends Vehicle implements Sailer {
25     1 reference | 0 overrides | prototype
26     public function __construct($maxLoad, $name) {
27         $this->maxLoad = $maxLoad;
28         $this->name = $name;
29     }
30
31     4 references | 0 overrides | prototype
32     public function calcFuelNeeds() {
33         $fuel = $this->calcFuelEfficiency();
34         $strip = $this->calcTripDistance();
35
36         return ceil($fuel / $strip);
37     }
38
39     2 references | 0 overrides
40     public function dock() {
41         return $this->name . ' berada di dermaga';
42     }
43
44     2 references | 0 overrides
45     public function cruise() {
46         return $this->name . ' sedang berlayar';
47     }
48 }
49
50 1 reference | 0 implementations
51 class Airplane2 implements Flyer {
52     2 references | 0 overrides
53     public function takeOff() {
54         return 'Pesawat lepas landas';
55     }
56
57     2 references | 0 overrides
58     public function land() {
59         return 'Pesawat mendarat';
60     }
61
62     2 references | 0 overrides
63     public function fly() {
64         return 'Pesawat dalam perjalanan';
65     }
66 }
67
68 1 reference | 0 implementations
69 class SeaPlane extends Vehicle implements Sailer {
```

```

50      1 reference | 0 overrides | prototype
51      public function __construct($maxLoad, $name) {
52          $this->maxLoad = $maxLoad;
53          $this->name = $name;
54      }
55
56      4 references | 0 overrides | prototype
57      public function calcFuelNeeds() {
58          $fuel = $this->calcFuelEfficiency();
59          $strip = $this->calcTripDistance();
60
61          return ceil($fuel / $strip);
62      }
63
64      2 references | 0 overrides
65      public function dock() {
66          return $this->name . ' berada di dermaga';
67      }
68
69      2 references | 0 overrides
70      public function cruise() {
71          return $this->name . ' sedang berlayar';
72      }
73
74      1 reference | 0 overrides
75      public function takeOff() {
76          return $this->name . ' lepas landas';
77      }
78
79      1 reference | 0 overrides
80      public function land() {
81          return $this->name . ' mendarat';
82      }
83
84      1 reference | 0 overrides
85      public function fly() {
86          return $this->name . ' dalam perjalanan';
87      }
88
89      1 reference | 0 implementations
90      class Helicopter extends Vehicle {
91          1 reference | 0 overrides | prototype
92          public function __construct($maxLoad, $name) {
93              $this->maxLoad = $maxLoad;
94              $this->name = $name;
95          }
96
97          4 references | 0 overrides | prototype
98          public function calcFuelNeeds() {
99              $fuel = $this->calcFuelEfficiency();
100             $strip = $this->calcTripDistance();
101
102             return ceil($fuel / $strip);
103         }
104
105         1 reference | 0 overrides
106         public function takeOff() {
107             return $this->name . ' lepas landas';
108         }
109
110         1 reference | 0 overrides
111         public function land() {
112             return $this->name . ' mendarat';
113         }
114
115         1 reference | 0 overrides
116         public function fly() {
117             return $this->name . ' dalam perjalanan';
118         }
119     }
120
121     $riverBarge2 = new RiverBarge2(35000, 'Muvidha');
122     $seaPlane = new SeaPlane(30000, 'Fatma');
123     $helicopter = new Helicopter(15000, 'Putri');
124     ?>
125
126     <!DOCTYPE html>
127     <html lang="en">
128
129     <head>
130         <!-- Bootstrap CSS -->
131         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
132             integrity="sha384-18mE4KBq781YhFlvKuhF7AU6auUBt194WrfHfj0brCEXSU1oBoqy12QvZ6jIM3" crossorigin="anonymous">
133
134         <title>PBO - Praktikum 6</title>
135     </head>
136     <body>
137         <h2 class="text-center">PBO - Praktikum 6</h2>
138         <div class="container">
139             <div class="row">
140                 <h4 class="text-center"><strong>Soal 4</strong></h4>
141                 <br><br>
142                 <div class="col-4 mx-auto border p-2 mt-2">
143                     <b><?=$riverBarge2->getMaxLoad() . ' kg'; ?><br></b>
144                     <?=$riverBarge2->addBox(15000) . ' kg'; ?><br>
145                     <?=$riverBarge2->addBox(13000) . ' kg'; ?><br>
146                     <?=$riverBarge2->addBox(2000) . ' kg'; ?><br>

```

```

134         <?= $riverBarge2->addBox(5000) . ' kg'; ?> <br>
135         <?= $riverBarge2->dock(); ?> <br>
136         <?= $riverBarge2->cruise(); ?> <br>
137     <?php
138         echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
139     ?>
140 </div>
141 <div class="col-4 mx-auto border p-2 mt-2">
142     <b><?= $seaPlane->getMaxLoad() . ' kg'; ?> <br></b>
143     <?= $seaPlane->addBox(14000) . ' kg'; ?> <br>
144     <?= $seaPlane->addBox(9000) . ' kg'; ?> <br>
145     <?= $seaPlane->dock(); ?> <br>
146     <?= $seaPlane->cruise(); ?> <br>
147     <?= $seaPlane->takeOff(); ?> <br>
148     <?= $seaPlane->fly(); ?> <br>
149     <?= $seaPlane->land(); ?> <br>
150     <?php
151         echo "Jadi, Butuh Bahan Bakar sebanyak " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
152     ?>
153 </div>
154 <div class="col mx-auto border p-2 mt-2">
155     <b><?= $helicopter->getMaxLoad() . ' kg'; ?> <br></b>
156     <?= $helicopter->addBox(10000) . ' kg'; ?> <br>
157     <?= $helicopter->addBox(3000) . ' kg'; ?> <br>
158     <?= $helicopter->takeOff(); ?> <br>
159     <?= $helicopter->fly(); ?> <br>
160     <?= $helicopter->land(); ?> <br>
161     <?php
162         echo "Jadi, Butuh Bahan Bakar sebanyak " . $helicopter->calcFuelNeeds() . ' Liter'. '<br>';
163     ?>
164 </div>
165 </div>
166 </div>
167 </body>
168
169 </html>
170 <?php
171
172 4 references | 5 implementations
abstract class Vehicle {
173     8 references
174     private $load = 0;
175     protected $maxLoad = 0, $name;
176
177     0 references | 4 overrides
protected function __construct($maxLoad, $name) {
178         $this->$maxLoad = $maxLoad;
179         $this->$name = $name;
180     }
181
182     0 references | 0 overrides
public function getLoad() {
183         return $this->load;
184     }
185
186     4 references | 0 overrides
public function getMaxLoad() {
187         echo 'Maksimal muatan ' . $this->name . ' ' ;
188         return $this->maxLoad;
189     }
190
191     12 references | 0 overrides
public function addBox($weight) {
192         if ($this->load >= $this->maxLoad) {
193             echo "$this->name menambah muatan sebesar $weight <br>";
194             echo 'Muatan telah penuh tidak bisa menambah lagi';
195         }else {
196             $this->load += $weight;
197             echo "$this->name menambah muatan sebesar $weight";
198         }
199     }
200
201     4 references | 4 overrides
abstract public function calcFuelNeeds();
202
203     4 references | 0 overrides
protected function calcFuelEfficiency() {
204         $range = 50000000;
205         $range /= $this->load;
206         return $range;
207     }
208
209     4 references | 0 overrides
protected function calcTripDistance() {
210         return 500;
211     }
212 }

```

- Output :

#### Soal 4

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

- **Penjelasan :**

Implementasi 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.