

LAPORAN PRAKTIKUM 6



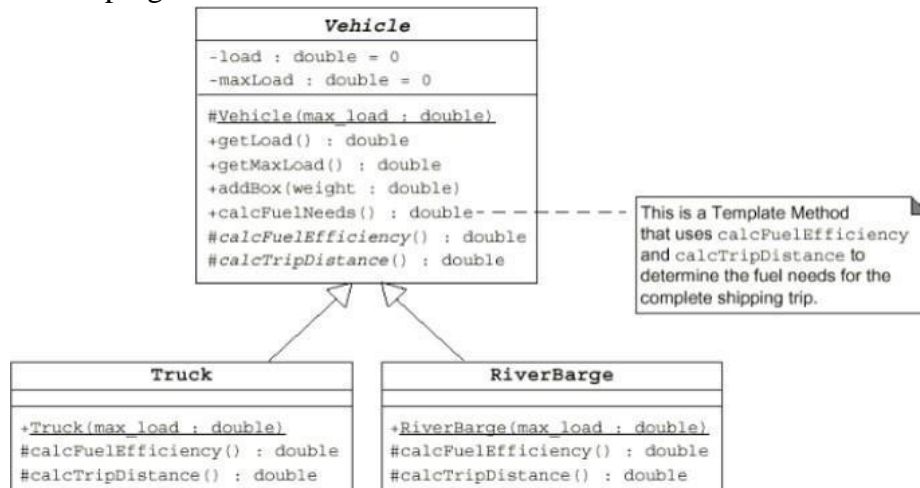
Oleh :

Filusive Nathan Fernanda

21091397073

**D4 MANAJEMEN INFORMATIKA
FAKULTAS VOKASI
UNIVERSITAS NEGERI
SURABAYA TAHUN 2022**

1. Buat program berdasarkan UML berikut



• Source Code

```

1  <?php
2  require_once "1a.php"; ?>
3
4  <!DOCTYPE html>
5  <html lang="id">
6
7  <head>
8      <!-- Bootstrap CSS -->
9      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" crossorigin="anonymous">
10
11     <title>Praktikum 6</title>
12 </head>
13
14 <body>
15     <div class="container">
16         <br>
17         <h2 class="text-center">Praktikum 6</h2>
18         <div class="row">
19             <div class="col-5 mx-auto border p-3 mt-2">
20                 <h4 class="text-center"><strong>Nomor 1</strong></h4>
21                 <br><br>
22                 <b><?=$truck->getMaxLoad() . " kg" ?> <br></b>
23                 <br>
24                 <?=$truck->addBox(3000) . " kg" ?> <br>
25                 <?=$truck->addBox(1000) . " kg" ?> <br>
26                 <?=$truck->addBox(6000) . " kg" ?> <br>
27
28                 <?php echo "Jadi, membutuhkan bahan bakar sebanyak " . $truck->calcFuelNeeds() . " Liter" . "<br>"; ?>
29                 <br>
30                 <br>
31                 <b><?=$riverBarge->getMaxLoad() . " kg" ?> <br></b>
32                 <br>
33                 <?=$riverBarge->addBox(1000) . " kg" ?> <br>
34                 <?=$riverBarge->addBox(6000) . " kg" ?> <br>
35                 <?=$riverBarge->addBox(8000) . " kg" ?> <br>
36
37                 <?php echo "Jadi, membutuhkan bahan bakar sebanyak " . $riverBarge->calcFuelNeeds() . " Liter"; ?>
38             </div>
39         </div>
40     </div>
41 </body>
42 </html>

```

```

1  <?php
2
3  require_once "1b.php";
4
5  1 reference | 0 implementations
6  class Truck extends Vehicle
7  {
8      1 reference | 0 overrides | prototype
9      public function __construct($maxLoad, $name)
10     {
11         $this->maxLoad = $maxLoad;
12         $this->name = $name;
13     }
14
15     2 references | 0 overrides | prototype
16     public function calcFuelNeeds()
17     {
18         $fuel = $this->calcFuelEfficiency();
19         $trip = $this->calcTripDistance();
20
21         return ceil($fuel /= $trip);
22     }
23 }

```

```

24  1 reference | 0 implementations
25  class RiverBarge extends Vehicle
26  {
27      1 reference | 0 overrides | prototype
28      public function __construct($maxLoad, $name)
29      {
30          $this->maxLoad = $maxLoad;
31          $this->name = $name;
32      }
33
34      2 references | 0 overrides | prototype
35      public function calcFuelNeeds()
36      {
37          $fuel = $this->calcFuelEfficiency();
38          $trip = $this->calcTripDistance();
39
40          return ceil($fuel /= $trip);
41      }
42  }
43
44  $truck = new Truck(10000, "Truk");
45  $riverBarge = new RiverBarge(15000, "Perahu");

```

```

1  <?php
2
3  2 references | 2 implementations
4  abstract class Vehicle
5  {
6      4 references
7      private $load = 0;
8      protected $maxLoad = 0,
9          $name;
10
11      0 references | 2 overrides
12      protected function __construct($maxLoad, $name)
13      {
14          $this->$maxLoad = $maxLoad;
15          $this->$name = $name;
16      }
17
18      0 references | 0 overrides
19      public function getLoad()
20      {
21          return $this->load;
22      }
23
24      2 references | 0 overrides
25      public function getMaxLoad()
26      {
27          echo "Maksimal muatan " . $this->name . " ";
28          return $this->maxLoad;
29      }
30

```

```
26 6 references | 0 overrides
    public function addBox($weight)
27 {
28     if ($this->load >= $this->maxLoad) {
29         echo "$this->name menambah muatan sebesar $weight <br>";
30         echo "Muatan telah penuh tidak bisa menambah lagi";
31     } else {
32         $this->load += $weight;
33         echo "$this->name menambah muatan sebesar $weight";
34     }
35 }
36
37 2 references | 2 overrides
    abstract public function calcFuelNeeds();
38
39 2 references | 0 overrides
    protected function calcFuelEfficiency()
40 {
41     $range = 50000000;
42     $range /= $this->load;
43     return $range;
44 }
45
46 2 references | 0 overrides
    protected function calcTripDistance()
47 {
48     return 500;
49 }
50 }
```

- **Output**

Praktikum 6

Nomor 1

Maksimal muatan Truk 10000 kg

Truk menambah muatan sebesar 3000 kg

Truk menambah muatan sebesar 1000 kg

Truk menambah muatan sebesar 6000 kg

Jadi, membutuhkan bahan bakar sebanyak 10 Liter

Maksimal muatan Perahu 15000 kg

Perahu menambah muatan sebesar 1000 kg

Perahu menambah muatan sebesar 6000 kg

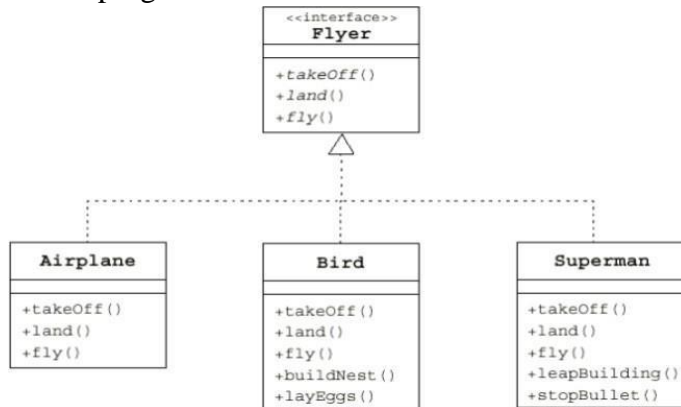
Perahu menambah muatan sebesar 8000 kg

Jadi, membutuhkan bahan bakar sebanyak 7 Liter

- **Analisis**

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



• Source Code

```
1  <?php
2  require_once "2a.php"; ?>
3
4  <!DOCTYPE html>
5  <html lang="en">
6
7  <head>
8      <!-- Bootstrap CSS -->
9      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" crossorigin="anonymous">
10
11     <title>Praktikum 6</title>
12 </head>
13
14 <body>
15     <div class="container">
16         <br>
17         <h2 class="text-center">Praktikum 6</h2>
18         <div class="row">
19             <div class="col-5 mx-auto border p-3 mt-2">
20                 <h4 class="text-center"><strong>Nomor 2</strong></h4>
21                 <br><br>
22                 <b><?php echo "Superman"; ?></b> <br>
23                 <?= $superman->land() ?> <br>
24                 <?= $superman->takeOff() ?> <br>
25                 <?= $superman->fly() ?> <br>
26                 <?= $superman->leapBuilding() ?> <br>
27                 <?= $superman->stopBullet() ?> <br>
28                 <br>
29                 <b><?php echo "Bird"; ?></b> <br>
30                 <?= $bird->buildNest() ?> <br>
31                 <?= $bird->takeOff() ?> <br>
32                 <?= $bird->fly() ?> <br>
33                 <?= $bird->land() ?> <br>
34                 <?= $bird->layEggs() ?> <br>
35                 <br>
36                 <b><?php echo "Airplane"; ?></b> <br>
37                 <?= $airplane->takeOff() ?> <br>
38                 <?= $airplane->fly() ?> <br>
39                 <?= $airplane->land() ?> <br>
40             </div>
41         </div>
42     </div>
43 </body>
44 </html>
```

```
1  <?php
2
3  require_once "2b.php";
4
5  1 reference | 0 implementations
6  class Airplane implements Flyer
7  {
8      3 references | 0 overrides
9      public function takeOff()
10     {
11         return "Pesawat lepas landas";
12     }
13
14     3 references | 0 overrides
15     public function land()
16     {
17         return "Pesawat mendarat";
18     }
19
20     3 references | 0 overrides
21     public function fly()
22     {
23         return "Pesawat dalam perjalanan";
24     }
25 }
```

```
1  1 reference | 0 implementations
2  class Bird implements Flyer
3  {
4      3 references | 0 overrides
5      public function takeOff()
```



```
56     }
57
58     3 references | 0 overrides
59     public function land()
60     {
61         return "Superman melawan Batman";
62     }
63
64     3 references | 0 overrides
65     public function fly()
66     {
67         return "Superman melancarkan pukulan";
68     }
69
70     1 reference | 0 overrides
71     public function leapBuilding()
72     {
73         return "Batman terpental menabrak bangunan pencakar langit";
74     }
75
76     1 reference | 0 overrides
77     public function stopBullet()
78     {
79         return "Polisi menembaki superman namun ditangkis";
80     }
81 }
82
83 $airplane = new Airplane();
84 $bird = new Bird();
85 $superman = new Superman();
```

```
1  <?php
2
3  3 references | 3 implementations
4  interface Flyer
5  {
6      3 references | 3 overrides
7      public function takeOff();
8      3 references | 3 overrides
9      public function land();
10     3 references | 3 overrides
11     public function fly();
12 }
13
14 0 references | 0 implementations
15 interface Sailer
16 {
17     0 references | 0 overrides
18     public function dock();
19     0 references | 0 overrides
20     public function cruise();
21 }
```

- **Output**

Praktikum 6

Nomor 2

Superman

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

Bird

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

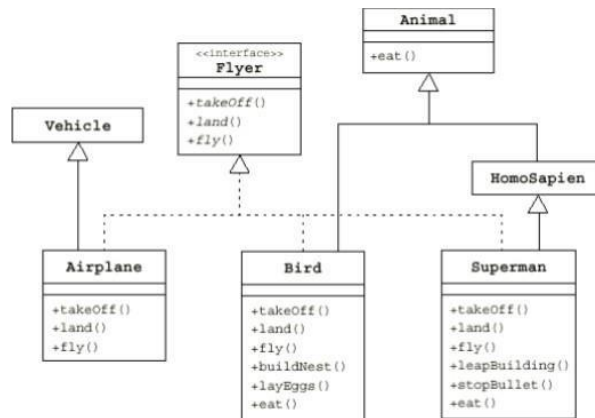
Airplane

Pesawat lepas landas
Pesawat dalam perjalanan
Pesawat mendarat

- **Analisis**

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



• Source Code

```

1  <?php
2  require_once "3a.php"; ?>
3
4  <!DOCTYPE html>
5  <html lang="id">
6
7  <head>
8      <!-- Bootstrap CSS -->
9      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" crossorigin="anonymous">
10
11     <title>Praktikum 6</title>
12 </head>
13 <body>
14     <div class="container">
15         <h2 class="text-center">Praktikum 6</h2>
16         <div class="row">
17             <div class="col-5 mx-auto border p-3 mt-2">
18                 <h4 class="text-center"><strong>Nomor 3</strong></h4>
19                 <br><br>
20                 <?= $harimau->eat() ?> <br>
21                 <?= $manusia->eat() ?> <br>
22                 <br>
23                 <b><?= $airplane2->getMaxLoad() . " kg" ?> <br></b>
24                 <?= $airplane2->addBox(5000) . " kg" ?> <br>
25                 <?= $airplane2->addBox(7000) . " kg" ?> <br>
26                 <?= $airplane2->addBox(3000) . " kg" ?> <br>
27                 <?= $airplane2->addBox(4000) . " kg" ?> <br>
28                 <?= $airplane2->takeOff() ?> <br>
29                 <?= $airplane2->fly() ?> <br>
30                 <?= $airplane2->land() ?> <br>

```

```

29         <?= $airplane2->fly() ?> <br>
30         <?= $airplane2->land() ?> <br>
31
32         <?php echo "Jadi, membutuhkan bahan bakar sebanyak " .
33             $airplane2->calcFuelNeeds() .
34             " Liter" .
35             "<br>"; ?>
36         <br>
37         <?= $superman2->eat() ?> <br>
38         <?= $superman2->land() ?> <br>
39         <?= $superman2->takeOff() ?> <br>
40         <?= $superman2->fly() ?> <br>
41         <?= $superman2->leapBuilding() ?> <br>
42         <?= $superman2->stopBullet() ?> <br>
43     </div>
44 </div>
45 </div>
46 </body>
47 </html>

```

```

1  <?php
2
3  require_once "3b.php";
4  require_once "3c.php";
5
6  2 references | 2 implementations
7  class Animal
8  {
9      6 references
10     protected $name;
11
12     1 reference | 0 overrides
13     public function __construct($name)
14     {
15         $this->name = $name;
16     }
17
18     3 references | 0 overrides
19     public function eat()
20     {
21         return $this->name . " sedang makan";
22     }
23 }
24
25 2 references | 1 implementation
26 class Homosapiens extends Animal
27 {
28 }
29
30 1 reference | 0 implementations
31 class Airplane2 extends Vehicle implements Flyer

```

```

26  {
27      1 reference | 0 overrides | prototype
    public function __construct($maxLoad, $name)
28  {
29      $this->maxLoad = $maxLoad;
30      $this->name = $name;
31  }
32
33      5 references | 0 overrides
    public function takeOff()
34  {
35      return "$this->name lepas landas";
36  }
37
38      5 references | 0 overrides
    public function land()
39  {
40      return "$this->name mendarat";
41  }
42
43      5 references | 0 overrides
    public function fly()
44  {
45      return "$this->name dalam perjalanan";
46  }
47
48      3 references | 0 overrides | prototype
    public function calcFuelNeeds()
49  {
50      $fuel = $this->calcFuelEfficiency();
51      $this->fuel = $fuel * $this->calcFuelDistance();

```

```

51         $trip = $this->calcTripDistance();
52
53         return ceil($fuel /= $trip);
54     }
55 }
56
57 1 reference | 0 implementations
58 class Superman2 extends Homosapiens implements Flyer
59 {
60     5 references | 0 overrides
61     public function takeOff()
62     {
63         return "$this->name mengejar Batman";
64     }
65
66     5 references | 0 overrides
67     public function land()
68     {
69         return "$this->name melawan Batman";
70     }
71
72     5 references | 0 overrides
73     public function fly()
74     {
75         return "$this->name melancarkan pukulan";
76     }
77
78     1 reference | 0 overrides
79     public function leapBuilding()
80     {
81         return "Batman terpentak menabrak bangunan pencakar langit";
82     }
83
84     1 reference | 0 overrides
85     public function stopBullet()
86     {
87         return "Polisi menembaki $this->name namun ditangkis";
88     }
89 }
90
91 $harimau = new Animal("Harimau");
92 $manusia = new Homosapiens("Núñez");
93 $airplane2 = new Airplane2(20000, "Batik Air");
94 $superman2 = new Superman2("Superman");

```

```
1  <?php
2
3  3 references | 3 implementations
4  abstract class Vehicle
5  {
6      8 references
7      private $load = 0;
8      protected $maxLoad = 0,
9          $name;
10
11      0 references | 3 overrides
12      protected function __construct($maxLoad, $name)
13      {
14          $this->$maxLoad = $maxLoad;
15          $this->$name = $name;
16      }
17
18      0 references | 0 overrides
19      public function getLoad()
20      {
21          return $this->load;
22      }
23
24      3 references | 0 overrides
25      public function getMaxLoad()
26      {
27          echo "Maksimal muatan " . $this->name . " ";
28          return $this->maxLoad;
29      }
30  }
```



```

26     public function addBox($weight)
27     {
28         if ($this->load >= $this->maxLoad) {
29             echo "$this->name menambah muatan sebesar $weight <br>";
30             echo "Muatan telah penuh tidak bisa menambah lagi";
31         } else {
32             $this->load += $weight;
33             echo "$this->name menambah muatan sebesar $weight";
34         }
35     }
36
37     3 references | 3 overrides
38     abstract public function calcFuelNeeds();
39
40     3 references | 0 overrides
41     protected function calcFuelEfficiency()
42     {
43         $range = 50000000;
44         $range /= $this->load;
45         return $range;
46     }
47
48     3 references | 0 overrides
49     protected function calcTripDistance()
50     {
51         return 500;
52     }
53 }

```

```

1  <?php
2
3  5 references | 5 implementations
4  interface Flyer
5  {
6      5 references | 5 overrides
7      public function takeOff();
8      5 references | 5 overrides
9      public function land();
10     5 references | 5 overrides
11     public function fly();
12 }
13
14 0 references | 0 implementations
15 interface Sailer
16 {
17     0 references | 0 overrides
18     public function dock();
19     0 references | 0 overrides
20     public function cruise();
21 }

```

- **Output**

Nomor 3

Harimau sedang makan

Núñez sedang makan

Maksimal muatan Batik Air 20000 kg

Batik Air menambah muatan sebesar 5000 kg

Batik Air menambah muatan sebesar 7000 kg

Batik Air menambah muatan sebesar 3000 kg

Batik Air menambah muatan sebesar 4000 kg

Batik Air lepas landas

Batik Air dalam perjalanan

Batik Air mendarat

Jadi, membutuhkan bahan bakar sebanyak 6 Liter

Superman sedang makan

Superman melawan Batman

Superman mengejar Batman

Superman melancarkan pukulan

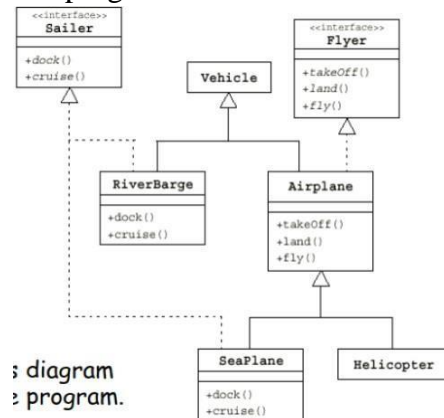
Batman terpental menabrak bangunan pencakar langit

Polisi menembaki Superman namun ditangkis

- **Analisis**

Terdapat interface Flyer dan abstract class Vehicle. Class airplane merupakan implementasi dari interface Flyer dan turunan dari Vehicle sehingga class Airplane harus memiliki method calcFuelNeeds, takeoff, land, dan fly. Class Bird adalah 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



• Source Code

```

1  <?php
2      require_once '4a.php';
3  >
4
5  <!DOCTYPE html>
6  <html lang="en">
7
8  <head>
9      <!-- Bootstrap CSS -->
10     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" crossorigin="anonymous">
11
12     <title>Praktikum 6</title>
13 </head>
14 <body>
15     <h2 class="text-center">Praktikum 6</h2>
16     <div class="container">
17         <div class="row">
18             <div class="col-4 text-center"><strong>Nomor 4</strong></div>
19             <div class="col-4 mx-auto border p-2 mt-2">
20                 <b><?=$riverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
21                 <?=$riverBarge2->addBox(12000) . ' kg'; ?> <br>
22                 <?=$riverBarge2->addBox(14000) . ' kg'; ?> <br>
23                 <?=$riverBarge2->addBox(1000) . ' kg'; ?> <br>
24                 <?=$riverBarge2->addBox(3000) . ' kg'; ?> <br>
25                 <?=$riverBarge2->dock(); ?> <br>
26                 <?=$riverBarge2->cruise(); ?> <br>
27                 <?php
28                     echo "Jadi, membutuhkan bahan bakar sebanyak " . $riverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
29                 >
30             </div>
19         </div>
20     </div>
21 </body>
22 </html>

```

```

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

```

```

1  <?php
2
3  require_once '4b.php';
4  require_once '4c.php';
5
6  1 reference | 0 implementations
7  class RiverBarge2 extends Vehicle implements Sailer {
8      1 reference | 0 overrides | prototype
9      public function __construct($maxLoad, $name) {
10         $this->maxLoad = $maxLoad;
11         $this->name = $name;
12     }
13
14     6 references | 0 overrides | prototype
15     public function calcFuelNeeds() {
16         $fuel = $this->calcFuelEfficiency();
17         $trip = $this->calcTripDistance();
18
19         return ceil($fuel /= $trip);
20     }
21
22     2 references | 0 overrides
23     public function dock() {
24         return $this->name . ' berada di dermaga';
25     }
26
27     2 references | 0 overrides
28     public function cruise() {
29         return $this->name . ' sedang berlayar';
30     }
31 }

```

```

28  class Airplane2 implements Flyer {
    5 references | 0 overrides
29      public function takeOff() {
30          return 'Pesawat lepas landas';
31      }
    5 references | 0 overrides
32      public function land() {
33          return 'Pesawat mendarat';
34      }
    5 references | 0 overrides
35      public function fly() {
36          return 'Pesawat dalam perjalanan';
37      }
38  }
39
    1 reference | 0 implementations
40  class SeaPlane extends Vehicle implements Sailer
    1 reference | 0 overrides | prototype
41      public function __construct($maxLoad, $name)
42          $this->maxLoad = $maxLoad;
43          $this->name = $name;
44      }
45
    6 references | 0 overrides | prototype
46      public function calcFuelNeeds() {
47          $fuel = $this->calcFuelEfficiency();
48          $trip = $this->calcTripDistance();
49
50          return ceil($fuel /= $trip);
51      }

```

```

53     public function dock() {
54         return $this->name . ' berada di dermaga';
55     }
56
57     2 references | 0 overrides
58     public function cruise() {
59         return $this->name . ' sedang berlayar';
60     }
61
62     1 reference | 0 overrides
63     public function takeOff() {
64         return $this->name . ' lepas landas';
65     }
66
67     1 reference | 0 overrides
68     public function land() {
69         return $this->name . ' mendarat';
70     }
71
72     1 reference | 0 overrides
73     public function fly() {
74         return $this->name . ' dalam perjalanan';
75     }
76 }
77
78 1 reference | 0 implementations
79 class Helicopter extends Vehicle {
80     1 reference | 0 overrides | prototype
81     public function __construct($maxLoad, $name) {
82         $this->maxLoad = $maxLoad;
83         $this->name = $name;

```

```

79      6 references | 0 overrides | prototype
80      public function calcFuelNeeds() {
81          $fuel = $this->calcFuelEfficiency();
82          $trip = $this->calcTripDistance();
83
84          return ceil($fuel /= $trip);
85      }
86      1 reference | 0 overrides
87      public function takeOff() {
88          return $this->name . ' lepas landas';
89      }
90
91      1 reference | 0 overrides
92      public function land() {
93          return $this->name . ' mendarat';
94      }
95
96      1 reference | 0 overrides
97      public function fly() {
98          return $this->name . ' dalam perjalanan';
99      }
100
101      $riverBarge2 = new RiverBarge2(30000, 'Storeum');
102      $seaPlane = new SeaPlane(20000, 'Falcon');
103      $helicopter = new Helicopter(10000, 'Hurricane');

```

```

1  <?php
2
3  6 references | 7 implementations
abstract class Vehicle {
4      12 references
    private $load = 0;
5      protected $maxLoad = 0, $name;
6
7      0 references | 6 overrides
    protected function __construct($maxLoad, $name) {
8          $this->$maxLoad = $maxLoad;
9          $this->$name = $name;
10     }
11
12     0 references | 0 overrides
    public function getLoad() {
13         return $this->load;
14     }
15
16     6 references | 0 overrides
    public function getMaxLoad() {
17         echo 'Maksimal muatan ' . $this->name . ' ' ;
18         return $this->maxLoad;
19     }
20
21     18 references | 0 overrides
    public function addBox($weight) {
22         if ($this->load >= $this->maxLoad) {
23             echo "$this->name menambah muatan sebesar $weight <br>";
24             echo 'Muatan telah penuh tidak bisa menambah lagi';
25         }

```

```

31     6 references | 6 overrides
    abstract public function calcFuelNeeds();
32
33     6 references | 0 overrides
    protected function calcFuelEfficiency() {
34         $range = 50000000;
35         $range /= $this->load;
36         return $range;
37     }
38
39     6 references | 0 overrides
    protected function calcTripDistance() {
40         return 500;
41     }
42 }

```



```

1  <?php
2
3  6 references | 6 implementations
4  interface Flyer {
5      5 references | 6 overrides
6      public function takeOff();
7      5 references | 6 overrides
8      public function land();
9      5 references | 6 overrides
10     public function fly();
11 }
12
13 2 references | 2 implementations
14 interface Sailer {
15     2 references | 2 overrides
16     public function dock();
17     2 references | 2 overrides
18     public function cruise();
19 }

```

• Output

Praktikum 6 Nomor 4

Maksimal muatan Storeum 30000 kg Storeum menambah muatan sebesar 12000 kg Storeum menambah muatan sebesar 14000 kg Storeum menambah muatan sebesar 1000 kg Storeum menambah muatan sebesar 3000 kg Storeum berada di dermaga Storeum sedang berlayar Jadi, membutuhkan bahan bakar sebanyak 4 Liter	Maksimal muatan Falcon 20000 kg Falcon menambah muatan sebesar 12000 kg Falcon menambah muatan sebesar 8000 kg Falcon berada di dermaga Falcon sedang berlayar Falcon lepas landas Falcon dalam perjalanan Falcon mendarat Jadi, membutuhkan bahan bakar sebanyak 5 Liter	Maksimal muatan Hurricane 10000 kg Hurricane menambah muatan sebesar 8000 kg Hurricane menambah muatan sebesar 2000 kg Hurricane lepas landas Hurricane dalam perjalanan Hurricane mendarat Jadi, membutuhkan bahan bakar sebanyak 10 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.

