

LAPORAN PRAKTIKUM 6



Oleh:

Alvin Febrianto

21091397031

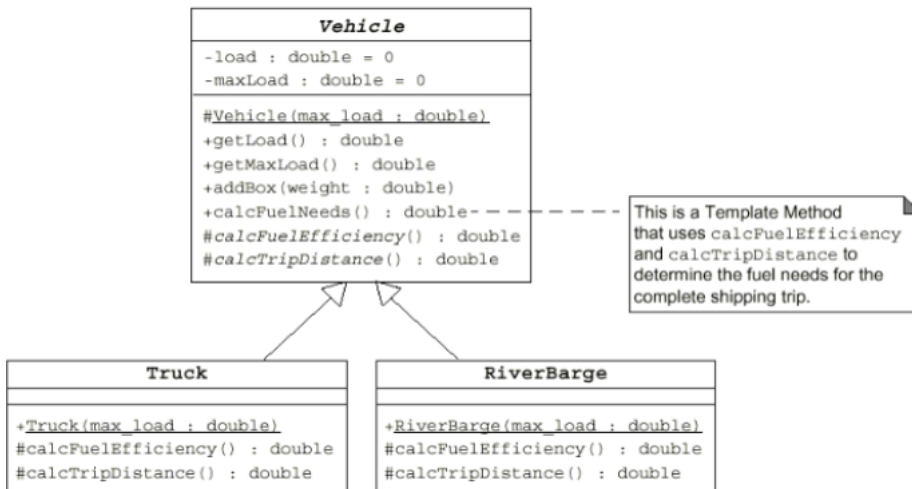
D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

TAHUN AJARAN 2022/2023

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  class Truck extends Vehicle
6  {
7      public function __construct($maxLoad, $name)
8      {
9          $this->maxLoad = $maxLoad;
10         $this->name = $name;
11     }
12
13     public function calcFuelNeeds()
14     {
15         $fuel = $this->calcFuelEfficiency();
16         $trip = $this->calcTripDistance();
17
18         return ceil($fuel /= $trip);
19     }
20 }
21
22 class RiverBarge extends Vehicle
23 {
24     public function __construct($maxLoad, $name)
25     {
26         $this->maxLoad = $maxLoad;
27         $this->name = $name;
28     }
29
30     public function calcFuelNeeds()
31     {
32         $fuel = $this->calcFuelEfficiency();
33         $trip = $this->calcTripDistance();
34
35         return ceil($fuel /= $trip);
36     }
37 }
38 $truck = new Truck(10000, "Truk");
39 $riverBarge = new RiverBarge(15000, "Perahu");

```

```

1  <?php
2
3  abstract class Vehicle
4  {
5      private $load = 0;
6      protected $maxLoad = 0,
7          $name;
8
9      protected function __construct($maxLoad, $name)
10     {
11         $this->$maxLoad = $maxLoad;
12         $this->$name = $name;
13     }

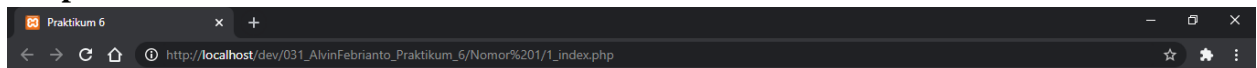
```

```

14
15     public function getLoad()
16     {
17         return $this->load;
18     }
19
20     public function getMaxLoad()
21     {
22         echo "Maksimal muatan " . $this->name . " ";
23         return $this->maxLoad;
24     }
25
26     public function addBox($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     abstract public function calcFuelNeeds();
38
39     protected function calcFuelEfficiency()
40     {
41         $range = 500000000;
42         $range /= $this->load;
43         return $range;
44     }
45
46     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

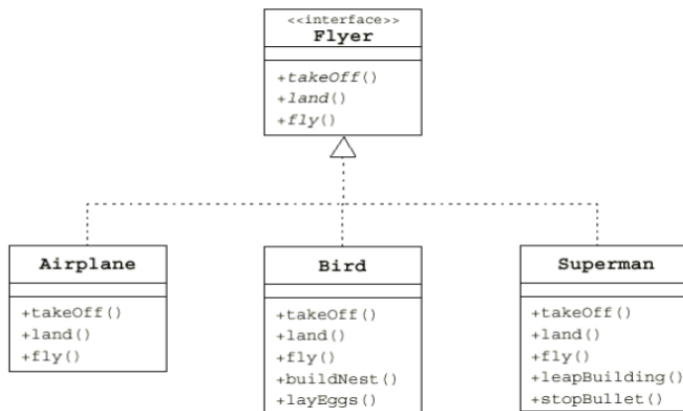
Maksimal muatan Perahu 15000 kg

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>
    
```

```
2a.php x
1 <?php
2
3 require_once "2b.php";
4
5 class Airplane implements Flyer
6 {
7     public function takeOff()
8     {
9         return "Pesawat lepas landas";
10    }
11
12    public function land()
13    {
14        return "Pesawat mendarat";
15    }
16
17    public function fly()
18    {
19        return "Pesawat dalam perjalanan";
20    }
21 }
22
23 class Bird implements Flyer
24 {
25     public function takeOff()
26     {
27         return "Burung mencari makan";
28     }
29
30     public function land()
31     {
32         return "Burung kembali pulang";
33     }
34
35     public function fly()
36     {
37         return "Burung terbang";
38     }
39
40     public function buildNest()
41     {
42         return "Burung membuat sarang";
43     }
44
45     public function layEggs()
```

```

46     {
47         return "Burung bertelur";
48     }
49 }
50
51 class Superman implements Flyer
52 {
53     public function takeOff()
54     {
55         return "Superman mengejar Batman";
56     }
57
58     public function land()
59     {
60         return "Superman melawan Batman";
61     }
62
63     public function fly()
64     {
65         return "Superman melancarkan pukulan";
66     }
67
68     public function leapBuilding()
69     {
70         return "Batman terpentak menabrak bangunan pencakar langit";
71     }
72
73     public function stopBullet()
74     {
75         return "Polisi menembaki superman namun ditangkis";
76     }
77 }
78
79 $airplane = new Airplane();
80 $bird = new Bird();
81 $superman = new Superman();

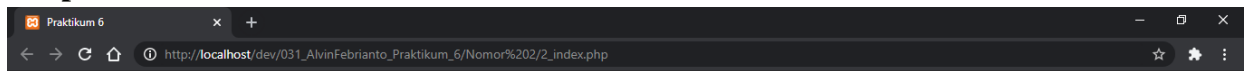
```

```

2b.php x
1 <?php
2
3 interface Flyer
4 {
5     public function takeOff();
6     public function land();
7     public function fly();
8 }
9
10 interface Sailer
11 {
12     public function dock();
13     public function cruise();
14 }

```

- **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

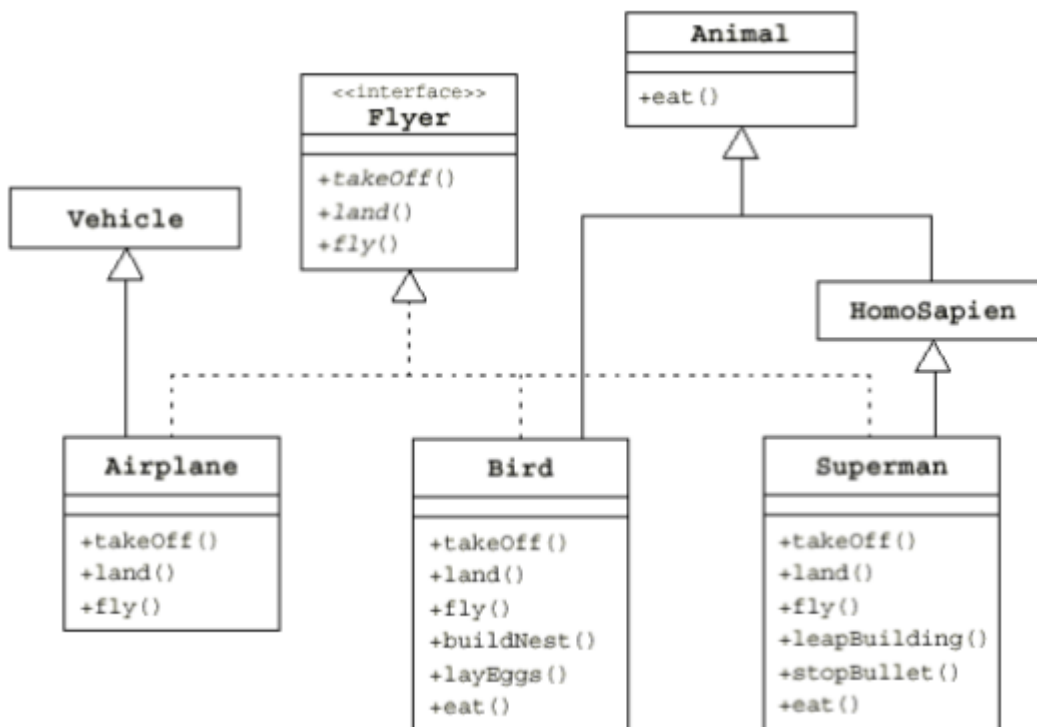
Bird
 Burung membuat sarang
 Burung mencari makan
 Burung terbang
 Burung kembali pulang
 Burung bertelur

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

```

3_index.php x
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>
31
32 <?php echo "Jadi, membutuhkan bahan bakar sebanyak " .
33 $airplane2->calcFuelNeeds() .
34 " Liter" .
35 "<br>"; ?>
36
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>

```

```

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

```

```

32
33     public function takeOff()
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         $trip = $this->calcTripDistance();
52
53         return ceil($fuel /= $trip);
54     }
55 }
56
57 class Superman2 extends Homosapiens implements Flyer
58 {
59     public function takeOff()
60     {
61         return "$this->name mengejar Batman";
62     }
63
64     public function land()
65     {
66         return "$this->name melawan Batman";
67     }
68
69     public function fly()
70     {
71         return "$this->name melancarkan pukulan";
72     }
73
74     public function leapBuilding()
75     {
76         return "Batman terpentak menabrak bangunan pencakar langit";
77     }
78
79     public function stopBullet()
80     {
81         return "Polisi menembaki $this->name namun ditangkis";
82     }
83 }
84
85 $harimau = new Animal("Harimau");
86 $manusia = new Homosapiens("Núñez");
87 $airplane2 = new Airplane2(20000, "Batik Air");
88 $superman2 = new Superman2("Superman");

```

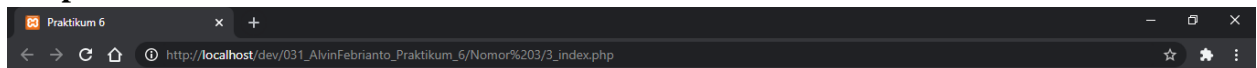
```
3b.php x
1  <?php
2
3  abstract class Vehicle
4  {
5      private $load = 0;
6      protected $maxLoad = 0,
7          $name;
8
9      protected function __construct($maxLoad, $name)
10     {
11         $this->$maxLoad = $maxLoad;
12         $this->$name = $name;
13     }
14
15     public function getload()
16     {
17         return $this->load;
18     }
19
20     public function getMaxLoad()
21     {
22         echo "Maksimal muatan " . $this->name . " ";
23         return $this->maxLoad;
24     }
25
26     public function addBox($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     abstract public function calcFuelNeeds();
38
39     protected function calcFuelEfficiency()
40     {
41         $range = 50000000;
42         $range /= $this->load;
43         return $range;
44     }
45
46     protected function calcTripDistance()
47     {
48         return 500;
49     }
50 }
```

```

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

```

- **Output**



Praktikum 6

Nomor 3

Harimau sedang makan
Núñez sedang makan

Maksimal muatan Batik Air 20000 kg

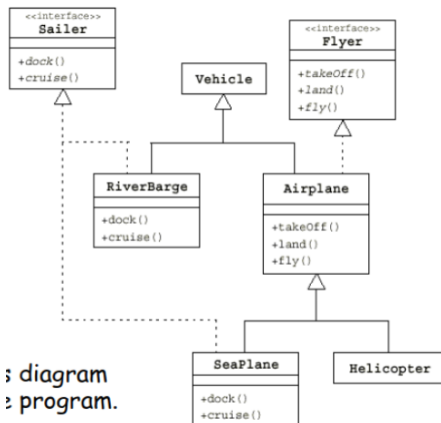
Batik Air menambah muatan sebesar 5000 kg
Batik Air menambah muatan sebesar 7000 kg
Batik Air menambah muatan sebesar 3000 kg
Batik Air menambah muatan sebesar 4000 kg
Batik Air lepas landas
Batik Air dalam perjalanan
Batik Air mendarat
Jadi, membutuhkan bahan bakar sebanyak 6 Liter

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



UML diagram
for program.

• Source Code

```

4_index.php
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             <h4 class="text-center"><strong>Nomor 4</strong></h4>
19             <br><br>
20             <div class="col-4 mx-auto border p-2 mt-2">
21                 <b><?= $riverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
22                 <?= $riverBarge2->addBox(12000) . ' kg'; ?> <br>
23                 <?= $riverBarge2->addBox(14000) . ' kg'; ?> <br>
24                 <?= $riverBarge2->addBox(1000) . ' kg'; ?> <br>
25                 <?= $riverBarge2->addBox(3000) . ' kg'; ?> <br>
26                 <?= $riverBarge2->dock(); ?> <br>
27                 <?= $riverBarge2->cruise(); ?> <br>
28                 <?php
29                     echo "Jadi, membutuhkan bahan bakar sebanyak " . $riverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
30                 ?>
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     </div>
58 </body>
59 </html>
  
```

```
4a.php x
1 <?php
2
3 require_once '4b.php';
4 require_once '4c.php';
5
6 class RiverBarge2 extends Vehicle implements Sailer {
7     public function __construct($maxLoad, $name) {
8         $this->maxLoad = $maxLoad;
9         $this->name = $name;
10    }
11
12    public function calcFuelNeeds() {
13        $fuel = $this->calcFuelEfficiency();
14        $trip = $this->calcTripDistance();
15
16        return ceil($fuel /= $trip);
17    }
18
19    public function dock() {
20        return $this->name . ' berada di dermaga';
21    }
22
23    public function cruise() {
24        return $this->name . ' sedang berlayar';
25    }
26 }
27
28 class Airplane2 implements Flyer {
29     public function takeOff() {
30         return 'Pesawat lepas landas';
31     }
32     public function land() {
33         return 'Pesawat mendarat';
34     }
35     public function fly() {
36         return 'Pesawat dalam perjalanan';
37     }
38 }
39
40 class SeaPlane extends Vehicle implements Sailer {
41     public function __construct($maxLoad, $name) {
42         $this->maxLoad = $maxLoad;
43         $this->name = $name;
44     }
45
46     public function calcFuelNeeds() {
47         $fuel = $this->calcFuelEfficiency();
48         $trip = $this->calcTripDistance();
49
50         return ceil($fuel /= $trip);
51     }
52
53     public function dock() {
54         return $this->name . ' berada di dermaga';
55     }
56
57     public function cruise() {
58         return $this->name . ' sedang berlayar';
59     }
60 }
```

```

60
61     public function takeOff() {
62         return $this->name . ' lepas landas';
63     }
64
65     public function land() {
66         return $this->name . ' mendarat';
67     }
68
69     public function fly() {
70         return $this->name . ' dalam perjalanan';
71     }
72 }
73
74 class Helicopter extends Vehicle {
75     public function __construct($maxLoad, $name) {
76         $this->maxLoad = $maxLoad;
77         $this->name = $name;
78     }
79
80     public function calcFuelNeeds() {
81         $fuel = $this->calcFuelEfficiency();
82         $trip = $this->calcTripDistance();
83
84         return ceil($fuel /= $trip);
85     }
86     public function takeOff() {
87         return $this->name . ' lepas landas';
88     }
89
90     public function land() {
91         return $this->name . ' mendarat';
92     }
93
94     public function fly() {
95         return $this->name . ' dalam perjalanan';
96     }
97 }
98
99 $riverBarge2 = new RiverBarge2(30000, 'Storeum');
100 $seaPlane = new SeaPlane(20000, 'Falcon');
101 $helicopter = new Helicopter(10000, 'Hurricane');

```

```
4b.php x
1 <?php
2
3 abstract class Vehicle {
4     private $load = 0;
5     protected $maxLoad = 0, $name;
6
7     protected function __construct($maxLoad, $name) {
8         $this->$maxLoad = $maxLoad;
9         $this->$name = $name;
10    }
11
12    public function getLoad() {
13        return $this->load;
14    }
15
16    public function getMaxLoad() {
17        echo 'Maksimal muatan ' . $this->name . ' ' ;
18        return $this->maxLoad;
19    }
20
21    public function addBox($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        }else {
26            $this->load += $weight;
27            echo "$this->name menambah muatan sebesar $weight";
28        }
29    }
30
31    abstract public function calcFuelNeeds();
32
33    protected function calcFuelEfficiency() {
34        $range = 50000000;
35        $range /= $this->load;
36        return $range;
37    }
38
39    protected function calcTripDistance() {
40        return 500;
41    }
42 }
```



```

1  <?php
2
3  interface Flyer {
4      public function takeOff();
5      public function land();
6      public function fly();
7  }
8
9  interface Sailer {
10     public function dock();
11     public function cruise();
12 }

```

• 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.