

LAPORAN UTS
PRAKTIKUM 6



Oleh :

Handika Dio Pradana

21091397021

2021A

MANAJEMEN INFORMATIKA UNIVERSITAS NEGERI
SURABAYA

PROGRAM VOKASI

2021

1. Source Code

```
1 <?php
2
3 require_once '../No1/No1abstract.php';
4
5 class Truck extends Vehicle {
6     public function __construct($maxLoad, $name)
7     {
8         $this->maxLoad = $maxLoad;
9         $this->name = $name;
10    }
11
12    public function calcFuelNeeds()
13    {
14        $fuel = $this->calcFuelEfficiency();
15        $trip = $this->calcTripDistance();
16
17        return ceil($fuel / $trip);
18    }
19 }
20
21 class RiverBarge extends Vehicle {
22     public function __construct($maxLoad, $name)
23     {
24         $this->maxLoad = $maxLoad;
25         $this->name = $name;
26     }
27
28     public function calcFuelNeeds()
29     {
30         $fuel = $this->calcFuelEfficiency();
31         $trip = $this->calcTripDistance();
32
33         return ceil($fuel / $trip);
34     }
35 }
36 $truck = new Truck(20000, 'Truk');
37 $riverBarge = new RiverBarge(12000, 'Perahu');
```

```

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     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 }

```

```

1 </php>
2 require_once 'hal.php';
3 }
4
5 <noctype.html>
6 <html lang="id">
7
8 <head>
9 <!-- bootstrap CSS -->
10 <link href="https://cdn.jsdelivr.net/npm/bootstrap@1.3/dist/css/bootstrap.min.css" rel="stylesheet"
11 integrity="sha384-Tk4iVvHedw35/6Lr2O4+is1y3q3838Lr16Zkr3yf2W714j8Z161u1Zd3R" crossorigin="anonymous">
12
13 <title>Modul Praktikum 6</title>
14 </head>
15
16 <body>
17 <div class="container">
18 <div>
19 <div class="text-center">Modul Praktikum 6</div>
20 <div class="row">
21 <div class="col-6 mx-auto border p-3 mt-2">
22 <div class="text-center"><strong>Soul NO. 1</strong></div>
23 <div>
24 <div>
25 <div>
26 <div>
27 <div>
28 <div>
29 <div>
30 <div>
31 <div>
32 <div>
33 <div>
34 <div>
35 <div>
36 <div>
37 <div>
38 <div>
39 <div>
40 <div>
41 <div>
42 <div>
43 <div>
44 <div>
45 <div>
46 <div>
47 <div>
48 <div>
49 <div>
50 <div>
51 <div>

```

Ouput :

Soal No.1

Maksimal muatan Truk 18000 kg

Truk menambah muatan sebesar 2000 kg

Truk menambah muatan sebesar 7000 kg

Truk menambah muatan sebesar 9000 kg

Jadi, Butuh Bahan Bakar sebanyak 6 Liter

Maksimal muatan Tongkang Sungai 20000 kg

Tongkang Sungai menambah muatan sebesar 5000 kg

Tongkang Sungai menambah muatan sebesar 7000 kg

Tongkang Sungai menambah muatan sebesar 8000 kg

Jadi, Butuh Bahan Bakar sebanyak 5 Liter

Analisa :

Program tersebut merupakan implementasi kelas abstrak dari kelas kendaraan. Metode `calcFuelNeeds` digunakan untuk menghitung konsumsi bahan bakar. Metode abstrak ditempatkan sebagai kelas induk dari kelas Kendaraan dan dipanggil oleh kelas turunannya (kelas Truk dan kelas RiverBarge) yang mengembalikan nilai yang dihasilkan dari pembagian dua metode (`calcFuelEfficiency` dan `calcTripDistance`).

2. Source Code

[illegible]

```

1  <?php
2
3  require_once '../No2/No2interface.php';
4
5  class Airplane implements Flyer {
6      public function takeOff() {
7          return 'Helicopter lepas landas';
8      }
9
10     public function land() {
11         return 'Helicopter mendarat';
12     }
13
14     public function fly() {
15         return 'Helicopter dalam perjalanan';
16     }
17 }
18
19 class Bird implements Flyer {
20     public function takeOff() {
21         return 'Elang mencari makan';
22     }
23
24     public function land() {
25         return 'Elang keebali pulang';
26     }
27
28     public function fly() {
29         return 'Elang terbang';
30     }
31
32     public function buildNest() {
33         return 'Elang membuat sarang';
34     }
35
36     public function layEggs() {
37         return 'Elang bertelur';
38     }
39 }
40
41 class Superman implements Flyer {
42     public function takeOff() {
43         return 'superman mengejar batman';
44     }
45
46     public function land() {
47         return 'superman melawan batman';
48     }
49
50     public function fly() {
51         return 'superman melancarkan pukulan';
52     }
53
54     public function leapBuilding() {
55         return 'batman terpelantai sonabrak bangunan pencakar langit';
56     }
57
58     public function stopBullet() {
59         return 'Polisi menembaki superman namun ditangkis';
60     }
61 }
62
63 $airplane = new Airplane;
64 $bird = new Bird;
65 $superman = new Superman;

```

Output:

Superman

Superman melawan Batman
Superman mengejar Batman
Superman melancarkan pukulan
Batman terpengal 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: Program ini merupakan implementasi dari Polymorphism dengan menggunakan Interface Flyer. Oleh karena itu, setiap kelas yang mengimplementasikan antarmuka Flyer harus memiliki metode peluncuran, pendaratan, pendaratan, dan penerbangan.

3. Source Code

```
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 }
```



```

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 = 500000000;
35         $range /= $this->load;
36         return $range;
37     }
38
39     protected function calcTripDistance() {
40         return 500;
41     }
42 }

```

```

1  </php
2  require_once 'Not.php';
3  ?>
4
5  <!DOCTYPE html>
6  <html lang="id">
7
8  <head>
9      <!-- Bootstrap Follow link (ctrl + click)
10     <link href="https://cdn.jsdelivr.net/npm/bootstrap@1.3/dist/css/bootstrap.min.css" rel="stylesheet"
11     integrity="sha384-1hne4k48q781VhF1dvKuhfTAU6au108T194wHFTJDbeC60u1080qy12QvZ6jIM3" crossorigin="anonymous">
12
13     <title>Modul Praktikum 6</title>
14 </head>
15 <body>
16     <div class="container">
17         <div class="text-center">Modul Praktikum 6</div>
18         <div class="row">
19             <div class="col-5 mx-auto border p-3 mt-2">
20                 <div class="text-center"><strong>Soal NO 3</strong></div>
21                 <br><br>
22                 <?=> $sings->eat(); ?> <br>
23                 <?=> $namusia->eat(); ?> <br>
24                 <br>
25
26                 <b><?=> $airplane2->getMaxload() . ' kg'; ?> <br></b>
27                 <?=> $airplane2->addBox(6000) . ' kg'; ?> <br>
28                 <?=> $airplane2->addBox(8000) . ' kg'; ?> <br>
29                 <?=> $airplane2->addBox(2000) . ' kg'; ?> <br>
30                 <?=> $airplane2->addBox(3000) . ' kg'; ?> <br>
31                 <?=> $airplane2->takeOff(); ?> <br>
32                 <?=> $airplane2->fly(); ?> <br>
33                 <?=> $airplane2->land(); ?> <br>
34
35                 <?=> $airplane2->land(); ?> <br>
36
37             </div>
38
39             <div class="col-5 mx-auto border p-3 mt-2">
40                 <div class="text-center"><strong>Soal NO 4</strong></div>
41                 <br><br>
42                 <?=> $superman2->eat(); ?> <br>
43                 <?=> $superman2->land(); ?> <br>
44                 <?=> $superman2->takeOff(); ?> <br>
45                 <?=> $superman2->fly(); ?> <br>
46                 <?=> $superman2->slapbullding(); ?> <br>
47                 <?=> $superman2->stopBullet(); ?> <br>
48                 <br>
49             </div>
50         </div>
51     </div>
52 </body>
53 </html>

```

```

1 <?php
2
3 require_once '../hal/halabstract.php';
4 require_once '../hal/halinterface.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 $this->name . ' sedang makan';
18    }
19 }
20
21 class Monopians extends Animal {}
22
23 class Airplane2 extends Vehicle implements Flyer
24 {
25     public function __construct($maxload, $name)
26     {
27         $this->maxload = $maxload;
28         $this->name = $name;
29     }
30
31     public function takeOff()
32     {
33         return $this->name lepas landas;
34     }
35
36     public function land()
37     {
38         return $this->name mendarat;
39     }
40
41     public function fly()
42     {
43         return $this->name dalam perjalanan;
44     }
45
46     public function calcFuelNeeds()
47     {
48         $fuel = $this->calcFuelEfficiency();
49         $strip = $this->calcTripDistance();
50
51         return ceil($fuel / $strip);
52     }
53 }
54
55 class Superman2 extends Monopians implements Flyer
56 {
57     public function takeOff()
58     {
59         return $this->name mengendarai pesawat;
60     }
61
62     public function land()
63     {
64         return $this->name mendarat sebagai Batman;
65     }
66
67     public function fly()
68 
```

```

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 terpelant menabrak bangunan pencakar langit";
77     }
78
79     public function stopBullet()
80     {
81         return "Polisi menembaki $this->name namun ditangkis";
82     }
83 }
84
85 $sings = new Animal('Sings');
86 $manusia = new Homosapiens('Refika');
87 $airplane2 = new Airplane2(25000, 'lion Air');
88 $superman2 = new Superman2('Superman');

```

Ouput :

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 terpelant menabrak bangunan pencakar langit
Polisi menembaki Superman namun ditangkis

Analisa : Program ini memiliki antarmuka selebaran dan kelas kendaraan abstrak. Kelas Pesawat merupakan implementasi dari antarmuka Flyer dan turunan dari Kendaraan. Oleh karena itu, kelas Airplane harus memiliki metode untuk menghitung Kebutuhan Bahan Bakar, lepas landas, mendarat, dan terbang. Implementasi kelas burung Berasal dari Flyer dan berasal dari Animal, sehingga memiliki cara lepas landas, mendarat, terbang, dan makan. Kelas Superman berasal dari Homosapiens. Homosapiens juga berasal dari Animal dan merupakan implementasi dari antarmuka Flyer. Kelas Superman meliputi cara makan, cara lepas landas, cara mendarat, dan cara terbang.

4. Source Code

```
1  <?php
2
3  require_once '../No4/No4abstract.php';
4  require_once '../No4/No4interface.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 }
```

```

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
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, 'Atomic');
100 $seaPlane = new SeaPlane(20000, 'Titanic');
101 $helicopter = new Helicopter(10000, 'Brooklyn');

```

```

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

```



```

1 </php>
2     require_once 'mod.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@1.3/dist/css/bootstrap.min.css" rel="stylesheet"
11        integrity="sha384-1m0dDgkzFtTlYfLdKubHrTAUkRMTZBnH1j0PDCXMSUoCng12Qv76JDeI" crossorigin="anonymous">
12
13    <title>Modul Praktikum 6</title>
14 </head>
15 <body>
16     <div class="text-center">Modul praktikum 6/</div>
17     <div class="container">
18         <div class="row">
19             <div class="text-center"><strong>Soal NO 4:</strong></div></div>
20             <br>
21             <div class="col-4 mx-auto border p-2 mt-2">
22                 <b><?=$RiverCargo->getMaxLoad() . " kg"; ?></b><br>
23                 <?=$RiverCargo->addBox(1000) . " kg"; ?><br>
24                 <?=$RiverCargo->addBox(1000) . " kg"; ?><br>
25                 <?=$RiverCargo->addBox(2000) . " kg"; ?><br>
26                 <?=$RiverCargo->addBox(4000) . " kg"; ?><br>
27                 <?=$RiverCargo->dock(); ?><br>
28                 <?=$RiverCargo->cruise(); ?><br>
29             </div>
30             <div class="col-4 mx-auto border p-2 mt-2">
31                 <p><?=$SeaPlane->getMaxLoad() . " kg"; ?></p><br>
32                 <?=$SeaPlane->addBox(1000) . " kg"; ?><br>
33                 <?=$SeaPlane->addBox(9000) . " kg"; ?><br>
34                 <?=$SeaPlane->dock(); ?><br>
35                 <?=$SeaPlane->cruise(); ?><br>
36             </div>
37             <div class="col-4 mx-auto border p-2 mt-2">
38                 <p><b><?=$Helicopter->getMaxLoad() . " kg"; ?></b></p><br>
39                 <?=$Helicopter->addBox(7000) . " kg"; ?><br>
40                 <?=$Helicopter->addBox(2000) . " kg"; ?><br>
41                 <?=$Helicopter->takeOff(); ?><br>
42                 <?=$Helicopter->fly(); ?><br>
43                 <?=$Helicopter->land(); ?><br>
44                 <?=$Helicopter->calcFuelNeeds() . " Liter". "<br>" ;
45             </div>
46         </div>
47     </div>
48 </body>
49 </html>
```



```

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 }

```

Ouput :

Maksimal muatan Atomic 40000 kg Atomic menambah muatan sebesar 15000 kg Atomic menambah muatan sebesar 10000 kg Atomic menambah muatan sebesar 8000 kg Atomic menambah muatan sebesar 2000 kg Atomic berada di dermaga Atomic sedang berlayar Jadi, Butuh Bahan Bakar sebanyak 3 Liter	Maksimal muatan Titanic 30000 kg Titanic menambah muatan sebesar 15000 kg Titanic menambah muatan sebesar 7000 kg Titanic berada di dermaga Titanic sedang berlayar Titanic lepas landas Titanic dalam perjalanan Titanic mendarat Jadi, Butuh Bahan Bakar sebanyak 5 Liter	Maksimal muatan Brooklyn 15000 kg Brooklyn menambah muatan sebesar 5000 kg Brooklyn menambah muatan sebesar 7000 kg Brooklyn lepas landas Brooklyn dalam perjalanan Brooklyn mendarat Jadi, Butuh Bahan Bakar sebanyak 9 Liter
--	--	---

Analisa :

Program ini merupakan implementasi dari polimorfisme menggunakan antarmuka dan kelas abstrak yang ditunjukkan pada kelas SeaPlane, yang mengimplementasikan antarmuka Sailer, yang diturunkan dari kelas Airplane, yang mengimplementasikan anak-anak Flyer dan Vehicle. Jadi kelas SeaPlane memiliki metode Dock, Cruise, Takeoff, Land, Fly, dan calcFuelNeeds.