

**LAPORAN PEMROGRAMAN BERORIENTASI OBJEK  
UJIAN TENGAH SEMESTER**



Disusun oleh :

Muvidha Fatmawati Putri (21091397057)  
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  <!-- Muvridha Fatmawati Putri - 21091397011 -->
2
3  <?php
4
5  2 references | 2 implementations
   abstract class Vehicle {
6      4 references
7      private $load = 0;
8      protected $maxLoad = 0, $name;
9
10     0 references | 2 overrides
       protected function __construct($maxLoad, $name) {
11         $this->$maxLoad = $maxLoad;
12         $this->$name = $name;
13     }
14
15     0 references | 0 overrides
       public function getLoad() {
16         return $this->load;
17     }
18
19     2 references | 0 overrides
       public function getMaxLoad() {
20         echo 'Maksimal muatan ' . $this->name . ' ' ;
21         return $this->maxLoad;
22     }
23
24     6 references | 0 overrides
       public function addBox($weight) {
25         if ($this->load >= $this->maxLoad) {
26             echo "$this->name menambah muatan sebesar $weight <br>";
27             echo 'Muatan telah penuh tidak bisa menambah lagi';
28         } else {
29             $this->load += $weight;
30             echo "$this->name menambah muatan sebesar $weight";
31         }
32     }
33
34     2 references | 2 overrides
       abstract public function calcFuelNeeds();
35
36     2 references | 0 overrides
       protected function calcFuelEfficiency() {
37         $range = 1000000000;
38         $range /= $this->load;
39         return $range;
40     }
41
42     2 references | 0 overrides
       protected function calcTripDistance() {
43         return 100000;
44     }
45 }
46
47 1 reference | 0 implementations
   class Truck extends Vehicle {
48     1 reference | 0 overrides | prototype
       public function __construct($maxLoad, $name)
49     {
50         $this->maxLoad = $maxLoad;
51         $this->name = $name;
52     }
53
54     2 references | 0 overrides | prototype
       public function calcFuelNeeds()
```

```

54     {
55         $fuel = $this->calcFuelEfficiency();
56         $trip = $this->calcTripDistance();
57     }
58     return ceil($fuel / $trip);
59 }
60 }
61
62 1 reference | 0 implementations
63 class RiverBarge extends Vehicle {
64     1 reference | 0 overrides | prototype
65     public function __construct($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         $trip = $this->calcTripDistance();
76     }
77     return ceil($fuel / $trip);
78 }
79
80 $truck = new Truck(20000, 'Truk');
81 $riverBarge = new RiverBarge(35000, 'Perahu');
82
83 >>
84
85 <!DOCTYPE html>
86 <html lang="id">
87
88 <head>
89     <!-- Bootstrap CSS -->
90     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
91         integrity="sha384-1BmE4kWBq78iYhFldvKuhtAU6auU8t194MrHftjDbrCEXSU1oBoqy12Qv26jIW3" crossorigin="anonymous">
92
93     <title>PBO - Praktikum 6</title>
94 </head>
95
96 <body>
97     <div class="container">
98         <br>
99         <h2 class="text-center">PBO - Praktikum 6</h2>
100         <div class="row">
101             <div class="col-5 mx-auto border p-3 mt-2">
102                 <h4 class="text-center"><strong>Soal 1</strong></h4>
103                 <br><br>
104                 <b><?=$truck->getMaxLoad() . ' kg'; ?></b><br><br>
105                 <?=$truck->addBox(3000) . ' kg'; ?><br>
106                 <?=$truck->addBox(8000) . ' kg'; ?><br>
107                 <?=$truck->addBox(9000) . ' kg'; ?><br>
108
109                 <?php
110                     echo "Jadi, Butuh Bahan Bakar sebanyak " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
111                 ?>
112
113                 <br>
114                 -----
115                 <br>
116                 <b><?=$riverBarge->getMaxLoad() . ' kg'; ?></b><br><br>
117                 <?=$riverBarge->addBox(12000) . ' kg'; ?><br>
118                 <?=$riverBarge->addBox(10000) . ' kg'; ?><br>
119                 <?=$riverBarge->addBox(7000) . ' kg'; ?><br>
120
121                 <?php
122                     echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge->calcFuelNeeds() . ' Liter';
123                 ?>
124             </div>
125         </div>
126     </body>
127
128 </html>

```

- Output :

**Soal 1**

**Maksimal muatan Truk 20000 kg**

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

-----

**Maksimal muatan Perahu 35000 kg**

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

- 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
   interface Flyer {
6      3 references | 3 overrides
       public function takeOff();
7      3 references | 3 overrides
       public function land();
8      3 references | 3 overrides
       public function fly();
9  }
10
11  0 references | 0 implementations
   interface Sailer {
12      0 references | 0 overrides
       public function dock();
13      0 references | 0 overrides
       public function cruise();
14  }
15  1 reference | 0 implementations
   class Airplane implements Flyer {
16      3 references | 0 overrides
       public function takeOff() {
17          return 'Pesawat lepas landas..';
18      }
19
20      3 references | 0 overrides
       public function land() {
21          return 'Pesawat mendarat';
22      }
23
24      3 references | 0 overrides
       public function fly() {
25          return 'Pesawat dalam perjalanan';
26      }
27  }
28
29  1 reference | 0 implementations
   class Bird implements Flyer {
30      3 references | 0 overrides
       public function takeOff() {
31          return 'Burung mencari makan';
32      }
33
34      3 references | 0 overrides
       public function land() {
35          return 'Burung kembali pulang';
36      }
37
38      3 references | 0 overrides
       public function fly() {
39          return 'Burung terbang';
40      }
41
42      1 reference | 0 overrides
       public function buildNest() {
43          return 'Burung membuat sarang';
44      }
45
46      1 reference | 0 overrides
       public function layEggs() {
47          return 'Burung bertelur';
48      }
49  }
```

```

50
51 1 reference | 0 implementations
class Superman implements Flyer {
52     3 references | 0 overrides
    public function takeOff() {
53         return 'Superman mengejar Batman';
54     }
55
56     3 references | 0 overrides
    public function land() {
57         return 'Superman melawan Batman';
58     }
59
60     3 references | 0 overrides
    public function fly() {
61         return 'Superman melancarkan pukulan';
62     }
63
64     1 reference | 0 overrides
    public function leapBuilding() {
65         return 'Batman terpelant menabrak bangunan pencakar langit';
66     }
67
68     1 reference | 0 overrides
    public function stopBullet() {
69         return 'Polisi menembaki superman namun ditangkis';
70     }
71 }
72
73 $airplane = new Airplane;
74 $bird = new Bird;
75 $superman = new Superman;
76 ?>
77
78 <!DOCTYPE html>
79 <html lang="en">
80
81 <head>
82     <!-- Bootstrap CSS -->
83     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
84         integrity="sha384-18mE4kWBq78iYhF1dVkuhF7AU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
85
86     <title>PBO - Praktikum 6</title>
87 </head>
88
89 <body>
90     <div class="container">
91         <br>
92         <h2 class="text-center">PBO - Praktikum 6</h2>
93         <div class="row">
94             <div class="col-5 mx-auto border p-3 mt-2">
95                 <h4 class="text-center"><strong>Soal 2</strong></h4>
96                 <br><br>
97                 <b><?php
98                     echo "Superman";
99                     ?></b> <br>
100                 <?>= $superman->land(); ?> <br>
101                 <?>= $superman->takeOff(); ?> <br>
102                 <?>= $superman->fly(); ?> <br>
103                 <?>= $superman->leapBuilding(); ?> <br>
104                 <?>= $superman->stopBullet(); ?> <br>
105                 <br>
106                 <b><?php
107                     echo "Bird";
108                     ?></b> <br>
109                 <?>= $bird->buildNest(); ?> <br>
110                 <?>= $bird->takeOff(); ?> <br>
111                 <?>= $bird->fly(); ?> <br>
112                 <?>= $bird->land(); ?> <br>
113                 <?>= $bird->layEggs(); ?> <br>
114                 <br>
115                 <b><?php
116                     echo "Airplane";
117                     ?></b> <br>
118                 <?>= $airplane->takeOff(); ?> <br>
119                 <?>= $airplane->fly(); ?> <br>
120                 <?>= $airplane->land(); ?> <br>
121             </div>
122         </div>
123     </div>
124 </body>
125
126 </html>

```

- 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
1  <!-- Muvidha Fatmawati Putri - 21091397011 -->
2  <?php
3
4  interface Flyer {
5      public function takeOff();
6      public function land();
7      public function fly();
8  }
9
10 interface Sailer {
11     public function dock();
12     public function cruise();
13 }
14 <?php
15 Kendaraan kelas abstrak {
16     pribadi $ beban = 0 ;
17     dilindungi $ maxLoad = 0 , $ nama ;
18
19     fungsi yang dilindungi __construct ( $ maxLoad , $ nama ) {
20         $ this -> $ maxLoad = $ maxLoad ;
21         $ ini -> $ nama = $ nama ;
22     }
23
24     fungsi publik getLoad () {
25         kembalikan $ this -> load ;
26     }
27
28     fungsi publik getMaxLoad () {
29         gema 'Maksimal muatan ' . $ ini -> nama . ' ' ;
30         kembalikan $ this -> maxLoad ;
31     }
32
33     fungsi publik addBox ( $ berat ) {
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         } lain {
38             $ this -> load += $ weight ;
39             echo " $ this -> name menambah muatan sebesar $ weight " ;
40         }
41     }
42
43     fungsi publik abstrak calcFuelNeeds () ;
44
45     fungsi yang dilindungi calcFuelEfficiency () {
46         $ kisaran = 50000000 ;
47         $ range /= $ this -> load ;
48         kembali kisaran $ ;
49     }
50
51     fungsi yang dilindungi calcTripDistance () {
52         kembali 500 ;
53     }
54 }
55 class Animal
56 {
57     protected $name;
58
59     public function __construct($name)
60     {
61         $this->name = $name;
62     }
63
64     public function eat()
65     {
66         return $this->name . ' sedang makan';
67     }
68 }
69
70 class Homosapiens extends Animal {}
71
72 class Airplane2 extends Vehicle implements Flyer
73 {
74     public function __construct($maxLoad, $name)
75     {
76         $this->maxLoad = $maxLoad;
77         $this->name = $name;
78     }
79
80     public function takeOff()
81     {
82         return "$this->name lepas landas";
83     }
84
85     public function land()
86     {
87         return "$this->name mendarat";
88     }
89
90     public function fly()
91     {
92         return "$this->name dalam perjalanan";
93     }
94
95     public function calcFuelNeeds()
```



```

96     {
97         $fuel = $this->calcFuelEfficiency();
98         $strip = $this->calcTripDistance();
99
100
101
102         return ceil($fuel / $strip);
103     }
104 }
105
106 class Superman2 extends Homosapiens implements Flyer
107 {
108     public function takeOff()
109     {
110         return "$this->name mengejar Batman";
111     }
112
113     public function land()
114     {
115         return "$this->name melawan Batman";
116     }
117
118     public function fly()
119     {
120         return "$this->name melancarkan pukulan";
121     }
122
123     public function leapBuilding()
124     {
125         return "Batman terpentak menabrak bangunan pencakar langit";
126     }
127
128     public function stopBullet()
129     {
130         return "Polisi menembaki $this->name namun ditangkis";
131     }
132 }
133
134 $singa = new Animal('kucing');
135 $manusia = new Homosapiens('Candra');
136 $airplane2 = new Airplane2(100000, 'okey plane');
137 $superman2 = new Superman2('Superman');
138 ?>
139
140 <!DOCTYPE html>
141 <html lang="id">
142
143 <head>
144     <!-- Bootstrap CSS -->
145     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
146         integrity="sha384-1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrfHfjDbrCEXSU1oBoqy12QvZ6jIW3" crossorigin="anonymous">
147
148     <title>PBO - Praktikum 6</title>
149 </head>
150 <body>
151     <div class="container">
152         <h2 class="text-center">PBO - Praktikum 6</h2>
153         <div class="row">
154             <div class="col-5 mx-auto border p-3 mt-2">
155                 <h4 class="text-center"><strong>Soal 3</strong></h4>
156                 <br>
157                 <?= $singa->eat(); ?> <br>
158                 <?= $manusia->eat(); ?> <br>
159                 <br>
160
161                 <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
162                 <?= $airplane2->addBox(2500) . ' kg'; ?> <br>
163                 <?= $airplane2->addBox(14000) . ' kg'; ?> <br>
164                 <?= $airplane2->addBox(5000) . ' kg'; ?> <br>
165                 <?= $airplane2->addBox(8000) . ' 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

Singa sedang makan  
Andi sedang makan

**Maksimal muatan Garuda Air 17000 kg**

Garuda Air menambah muatan sebesar 2000 kg  
Garuda Air menambah muatan sebesar 4000 kg  
Garuda Air menambah muatan sebesar 2000 kg  
Garuda Air menambah muatan sebesar 9000 kg  
Garuda Air lepas landas  
Garuda Air dalam perjalanan  
Garuda Air 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 - 21091397011 -->
2
3  <?php
4
5  interface Flyer {
6      public function takeOff();
7      public function land();
8      public function fly();
9  }
10
11 interface Sailer {
12     public function dock();
13     public function cruise();
14 }
15 <?php
16
17 abstract class Vehicle {
18     private $load = 0;
19     protected $maxLoad = 0, $name;
20
21     protected function __construct($maxload, $name) {
22         $this->$maxLoad = $maxLoad;
23         $this->$name = $name;
24     }
25
26     public function getLoad() {
27         return $this->load;
28     }
29
30     public function getMaxLoad() {
31         echo 'Maksimal muatan ' . $this->name . ' ' ;
32         return $this->maxLoad;
33     }
34
35     public function addBox($weight) {
36         if ($this->load >= $this->maxLoad) {
37             echo "$this->name menambah muatan sebesar $weight <br>";
38             echo 'Muatan telah penuh tidak bisa menambah lagi';
39         } else {
40             $this->load += $weight;
41             echo "$this->name menambah muatan sebesar $weight";
42         }
43     }
44
45     abstract public function calcFuelNeeds();
46
47     protected function calcFuelEfficiency() {
48         $range = 50000000;
49         $range /= $this->load;
50         return $range;
51     }
52
53     protected function calcTripDistance() {
54         return 500;
55     }
56 }
57 class RiverBarge2 extends Vehicle implements Sailer {
58     public function __construct($maxLoad, $name) {
59         $this->maxLoad = $maxLoad;
60         $this->name = $name;
61     }
62
63     public function calcFuelNeeds() {
```

```

64         $fuel = $this->calcFuelEfficiency();
65         $strip = $this->calcTripDistance();
66
67         return ceil($fuel / $strip);
68     }
69
70     public function dock() {
71         return $this->name . ' berada di dermaga';
72     }
73
74     public function cruise() {
75         return $this->name . ' sedang berlayar';
76     }
77 }
78
79 class Airplane2 implements Flyer {
80     public function takeOff() {
81         return 'Pesawat lepas landas';
82     }
83     public function land() {
84         return 'Pesawat mendarat';
85     }
86     public function fly() {
87         return 'Pesawat dalam perjalanan';
88     }
89 }
90
91 class SeaPlane extends Vehicle implements Sailer {
92     public function __construct($maxLoad, $name) {
93         $this->maxLoad = $maxLoad;
94         $this->name = $name;
95     }
96
97     public function calcFuelNeeds() {
98         $fuel = $this->calcFuelEfficiency();
99         $strip = $this->calcTripDistance();
100
101         return ceil($fuel / $strip);
102     }
103
104     public function dock() {
105         return $this->name . ' berada di dermaga';
106     }
107
108     public function cruise() {
109         return $this->name . ' sedang berlayar';
110     }
111
112     public function takeOff() {
113         return $this->name . ' lepas landas';
114     }
115
116     public function land() {
117         return $this->name . ' mendarat';
118     }
119
120     public function fly() {
121         return $this->name . ' dalam perjalanan';
122     }
123 }
124
125 class Helicopter extends Vehicle {
126     public function __construct($maxLoad, $name) {
127     }
128
129     2 references | 0 overrides | prototype
130     public function calcFuelNeeds() {
131         $fuel = $this->calcFuelEfficiency();
132         $strip = $this->calcTripDistance();
133
134         return ceil($fuel / $strip);
135     }
136
137     0 references | 0 overrides
138     public function takeOff() {
139         return $this->name . ' lepas landas';
140     }
141
142     0 references | 0 overrides
143     public function land() {
144         return $this->name . ' mendarat';
145     }
146
147     0 references | 0 overrides
148     public function fly() {
149         return $this->name . ' dalam perjalanan';
150     }
151 }
152
153 $riverBarge2 = new RiverBarge2(30000, 'Atomic');
154 $seaPlane = new SeaPlane(20000, 'Titanic');
155 $helicopter = new Helicopter(10000, 'Brocklyn');

```

- Output :

#### Soal 4

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

- 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.