

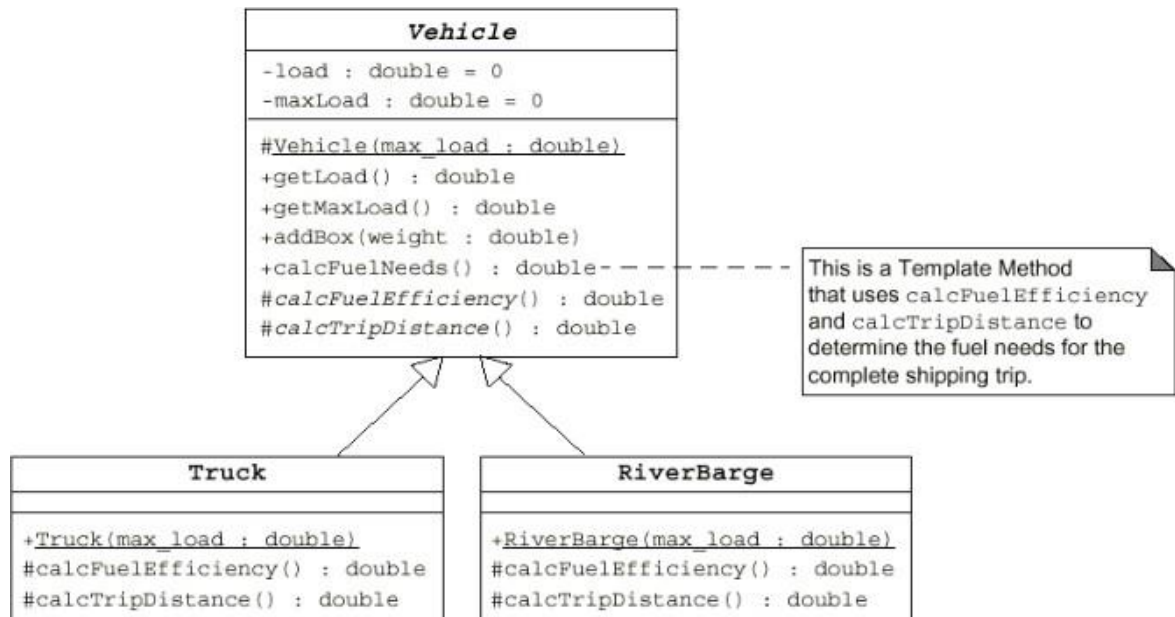
**LAPORAN PRAKTIKUM 6 MATAKULIAH
PEMROGRAMAN BERORIENTASI OBYEK**



**Oleh:
Revina Aurigha Firdaus
21091397003**

**D4 MANAJEMEN INFORMATIKA
UNIVERSITAS NEGERI SURABAYA
2021/2022**

1. Buat program berdasarkan UML berikut



- Source Code

```
1 <!-- Revina Aurigha Firdaus -->
2 <!-- ABSTRACT -->
3
4 <?php
5
6 abstract class Vehicle {
7     private $load = 0;
8     protected $maxLoad = 0, $name;
9
10    protected function __construct($maxLoad, $name) {
11        $this->$maxLoad = $maxLoad;
12        $this->$name = $name;
13    }
14
15    public function getLoad() {
16        return $this->load;
17    }
18
19    public function getMaxLoad() {
20        echo 'Maksimal muatan ' . $this->name . ' ' ;
21        return $this->maxLoad;
22    }
23
24    public function addBox($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 }
```

```

32     }
33
34     abstract public function calcFuelNeeds();
35
36     protected function calcFuelEfficiency() {
37         $range = 1000000000;
38         $range /= $this->load;
39         return $range;
40     }
41
42     protected function calcTripDistance() {
43         return 100000;
44     }
45 }
46
47 class Truck extends Vehicle {
48     public function __construct($maxLoad, $name)
49     {
50         $this->maxLoad = $maxLoad;
51         $this->name = $name;
52     }
53
54     public function calcFuelNeeds()
55     {
56         $fuel = $this->calcFuelEfficiency();
57         $trip = $this->calcTripDistance();
58
59         return ceil($fuel /= $trip);
60     }
61 }
62
63 class RiverBarge extends Vehicle {
64     public function __construct($maxLoad, $name)

```

```

64     public function __construct($maxLoad, $name)
65     {
66         $this->maxLoad = $maxLoad;
67         $this->name = $name;
68     }
69
70     public function calcFuelNeeds()
71     {
72         $fuel = $this->calcFuelEfficiency();
73         $trip = $this->calcTripDistance();
74
75         return ceil($fuel /= $trip);
76     }
77 }
78 $truck = new Truck(13000, 'Truk');
79 $riverBarge = new RiverBarge(42000, 'Perahu');
80
81 ?>
82
83 <!DOCTYPE html>
84 <html lang="id">
85
86 <head>
87     <!-- Bootstrap CSS -->
88     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
89         integrity="sha384-1BmE4kqBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
90
91     <title>Praktikum 6</title>
92 </head>
93
94 <body>
95     <div class="container">

```

```

94 <body>
95 <div class="container">
96 <br>
97 <h2 class="text-center">Praktikum 6</h2>
98 <div class="row">
99 <div class="col-5 mx-auto border p-3 mt-2">
100 <h4 class="text-center"><strong>Soal 1</strong></h4>
101 <br><br>
102 <b><?= $truck->getMaxLoad() . ' kg'; ?> <br></b>
103 <br>
104 <?= $truck->addBox(1200) . ' kg'; ?> <br>
105 <?= $truck->addBox(3000) . ' kg'; ?> <br>
106 <?= $truck->addBox(1100) . ' kg'; ?> <br>
107
108 <?php
109 | echo "Jadi, Butuh Bahan Bakar sebanyak " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
110 | ?>
111 <br>
112 -----
113 <br>
114 <br>
115 <b><?= $riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
116 <br>
117 <?= $riverBarge->addBox(10000) . ' kg'; ?> <br>
118 <?= $riverBarge->addBox(13000) . ' kg'; ?> <br>
119 <?= $riverBarge->addBox(1600) . ' kg'; ?> <br>
120
121 <?php
122 | echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge->calcFuelNeeds() . ' Liter';
123 | ?>
124 </div>
125 </div>
126 </div>
127 </body>
128 </html>

```

- Output

Praktikum 6

Soal 1

Maksimal muatan Truk 13000 kg

Truk menambah muatan sebesar 1200 kg
 Truk menambah muatan sebesar 3000 kg
 Truk menambah muatan sebesar 1100 kg
 Jadi, Butuh Bahan Bakar sebanyak 2 Liter

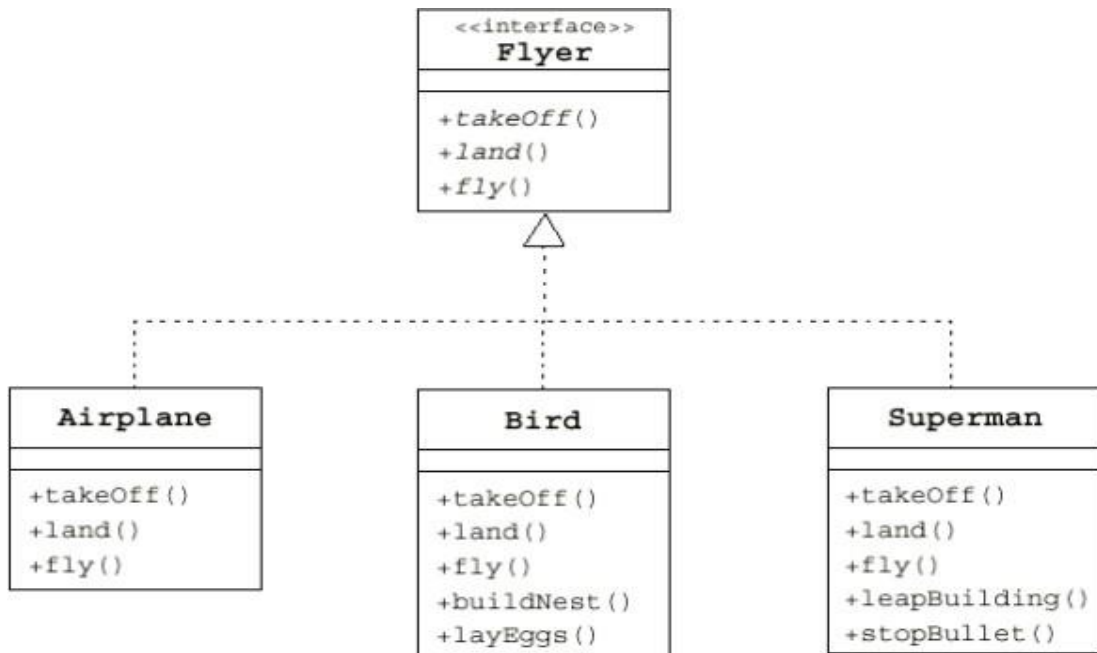
Maksimal muatan Perahu 42000 kg

Perahu menambah muatan sebesar 10000 kg
 Perahu menambah muatan sebesar 13000 kg
 Perahu menambah muatan sebesar 1600 kg
 Jadi, Butuh Bahan Bakar sebanyak 1 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 perahu yang akan mengembalikan nilai yang dihasilkan dari pembagian 2 method yaitu calcFuelEfficiency dan calcTripDistance

2. Buat program berdasarkan UML berikut



- Source Code

```
1 <!-- Revina Aurigha Firdaus -->
2 <!-- INTERFACE -->
3
4 <?php
5
6 interface Flyer {
7     public function takeOff();
8     public function land();
9     public function fly();
10 }
11
12 interface Sailer {
13     public function dock();
14     public function cruise();
15 }
16 class Airplane implements Flyer {
17     public function takeOff() {
18         return 'Pesawat lepas landas..';
19     }
20
21     public function land() {
22         return 'Pesawat mendarat';
23     }
24
25     public function fly() {
26         return 'Pesawat dalam perjalanan';
27     }
28 }
29
30 class Bird implements Flyer {
31     public function takeOff() {
32         return 'Burung mencari makan';
```

```

32     |   return 'Burung menerbangkan';
33     |   }
34
35     |   public function land() {
36     |   |   return 'Burung kembali pulang';
37     |   |   }
38
39     |   public function fly() {
40     |   |   return 'Burung terbang';
41     |   |   }
42
43     |   public function buildNest() {
44     |   |   return 'Burung membuat sarang';
45     |   |   }
46
47     |   public function layEggs() {
48     |   |   return 'Burung bertelur';
49     |   |   }
50     |   }
51
52     |   class Superman implements Flyer {
53     |   |   public function takeOff() {
54     |   |   |   return 'Superman mengejar Batman';
55     |   |   |   }
56
57     |   |   public function land() {
58     |   |   |   return 'Superman melawan Batman';
59     |   |   |   }
60
61     |   |   public function fly() {
62     |   |   |   return 'Superman melancarkan pukulan';
63     |   |   |   }
64     |   }
65
66     |   public function leapBuilding() {
67     |   |   return 'Batman terpelantak menabrak bangunan pencakar langit';
68     |   |   }
69
70     |   public function stopBullet() {
71     |   |   return 'Polisi menembaki superman namun ditangkis';
72     |   |   }
73     |   }
74
75     |   $airplane = new Airplane;
76     |   $bird = new Bird;
77     |   $superman = new Superman;
78     |   }>
79
80     |   <!DOCTYPE html>
81     |   <html lang="en">
82     |   <head>
83     |   |   <!-- Bootstrap CSS -->
84     |   |   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
85     |   |   |   integrity="sha384-1BmE4kWBq78iYhFtdvKuhfTTAU6au08tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
86     |   |
87     |   |   <title>Praktikum 6</title>
88     |   |   </head>
89
90     |   <body>
91     |   |   <div class="container">
92     |   |   |   <br>
93     |   |   |   <h2 class="text-center">Praktikum 6</h2>
94     |   |   |   <div class="row">
95     |   |   |   |   <div class="col-5 mx-auto border p-3 mt-2">

```

```

96      <h4 class="text-center"><strong>Soal 2</strong></h4>
97      <br><br>
98      <b><?php
99          |     echo "Superman";
100         ?></b> <br>
101         <?= $superman->land(); ?> <br>
102         <?= $superman->takeOff(); ?> <br>
103         <?= $superman->fly(); ?> <br>
104         <?= $superman->leapBuilding(); ?> <br>
105         <?= $superman->stopBullet(); ?> <br>
106         <br>
107         <b><?php
108             |     echo "Bird";
109             ?></b> <br>
110             <?= $bird->buildNest(); ?> <br>
111             <?= $bird->takeOff(); ?> <br>
112             <?= $bird->fly(); ?> <br>
113             <?= $bird->land(); ?> <br>
114             <?= $bird->layEggs(); ?> <br>
115             <br>
116             <b><?php
117                 |     echo "Airplane";
118                 ?></b> <br>
119                 <?= $airplane->takeOff(); ?> <br>
120                 <?= $airplane->fly(); ?> <br>
121                 <?= $airplane->land(); ?> <br>
122             </div>
123         </div>
124     </div>
125 </body>
126
127 </html>

```

- Output

Praktikum 6

Soal 2

Superman

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

Bird

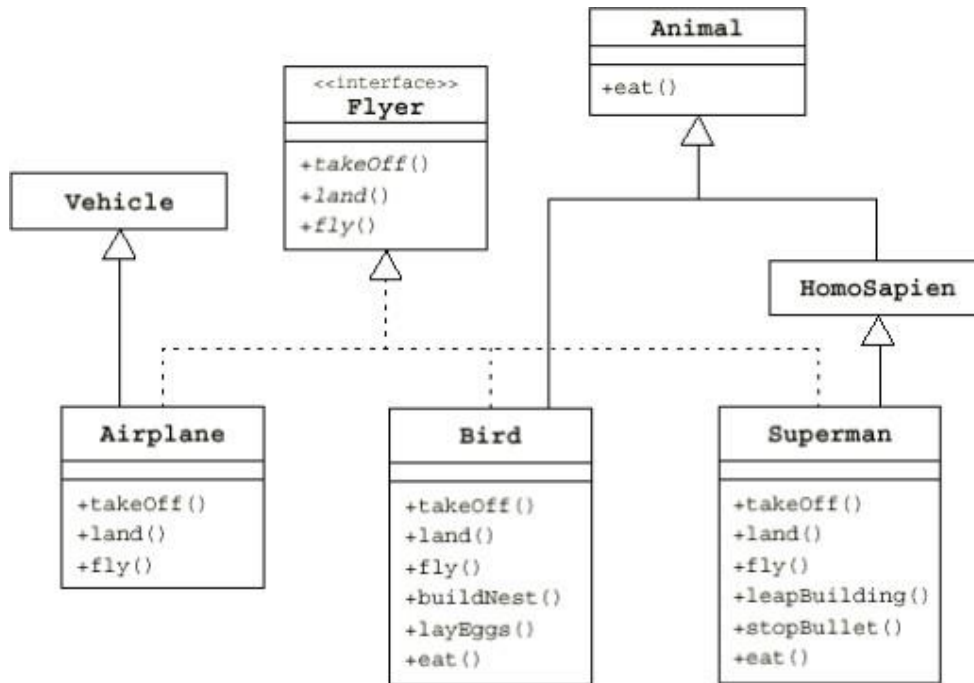
Burung membuat sarang
 Burung mencari makan
 Burung terbang
 Burung kembali pulang
 Burung bertelur

Airplane

Pesawat lepas landas..
 Pesawat dalam perjalanan
 Pesawat mendarat

- Analisis
 Implementasi Polymorphism dengan penggunaan Interface Flyer sehingga semua class dari superman, bird, airplane harus memiliki method takeoff, land, dan fly.

3. Buat program berdasarkan UML berikut



- Source Code

```

C:\xampp\htdocs\praktikum2> cd nos (interface abstract.php)
1  <!-- Revina Aurigha Firdaus (21091397003) -->
2  <!-- INTERFACE & ABSTRACT -->
3
4  <?php
5
6  interface Flyer {
7      public function takeOff();
8      public function land();
9      public function fly();
10 }
11
12 interface Sailer {
13     public function dock();
14     public function cruise();
15 }
16
17
18 abstract class Vehicle {
19     private $load = 0;
20     protected $maxLoad = 0, $name;
21
22     protected function __construct($maxLoad, $name) {
23         $this->$maxLoad = $maxLoad;
24         $this->$name = $name;
25     }
26
27     public function getLoad() {
28         return $this->load;
29     }
30
31     public function getMaxLoad() {
32         echo 'Maksimal muatan ' . $this->name . ' ' ;

```



```

32     echo "Muatan muatan : $this->name ";
33     return $this->maxLoad;
34 }
35
36 public function addBox($weight) {
37     if ($this->load >= $this->maxLoad) {
38         echo "$this->name menambah muatan sebesar $weight <br>";
39         echo 'Muatan telah penuh tidak bisa menambah lagi';
40     } else {
41         $this->load += $weight;
42         echo "$this->name menambah muatan sebesar $weight";
43     }
44 }
45
46 abstract public function calcFuelNeeds();
47
48 protected function calcFuelEfficiency() {
49     $range = 50000000;
50     $range /= $this->load;
51     return $range;
52 }
53
54 protected function calcTripDistance() {
55     return 500;
56 }
57 }
58 class Animal
59 {
60     protected $name;
61
62     public function __construct($name)
63     {
64         $this->name = $name;

```

```

64         $this->name = $name;
65     }
66
67     public function eat()
68     {
69         return $this->name . ' sedang makan';
70     }
71 }
72
73 class Homosapiens extends Animal {}
74
75 class Airplane2 extends Vehicle implements Flyer
76 {
77     public function __construct($maxLoad, $name)
78     {
79         $this->maxLoad = $maxLoad;
80         $this->name = $name;
81     }
82
83     public function takeOff()
84     {
85         return "$this->name lepas landas";
86     }
87
88     public function land()
89     {
90         return "$this->name mendarat";
91     }
92
93     public function fly()
94     {
95         return "$this->name dalam perjalanan";

```

```

96     }
97
98     public function calcFuelNeeds()
99     {
100         $fuel = $this->calcFuelEfficiency();
101         $trip = $this->calcTripDistance();
102
103
104
105         return ceil($fuel /= $trip);
106     }
107 }
108
109 class Superman2 extends Homosapiens implements Flyer
110 {
111     public function takeOff()
112     {
113         return "$this->name mengejar Batman";
114     }
115
116     public function land()
117     {
118         return "$this->name melawan Batman";
119     }
120
121     public function fly()
122     {
123         return "$this->name melancarkan pukulan";
124     }
125
126     public function leapBuilding()
127     {

```

```

C:/xampp/htdocs/praktikum6/6-homosapiens-interface/abstract.php
127     {
128         return "Batman terpenjalar menabrak bangunan pencakar langit";
129     }
130
131     public function stopBullet()
132     {
133         return "Polisi menembaki $this->name namun ditangkis";
134     }
135 }
136
137 $singa = new Animal('kucing');
138 $manusia = new Homosapiens('Revina');
139 $airplane2 = new Airplane2(100000, 'plane');
140 $superman2 = new Superman2('Superman');
141 ?>
142
143 <!DOCTYPE html>
144 <html lang="id">
145
146 <head>
147     <!-- Bootstrap CSS -->
148     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
149     integrity="sha384-18mE4kbWq781YhF1dvKuhfTAU6auU8tT94WwHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
150
151     <title>Praktikum 6</title>
152 </head>
153 <body>
154     <div class="container">
155         <h2 class="text-center">Praktikum 6</h2>
156         <div class="row">
157             <div class="col-5 mx-auto border p-3 mt-2">
158                 <h4 class="text-center"><strong>Soal 3</strong></h4>
159                 <div class="form-group">

```

```

159 <br><br>
160 <?= $singa->eat(); ?> <br>
161 <?= $manusia->eat(); ?> <br>
162 <br>
163
164 <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
165 <?= $airplane2->addBox(1000) . ' kg'; ?> <br>
166 <?= $airplane2->addBox(1300) . ' kg'; ?> <br>
167 <?= $airplane2->addBox(1400) . ' kg'; ?> <br>
168 <?= $airplane2->addBox(1600) . ' kg'; ?> <br>
169 <?= $airplane2->takeOff(); ?> <br>
170 <?= $airplane2->fly(); ?> <br>
171 <?= $airplane2->land(); ?> <br>
172
173 <?php
174 |     echo "Jadi, Butuh Bahan Bakar sebanyak " . $airplane2->calcFuelNeeds() . ' Liter'. '<br>';
175 | ?>
176 <br>
177 <?= $superman2->eat(); ?> <br>
178 <?= $superman2->land(); ?> <br>
179 <?= $superman2->takeOff(); ?> <br>
180 <?= $superman2->fly(); ?> <br>
181 <?= $superman2->leapBuilding(); ?> <br>
182 <?= $superman2->stopBullet(); ?> <br>
183 </div>
184 </div>
185 </div>
186 </body>
187
188 </html>

```

- Output

Praktikum 6

Soal 3

kucing sedang makan
Revina sedang makan

Maksimal muatan plane 100000 kg

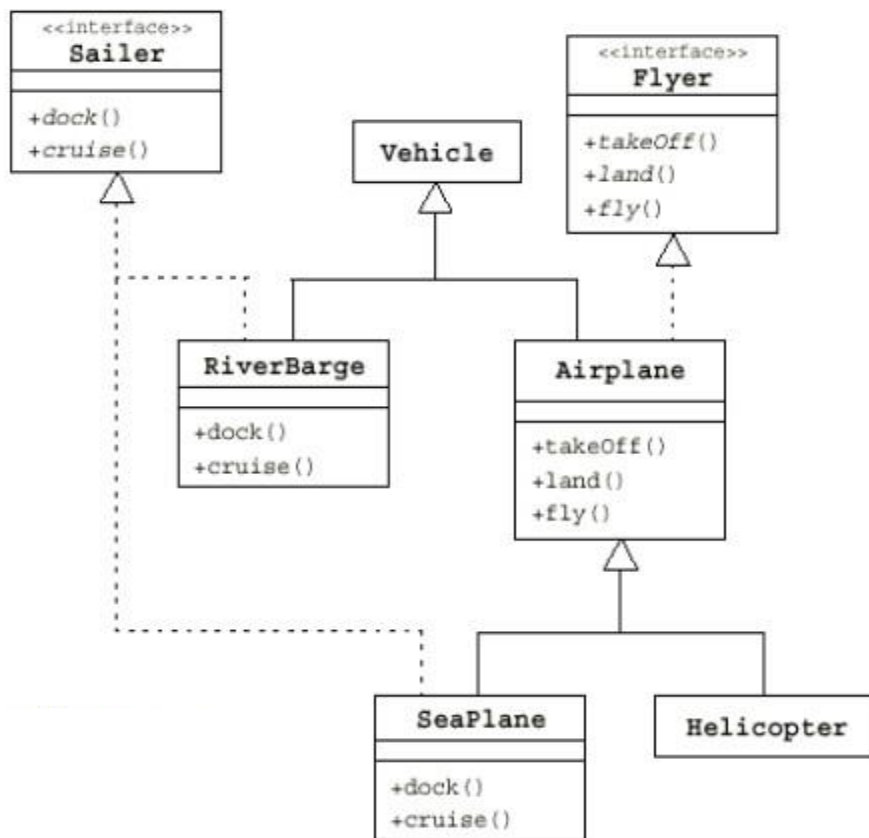
plane menambah muatan sebesar 1000 kg
plane menambah muatan sebesar 1300 kg
plane menambah muatan sebesar 1400 kg
plane menambah muatan sebesar 1600 kg
plane lepas landas
plane dalam perjalanan
plane mendarat
Jadi, Butuh Bahan Bakar sebanyak 19 Liter

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 dari turunan interface Flyer dan turunan dari Vehicle sehingga class Airplane harus memiliki method calcFuelNeeds, takeoff, land, dan fly. Class Bird turunan 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

```

C: > xampp > htdocs > praktikum6 > no4(interface dan abstract).php
1  <!-- Revina Aurigha Firdaus -->
2  <!-- INTERFACE -->
3
4  <?php
5
6  interface Flyer {
7      public function takeOff();
8      public function land();
9      public function fly();
10 }
11
12 interface Sailer {
13     public function dock();
14     public function cruise();
15 }
16 class RiverBarge2 extends Vehicle implements Sailer {
17     public function __construct($maxLoad, $name) {
18         $this->maxLoad = $maxLoad;
19         $this->name = $name;
20     }
21
22     public function calcFuelNeeds() {
23         $fuel = $this->calcFuelEfficiency();
24         $trip = $this->calcTripDistance();
25
26         return ceil($fuel / $trip);
27     }
28
29     public function dock() {
30         return $this->name . ' berada di dermaga';
31     }
32

```

```

33     public function cruise() {
34         return $this->name . ' sedang berlayar';
35     }
36 }
37
38 class Airplane2 implements Flyer {
39     public function takeOff() {
40         return 'Pesawat lepas landas';
41     }
42     public function land() {
43         return 'Pesawat mendarat';
44     }
45     public function fly() {
46         return 'Pesawat dalam perjalanan';
47     }
48 }
49
50 class SeaPlane extends Vehicle implements Sailer {
51     public function __construct($maxLoad, $name) {
52         $this->maxLoad = $maxLoad;
53         $this->name = $name;
54     }
55
56     public function calcFuelNeeds() {
57         $fuel = $this->calcFuelEfficiency();
58         $trip = $this->calcTripDistance();
59
60         return ceil($fuel /= $trip);
61     }
62
63     public function dock() {
64         return $this->name . ' berada di dermaga';

```

```

C: > xampp > htdocs > praktikum6 > no4(interface dan abstract).php
64         return $this->name . ' berada di dermaga';
65     }
66
67     public function cruise() {
68         return $this->name . ' sedang berlayar';
69     }
70
71     public function takeOff() {
72         return $this->name . ' lepas landas';
73     }
74
75     public function land() {
76         return $this->name . ' mendarat';
77     }
78
79     public function fly() {
80         return $this->name . ' dalam perjalanan';
81     }
82 }
83
84 class Helicopter extends Vehicle {
85     public function __construct($maxLoad, $name) {
86         $this->maxLoad = $maxLoad;
87         $this->name = $name;
88     }
89
90     public function calcFuelNeeds() {
91         $fuel = $this->calcFuelEfficiency();
92         $trip = $this->calcTripDistance();
93
94         return ceil($fuel /= $trip);
95     }

```

```

96     public function takeOff() {
97         return $this->name . ' lepas landas';
98     }
99
100    public function land() {
101        return $this->name . ' mendarat';
102    }
103
104    public function fly() {
105        return $this->name . ' dalam perjalanan';
106    }
107 }
108
109 $riverBarge2 = new RiverBarge2(12000, 'Revina');
110 $seaPlane = new SeaPlane(13000, 'Aurigha');
111 $helicopter = new Helicopter(15000, 'Firdaus');
112 ?>
113
114 <!DOCTYPE html>
115 <html lang="en">
116
117     <head>
118         <!-- Bootstrap CSS -->
119         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
120             integrity="sha384-1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
121
122         <title>Praktikum 6</title>
123     </head>
124     <body>
125         <h2 class="text-center">Praktikum 6</h2>
126         <div class="container">
127             <div class="row">

```

```

C:\xampp\htdocs\praktikum6\ No-Interface dan abstract.php
127     <div class="row">
128         <div class="col-4 mx-auto border p-2 mt-2">
129             <h4 class="text-center"><strong>Soal 4</strong></h4>
130             <br><br>
131             <b><?php echo $riverBarge2->getMaxLoad(); </b> <br></b>
132             <?php echo $riverBarge2->addBox(3000); <br>
133             <?php echo $riverBarge2->addBox(13000); <br>
134             <?php echo $riverBarge2->addBox(6000); <br>
135             <?php echo $riverBarge2->addBox(2000); <br>
136             <?php echo $riverBarge2->dock(); <br>
137             <?php echo $riverBarge2->cruise(); <br>
138             <?php
139             <?php echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge2->calcFuelNeeds(); <br>
140             <?php
141         </div>
142         <div class="col-4 mx-auto border p-2 mt-2">
143             <b><?php echo $seaPlane->getMaxLoad(); </b> <br></b>
144             <?php echo $seaPlane->addBox(3000); <br>
145             <?php echo $seaPlane->addBox(7000); <br>
146             <?php echo $seaPlane->dock(); <br>
147             <?php echo $seaPlane->cruise(); <br>
148             <?php echo $seaPlane->takeOff(); <br>
149             <?php echo $seaPlane->fly(); <br>
150             <?php echo $seaPlane->land(); <br>
151             <?php
152             <?php echo "Jadi, Butuh Bahan Bakar sebanyak " . $seaPlane->calcFuelNeeds(); <br>
153             <?php
154         </div>
155         <div class="col mx-auto border p-2 mt-2">
156             <b><?php echo $helicopter->getMaxLoad(); </b> <br></b>
157             <?php echo $helicopter->addBox(2000); <br>
158             <?php echo $helicopter->addBox(15000); <br>

```

```
Run Terminal Help no4(interface dan abstract).php - numpypjproject - Visual Studio Code
no2(interface).php no3 (interface-abstract).php no1(abstract).php no.1.php no.2.php no4(interface dan abstract).php
C: > xampp > htdocs > praktikum6 > no4(interface dan abstract).php
159 <?= $helicopter->takeOff(); ?> <br>
160 <?= $helicopter->fly(); ?> <br>
161 <?= $helicopter->land(); ?> <br>
162 <?php
163 |     echo "Jadi, Butuh Bahan Bakar sebanyak " . $helicopter->calcFuelNeeds() . ' Liter'. '<br>';
164 |     ?>
165 </div>
166 </div>
167 </div>
168 </body>
169
170 </html>
171
172 <!-- ABSTRACT -->
173 <?php
174
175 abstract class Vehicle {
176     private $load = 0;
177     protected $maxLoad = 0, $name;
178
179     protected function __construct($maxLoad, $name) {
180         $this->$maxLoad = $maxLoad;
181         $this->$name = $name;
182     }
183
184     public function getLoad() {
185         return $this->load;
186     }
187
188     public function getMaxLoad() {
189         echo 'Maksimal muatan ' . $this->name . ' ' ;
190         return $this->maxLoad;
```

```
Run Terminal Help no4(interface dan abstract).php - numpypjproject - Visual Studio Code
no2(interface).php no3 (interface-abstract).php no1(abstract).php no.1.php
C: > xampp > htdocs > praktikum6 > no4(interface dan abstract).php
190         return $this->maxLoad;
191     }
192
193     public function addBox($weight) {
194         if ($this->load >= $this->maxLoad) {
195             echo "$this->name menambah muatan sebesar $weight <br>";
196             echo 'Muatan telah penuh tidak bisa menambah lagi';
197         }else {
198             $this->load += $weight;
199             echo "$this->name menambah muatan sebesar $weight";
200         }
201     }
202
203     abstract public function calcFuelNeeds();
204
205     protected function calcFuelEfficiency() {
206         $range = 50000000;
207         $range /= $this->load;
208         return $range;
209     }
210
211     protected function calcTripDistance() {
212         return 500;
213     }
214 }
```

- Output

Praktikum 6

Soal 4

Maksimal muatan Revina 12000 kg Revina menambah muatan sebesar 3000 kg Revina menambah muatan sebesar 13000 kg Revina menambah muatan sebesar 6000 Muatan telah penuh tidak bisa menambah lagi kg Revina menambah muatan sebesar 2000 Muatan telah penuh tidak bisa menambah lagi kg Revina berada di dermaga Revina sedang berlayar Jadi, Butuh Bahan Bakar sebanyak 7 Liter	Maksimal muatan Aurigha 13000 kg Aurigha menambah muatan sebesar 3000 kg Aurigha menambah muatan sebesar 7000 kg Aurigha berada di dermaga Aurigha sedang berlayar Aurigha lepas landas Aurigha dalam perjalanan Aurigha mendarat Jadi, Butuh Bahan Bakar sebanyak 10 Liter	Maksimal muatan Firdaus 15000 kg Firdaus menambah muatan sebesar 2000 kg Firdaus menambah muatan sebesar 15000 kg Firdaus lepas landas Firdaus dalam perjalanan Firdaus mendarat Jadi, Butuh Bahan Bakar sebanyak 6 Liter
---	--	--

- Analisis

Implementasi polymorphism dengan interface dan abstract class yaitu pada class Plane yang turuna dari interface Sailer, class Airplane yang turunan Flyer dan child dari Vehicle sehingga class SeaPlane memiliki method dock, cruise, takeoff, land, fly, dan calcFuelNeeds.