

LAPORAN UTS
PRAKTIKUM 5



Oleh :

Akhmad Ilham Muharram (21091397009)

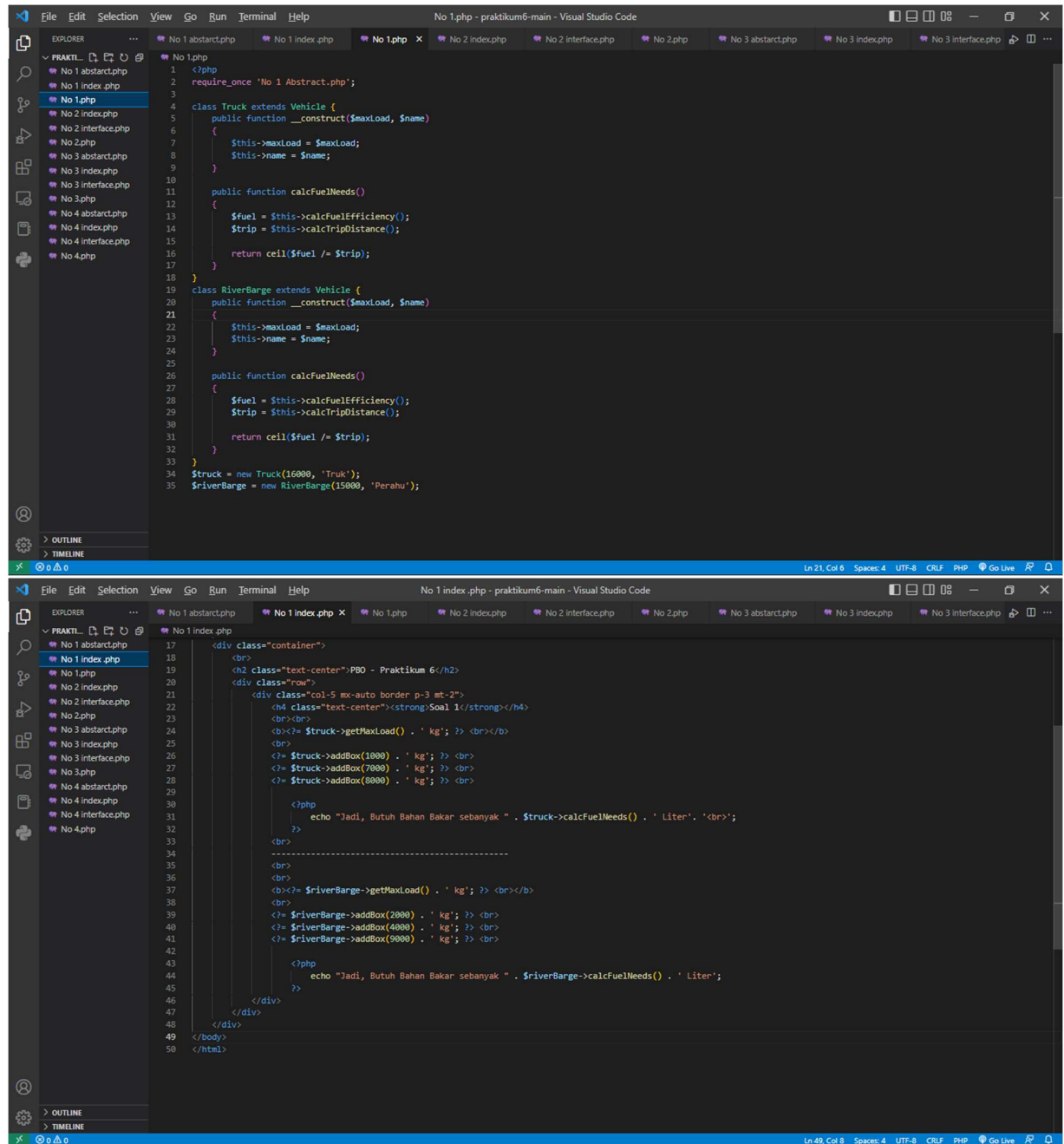
2021A

MANAJEMEN INFORMATIKA UNIVERSITAS NEGERI
SURABAYA

PROGRAM VOKASI

2021

1. Source Code



```
1 <?php
2 require_once 'No 1 Abstract.php';
3
4 class Truck extends Vehicle {
5     public function __construct($maxLoad, $name)
6     {
7         $this->maxLoad = $maxLoad;
8         $this->name = $name;
9     }
10
11     public function calcFuelNeeds()
12     {
13         $fuel = $this->calcFuelEfficiency();
14         $strip = $this->calcTripDistance();
15
16         return ceil($fuel / $strip);
17     }
18 }
19
20 class RiverBarge extends Vehicle {
21     public function __construct($maxLoad, $name)
22     {
23         $this->maxLoad = $maxLoad;
24         $this->name = $name;
25     }
26
27     public function calcFuelNeeds()
28     {
29         $fuel = $this->calcFuelEfficiency();
30         $strip = $this->calcTripDistance();
31
32         return ceil($fuel / $strip);
33     }
34 }
35
36 $truck = new Truck(16000, 'Truk');
37 $riverBarge = new RiverBarge(15000, 'Perahu');
```

```
17 <div class="container">
18     <br>
19     <h2 class="text-center">PBO - Praktikum 6</h2>
20     <div class="row">
21         <div class="col-5 mx-auto border p-3 mt-2">
22             <h4 class="text-center"><strong>Soal 1</strong></h4>
23             <br><br>
24             <b><?=> $truck->getMaxLoad() . ' kg'; ?> <br></b>
25             <br>
26             <b><?=> $truck->addBox(1000) . ' kg'; ?> <br>
27             <b><?=> $truck->addBox(7000) . ' kg'; ?> <br>
28             <b><?=> $truck->addBox(8000) . ' kg'; ?> <br>
29
30             <?php
31                 echo "Jadi, Butuh Bahan Bakar sebanyak " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
32             ?>
33         </div>
34         <br>
35         <br>
36         <b><?=> $riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
37         <br>
38         <b><?=> $riverBarge->addBox(2000) . ' kg'; ?> <br>
39         <b><?=> $riverBarge->addBox(4000) . ' kg'; ?> <br>
40         <b><?=> $riverBarge->addBox(9000) . ' kg'; ?> <br>
41
42         <?php
43             echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge->calcFuelNeeds() . ' Liter';
44         ?>
45     </div>
46 </div>
47
48 </body>
49 </html>
```

```
1 <?php
2 require_once 'No 1 Abstract.php';
3
4 class Truck extends Vehicle {
5     public function __construct($maxload, $name)
6     {
7         $this->maxload = $maxload;
8         $this->name = $name;
9     }
10
11     public function calcFuelNeeds()
12     {
13         $fuel = $this->calcFuelEfficiency();
14         $strip = $this->calcTripDistance();
15
16         return ceil($fuel / $strip);
17     }
18 }
19
20 class RiverBarge extends Vehicle {
21     public function __construct($maxload, $name)
22     {
23         $this->maxload = $maxload;
24         $this->name = $name;
25     }
26
27     public function calcFuelNeeds()
28     {
29         $fuel = $this->calcFuelEfficiency();
30         $strip = $this->calcTripDistance();
31
32         return ceil($fuel / $strip);
33     }
34 }
35
36 $truck = new Truck(16000, 'Truk');
37 $riverBarge = new RiverBarge(15000, 'Perahu');
```

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

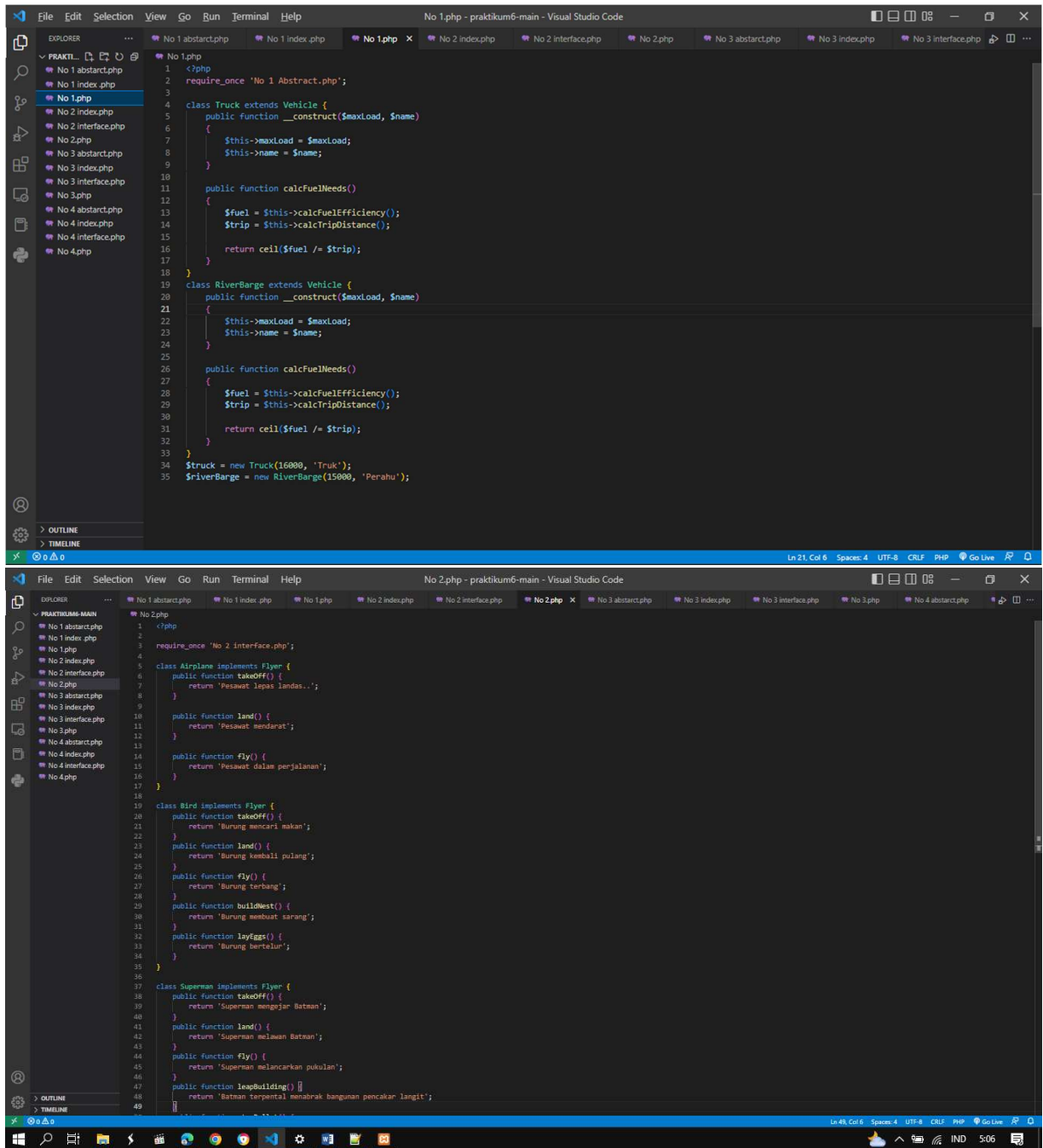
Tongkang Sungai menambah muatan sebesar 8000 kg

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



```
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:

Superman

Superman melawan Batman
Superman mengejar Batman
Superman melancarkan pukulan
Batman terpelant 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 Polymhorphism 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 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 mustan " . $this->name . " ";
18        return $this->maxload;
19    }
20
21    public function addFuel($weight) {
22        if ($this->load >= $this->maxload) {
23            echo "This->name menambah mustan sebesar $weight <br>";
24            echo "Mustan telah penuh tidak bisa menambah lagi!";
25        } else {
26            $this->load += $weight;
27            echo "This->name menambah mustan 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 }
```

The screenshot shows a Visual Studio Code editor window with a project titled "Visual Studio Code - No 3 index.php". The Explorer sidebar on the left lists several files, with "No 3 index.php" selected. The main editor area displays the contents of this file, which is a PHP script. The code starts with a document type declaration and a Bootstrap CSS link. It then sets a title "Praktikum 6" and creates a container with a heading "Shurung-heat()" and a paragraph "Manusia-eat()". Following this is a JavaScript section enclosed in a <script> tag, where a jQuery-like object \$airplane2 is defined with various methods such as getMaxload(), addBox(), stakeOff(), fly(), land(), calcFuelNeeds(), start(), stopBullet(), and stop(). The code uses single quotes for strings and double slashes for comments.

```

<!DOCTYPE html>
<html lang="id">
<head>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@1.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-1m24wUkzgPcwCh7tNGaySVDhMOI9kdYtxvWyyNHM7F4dKIDUXaJuWsgWB2H3yMiN" crossorigin="anonymous">
    <title>Praktikum 6</title>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col-5 mx-auto border p-3 mt-2">
                <div class="text-center"><strong>Soal 3</strong></div>
                <br><br>
                <?php $shurung->heat(); ?> <br>
                <?php $manusia->eat(); ?> <br>
                <br>
                <?php $airplane2->getMaxload() . ' kg'; ?> <br></div>
                <?php $airplane2->addBox(6000) . ' kg'; ?> <br>
                <?php $airplane2->addBox(2000) . ' kg'; ?> <br>
                <?php $airplane2->addBox(7000) . ' kg'; ?> <br>
                <?php $airplane2->addBox(5000) . ' kg'; ?> <br>
                <?php $airplane2->stakeOff(); ?> <br>
                <?php $airplane2->fly(); ?> <br>
                <?php $airplane2->land(); ?> <br>
            </div>
            <div class="col-2">
                <p><b>Jawab:</b></p>
                <pre><code>
                    <?php
                        echo "Jadi, Butuh Bahan Bakar sebanyak ". $airplane2->calcFuelNeeds() . ' Liter'. "<br>";
                    </?php>
                </code>
                </pre>
                <?php $superman2->start(); ?> <br>
                <?php $superman2->land(); ?> <br>
                <?php $superman2->stakeOff(); ?> <br>
                <?php $superman2->fly(); ?> <br>
                <?php $superman2->stopBuildingUp(); ?> <br>
                <?php $superman2->stopBullet(); ?> <br>
                </div>
            </div>
        </div>
    </body>
</html>
```

The screenshot shows the Visual Studio Code interface with a PHP file open. The Explorer sidebar on the left displays a project structure with a folder named 'PRAKTIKUM6-main' containing subfolders 'No 1', 'No 2', 'No 3', and 'No 4'. Each subfolder contains files like 'abstract.php', 'index.php', and 'php'. The file 'No 3 interface.php' is selected and open in the editor. The editor shows the following PHP code:

```

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

```

The status bar at the bottom indicates the current line and column as 'Ln 2, Col 1', the source file as 'Source: 4', the encoding as 'UTF-8', the line endings as 'CRLF', the language as 'PHP', and the 'Go Live' status.


```
1 <?php
2
3 require_once 'No 3 abstract.php';
4 require_once 'No 3 interface.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 $this->name . ' sedang makan';
18    }
19 }
20
21 class Homosapiens extends Animal {}
22
23 class Airplane2 extends Vehicle implements Flyer
24 {
25     public function __construct($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
56
57 class Superman2 extends Homosapiens implements Flyer
58 {
59     public function takeOff()
60     {
61         return $this->name mengajar 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 $burung = new Animal('Burung');
86 $monusia = new Homosapiens('manusia');
87 $airplane2 = new Airplane2(25000, 'Batik Air');
88 $superman2 = new Superman2('Superman');
```

Ouput :

Burung sedang makan
Aransha sedang makan

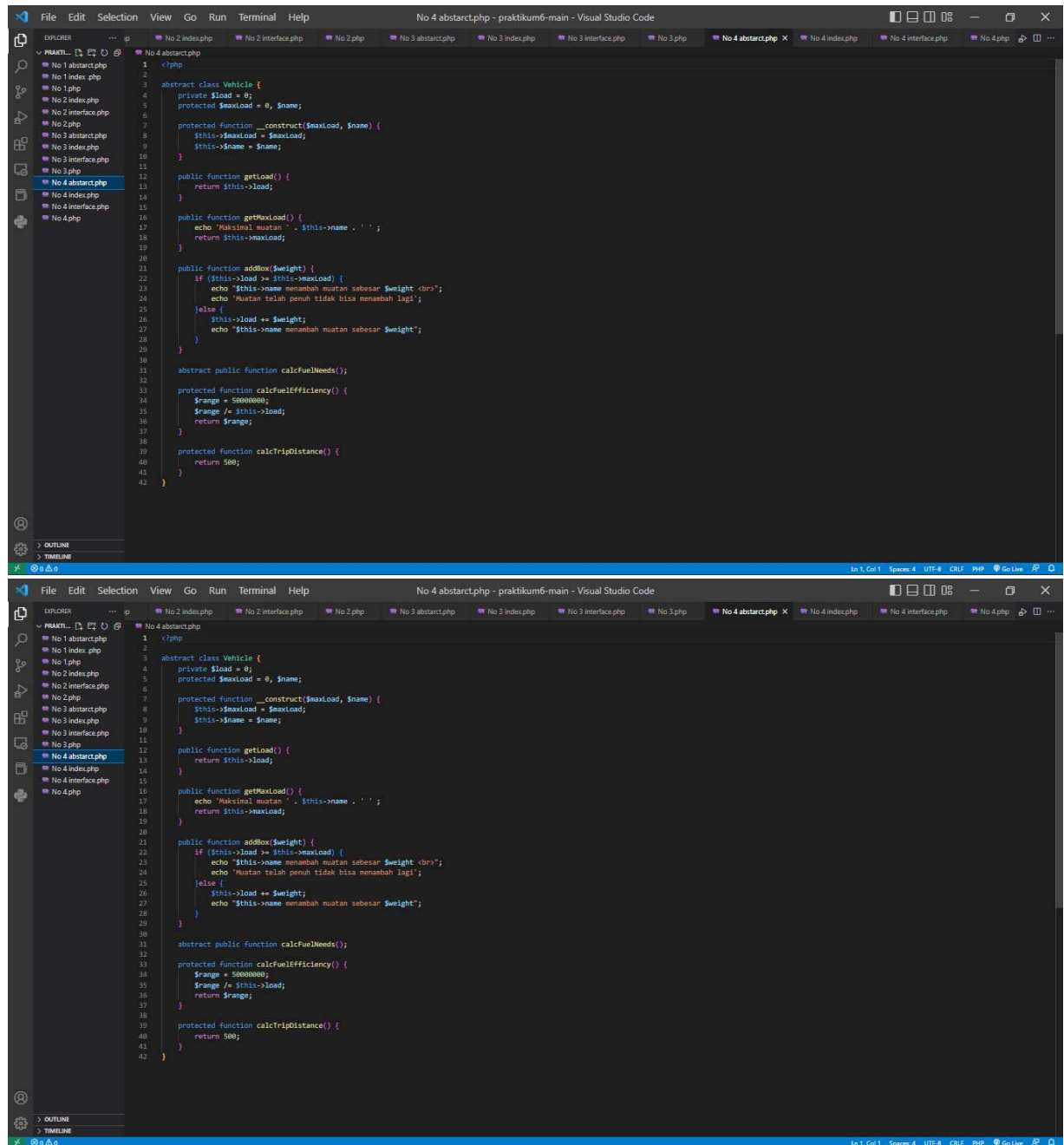
Maksimal muatan Batik Air 25000 kg

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 terpental 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



The image displays two screenshots of a Visual Studio Code editor window, showing the source code for a PHP project. The editor is titled "No 4 abstract.php - praktikum6-main - Visual Studio Code". The Explorer sidebar on the left shows a file structure with folders "PRAKTIKUM6-main" and "No 4", containing files like "No 1 abstract.php", "No 1 index.php", "No 1.php", "No 2 abstract.php", "No 2 index.php", "No 2 interface.php", "No 2.php", "No 3 abstract.php", "No 3 index.php", "No 3 interface.php", "No 3.php", "No 4 abstract.php", "No 4 index.php", "No 4 interface.php", and "No 4.php". The "No 4 abstract.php" file is selected and its content is displayed in the main editor area.

```
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 "Maksimal muatan telah penuh. Tidak bisa menambah lagi!";
24            echo "Maksimal muatan yang bisa ditambah lagi!";
25        } else {
26            $this->load += $weight;
27            echo "Maksimal muatan telah penuh. Tidak bisa menambah lagi!";
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 }
```

[illegible]

```
File Edit Selection View Go Run Terminal Help
No 4.php - praktikum6-main - Visual Studio Code

PRATIUM6-main
  No 1.php
  No 2.php
  No 3.php
  No 4.php
  No 1 abstract.php
  No 2 abstract.php
  No 3 abstract.php
  No 4 abstract.php
  No 1 index.php
  No 2 index.php
  No 3 index.php
  No 4 index.php
  No 1 interface.php
  No 2 interface.php
  No 3 interface.php
  No 4 interface.php

No 4.php
1 <?php
2
3 require_once 'No 4 Abstract.php';
4 require_once 'No 4 Interface.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        $strip = $this->calcTripDistance();
15
16        return ceil($fuel / $strip);
17    }
18
19    public function dock() {
20        return $this->name . ' beranda di dermaga';
21    }
22
23    public function cruise() {
24        return $this->name . ' sedang berlayar';
25    }
26
27    class Airplane2 implements Flyer {
28        public function takeOff() {
29            return 'Pesawat lepas landas';
30        }
31
32        public function land() {
33            return 'Pesawat mendarat';
34        }
35
36        public function fly() {
37            return 'Pesawat dalam perjalanan';
38        }
39    }
40
41    class SeaPlane extends Vehicle implements Sailer {
42        public function __construct($maxload, $name) {
43            $this->maxload = $maxload;
44            $this->name = $name;
45        }
46
47        public function calcFuelNeeds() {
48            $fuel = $this->calcFuelEfficiency();
49            $strip = $this->calcTripDistance();
50
51            return ceil($fuel / $strip);
52        }
53
54        public function dock() {
55            return $this->name . ' beranda di dermaga';
56        }
57
58        public function cruise() {
59            return $this->name . ' sedang berlayar';
60        }
61
62        public function takeOff() {
63            return $this->name . ' lepas landas';
64        }
65
66        public function land() {
67            return $this->name . ' mendarat';
68        }
69
70        public function fly() {
71            return $this->name . ' dalam perjalanan';
72        }
73    }
74
75    class Helicopter extends Vehicle {
76        public function __construct($maxload, $name) {
77            $this->maxload = $maxload;
78            $this->name = $name;
79        }
80
81        public function calcFuelNeeds() {
82            $fuel = $this->calcFuelEfficiency();
83            $strip = $this->calcTripDistance();
84
85            return ceil($fuel / $strip);
86        }
87
88        public function takeOff() {
89            return $this->name . ' lepas landas';
90        }
91
92        public function land() {
93            return $this->name . ' mendarat';
94        }
95
96        public function fly() {
97            return $this->name . ' dalam perjalanan';
98        }
99    }
100
101    $riverBarge2 = new RiverBarge2(30000, 'Atomic');
102    $seaPlane = new SeaPlane(20000, 'Titanic');
103    $helicopter = new Helicopter(10000, 'Brooklyn');
```

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 Brocklyn 15000 kg Brocklyn menambah muatan sebesar 5000 kg Brocklyn menambah muatan sebesar 7000 kg Brocklyn lepas landas Brocklyn dalam perjalanan Brocklyn 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.