LAPORAN UTS

PRAKTIKUM 5

Oleh :

Akhmad Ilham Muharram (21091397009)

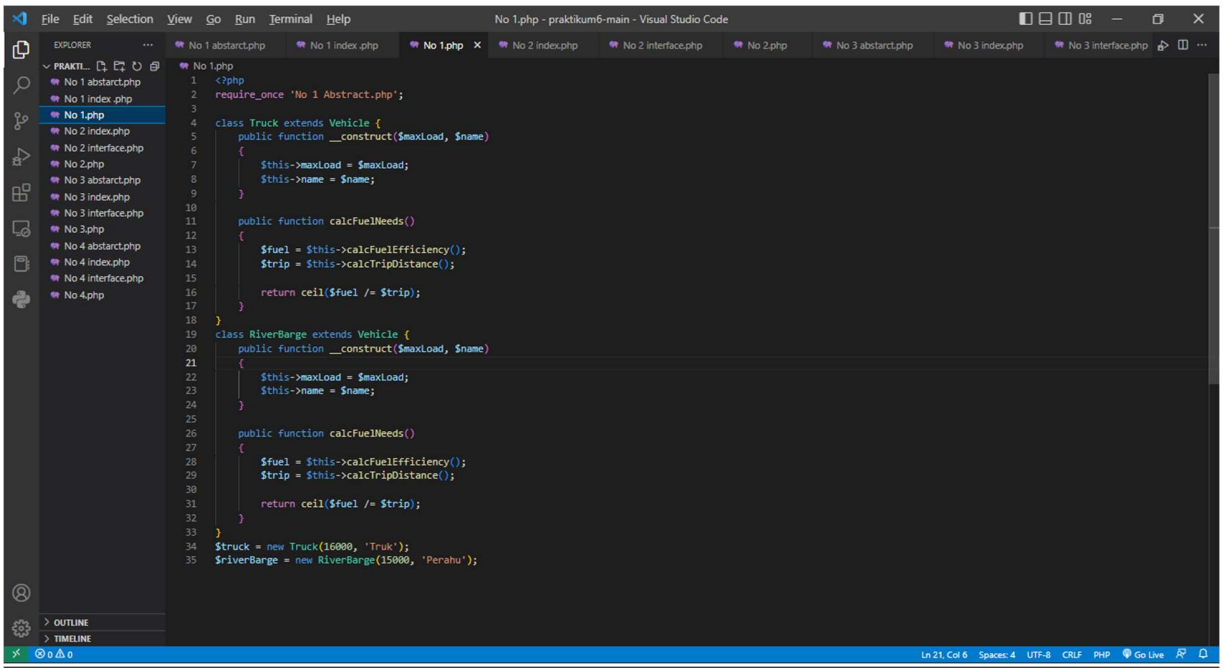
2021A

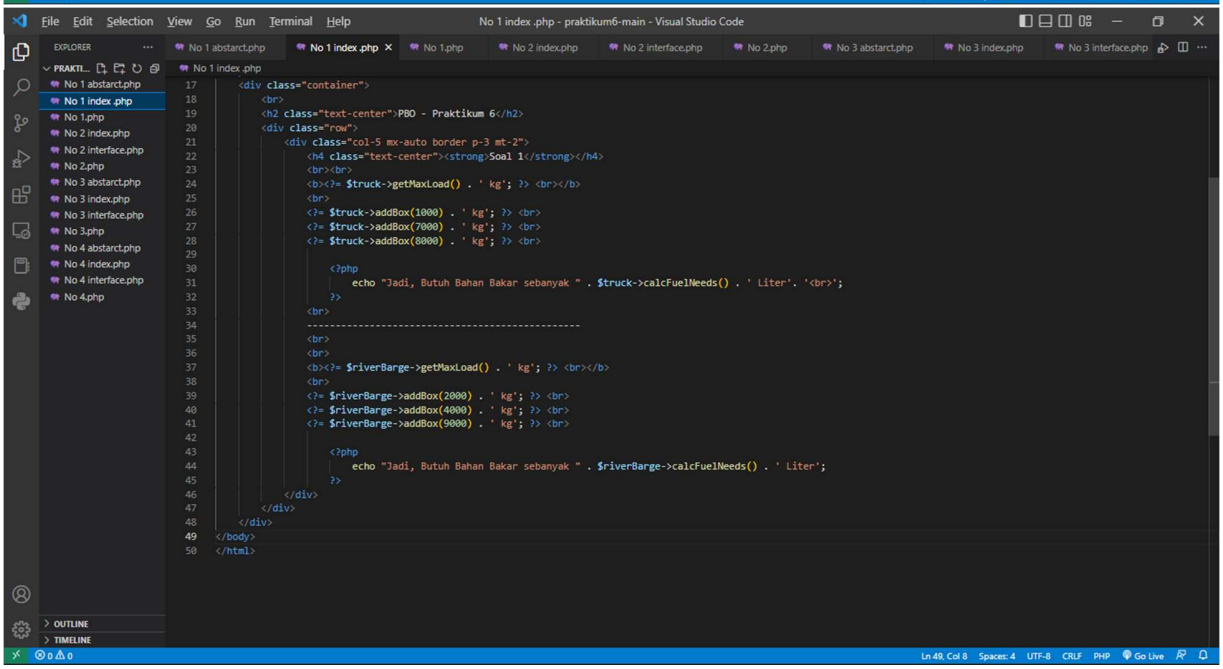
MANAJEMEN INFORMATIKA UNIVERSITAS NEGERI SURABAYA

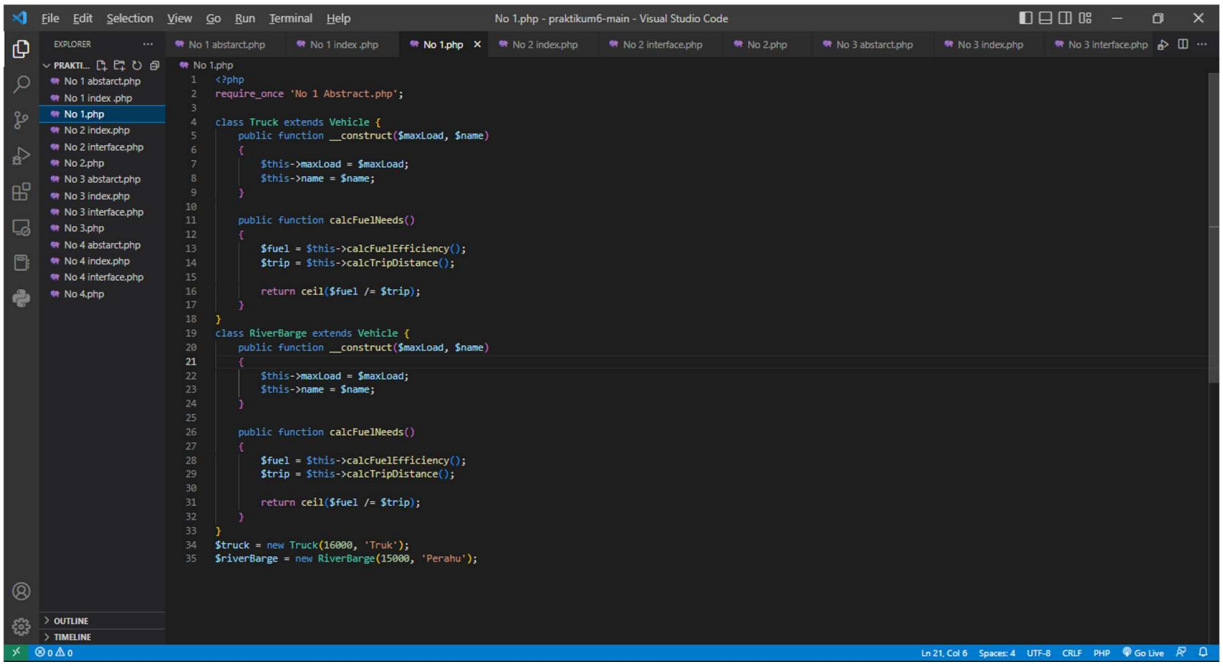
PROGRRAM VOKASI

2021

1. Source Code






File Edit Selection View Go Run Terminal Help
No 1.php - praktikum6-main - Visual Studio Code
EXPLORER
No 1 abstarct.php
No 1 index .php
No 1.php
No 2 index.php
No 2 interface.php
No 2.php
No 3 abstarct.php
No 3 index.php
No 3 interface.php
No 3.php
No 4 abstarct.php
No 4 index.php
No 4 interface.php
No 4.php
OUTLINE
TIMELINE
<?php
require_once 'No 1 Abstract.php';
class Truck extends Vehicle {
public function __construct($maxLoad, $name)
{
$this->maxLoad = $maxLoad;
$this->name = $name;
}
public function calcFuelNeeds()
{
$fuel = $this->calcFuelEfficiency();
$trip = $this->calcTripDistance();
return ceil($fuel /= $trip);
}
}
class RiverBarge extends Vehicle {
public function __construct($maxLoad, $name)
{
$this->maxLoad = $maxLoad;
$this->name = $name;
}
public function calcFuelNeeds()
{
$fuel = $this->calcFuelEfficiency();
$trip = $this->calcTripDistance();
return ceil($fuel /= $trip);
}
}
$truck = new Truck(16000, 'Truk');
$riverBarge = new RiverBarge(15000, 'Perahu');
Ln 21, Col 6 Spaces: 4 UTF-8 CRLF PHP Go Live

Ouput :

Soal No.1

Maksimal muatan Truk 18000 kg

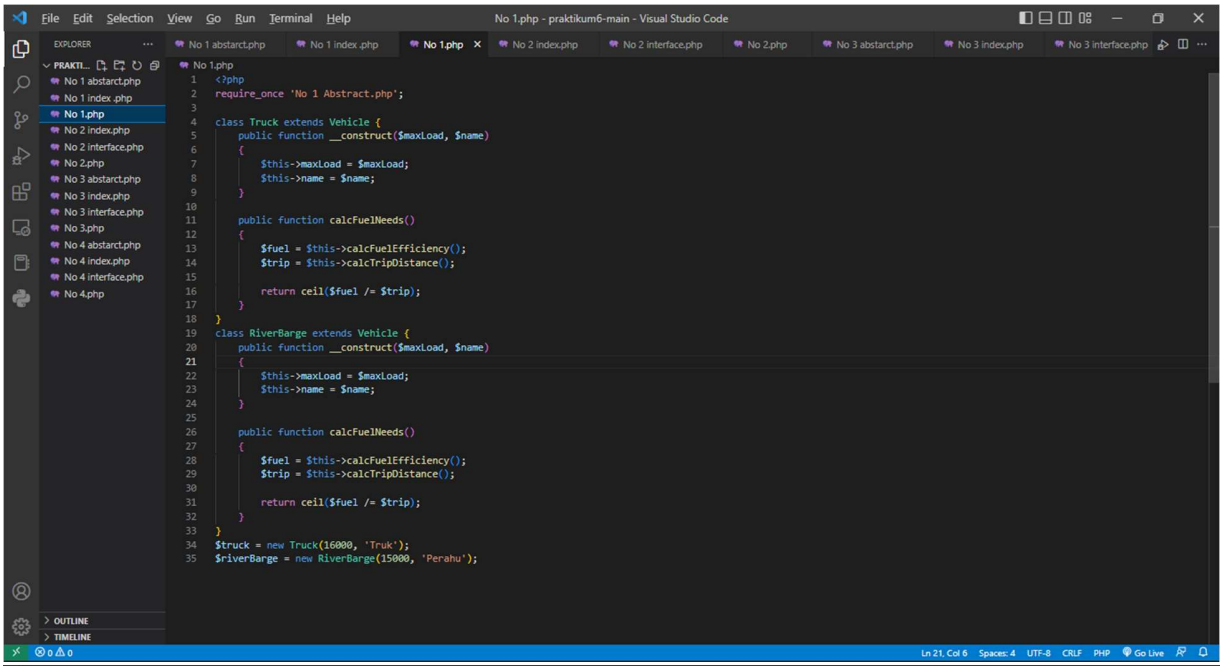
Truk menambah muatan sebesar 2000 kg
Truk menambah muatan sebesar 7000 kg
Truk menambah muatan sebesar 9000 kg
Jadi, Butuh Bahan Bakar sebanyak 6 Liter

Maksimal muatan Tongkang Sungai 20000 kg

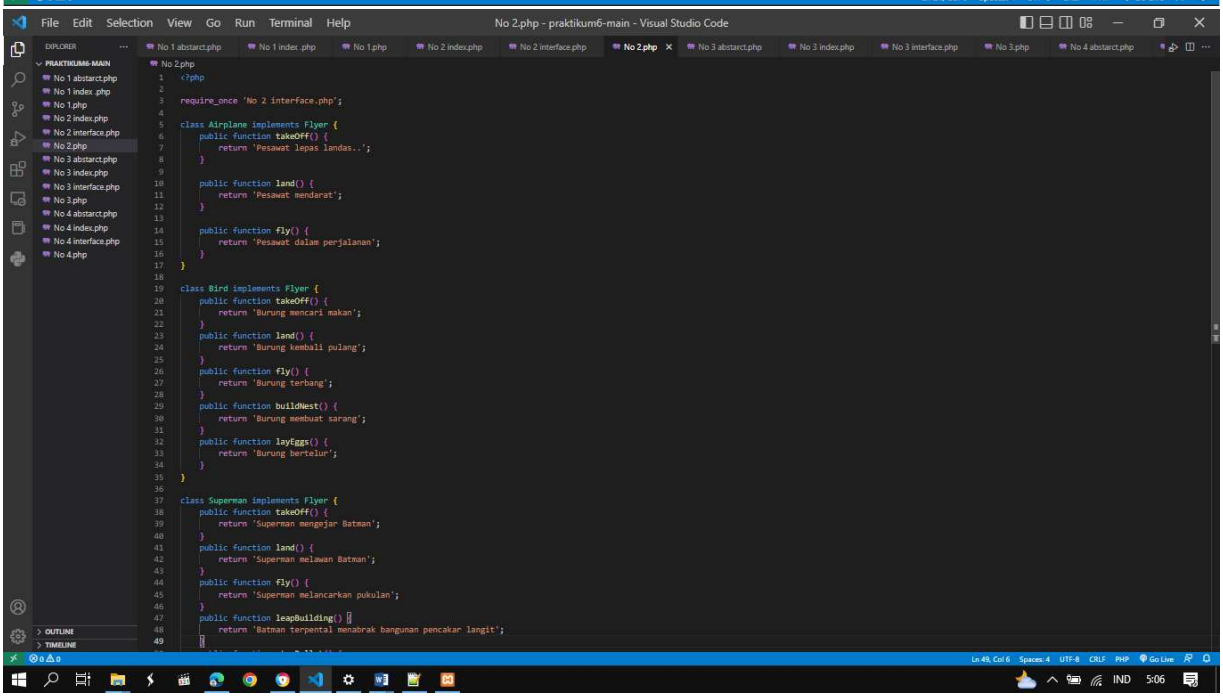
Tongkang Sungai menambah muatan sebesar 5000 kg
Tongkang Sungai menambah muatan sebesar 7000 kg
Tongkang Sungai menambah muatan sebesar 8000 kg
Jadi, Butuh Bahan Bakar sebanyak 5 Liter

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).

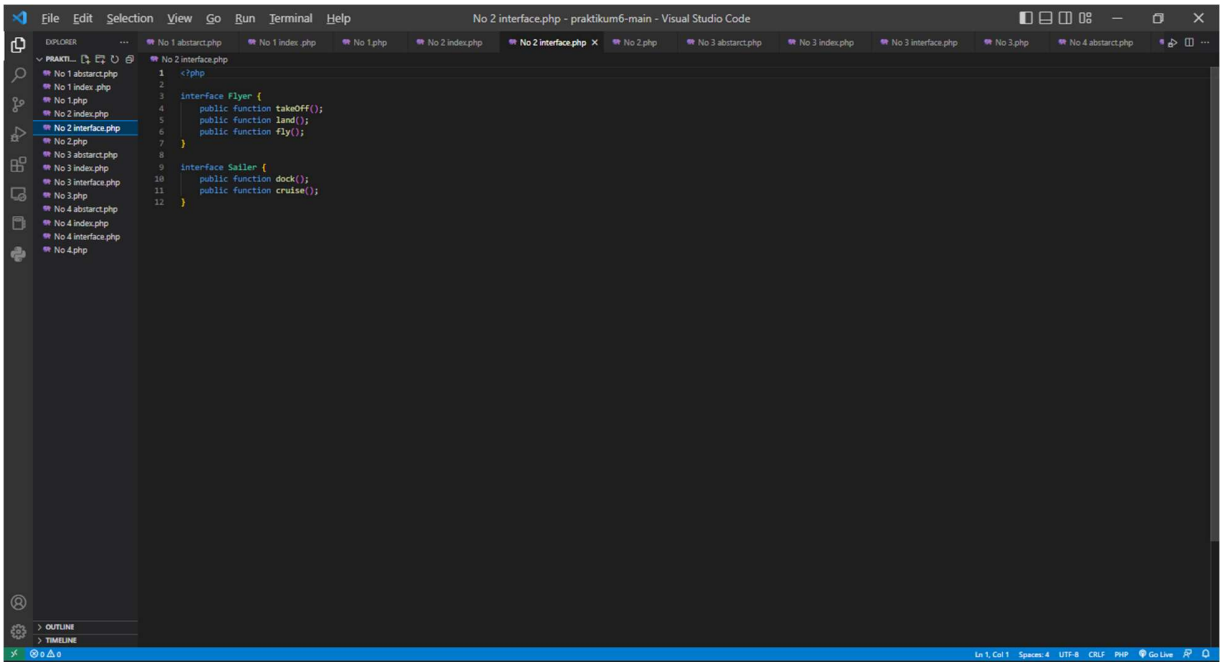
2. Source Code




```php
<?php
require_once 'No 1 Abstract.php';

class Truck extends Vehicle {
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
}
class RiverBarge extends Vehicle {
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
}
$truck = new Truck(16000, 'Truk');
$riverBarge = new RiverBarge(15000, 'Perahu');
```




```php
<?php

require_once 'No 2 interface.php';

class Airplane implements Flyer {
    public function takeOff() {
        return 'Pesawat lepas landas..';
    }

    public function land() {
        return 'Pesawat mendarat';
    }

    public function fly() {
        return 'Pesawat dalam perjalanan';
    }
}

class Bird implements Flyer {
    public function takeOff() {
        return 'Burung mencari makan';
    }
    public function land() {
        return 'Burung kembali pulang';
    }
    public function fly() {
        return 'Burung terbang';
    }
    public function buildNest() {
        return 'Burung membuat sarang';
    }
    public function layEggs() {
        return 'Burung bertelur';
    }
}

class Superman implements Flyer {
    public function takeOff() {
        return 'Superman mengejar Batman';
    }
    public function land() {
        return 'Superman melawan Batman';
    }
    public function fly() {
        return 'Superman melancarkan pukulan';
    }
    public function leapBuilding() {
        return 'Batman terpental menabrak bangunan pencakar langit';
    }
```



Output:

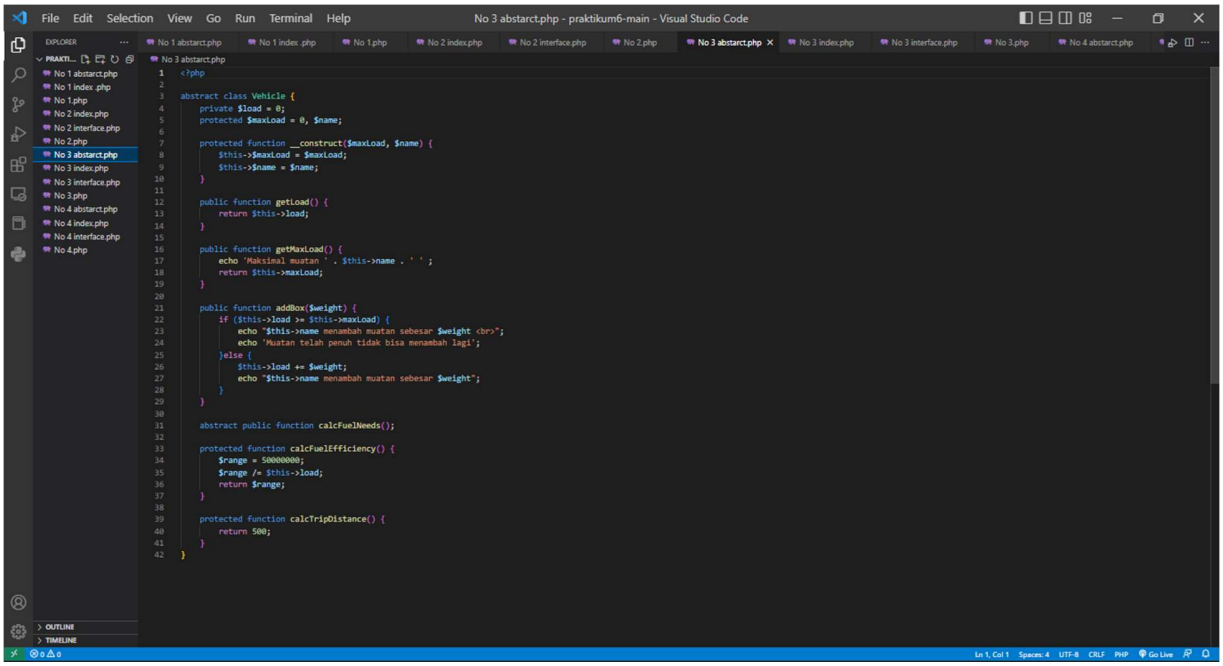
Superman
Superman melawan Batman
Superman mengejar Batman
Superman melancarkan pukulan
Batman terpental menabrak bangunan pencakar langit
Polisi menembaki superman namun ditangkis

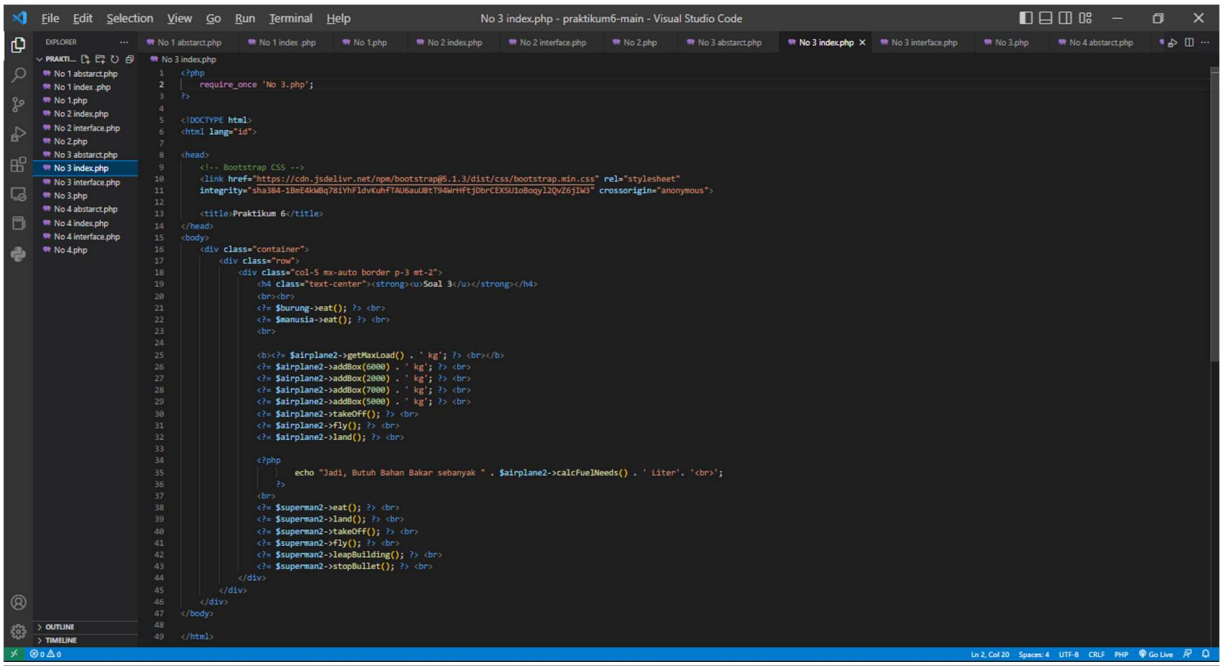
Bird
Burung membuat sarang
Burung mencari makan
Burung terbang
Burung kembali pulang
Burung bertelur

Airplane
Pesawat lepas landas..
Pesawat dalam perjalanan
Pesawat mendarat

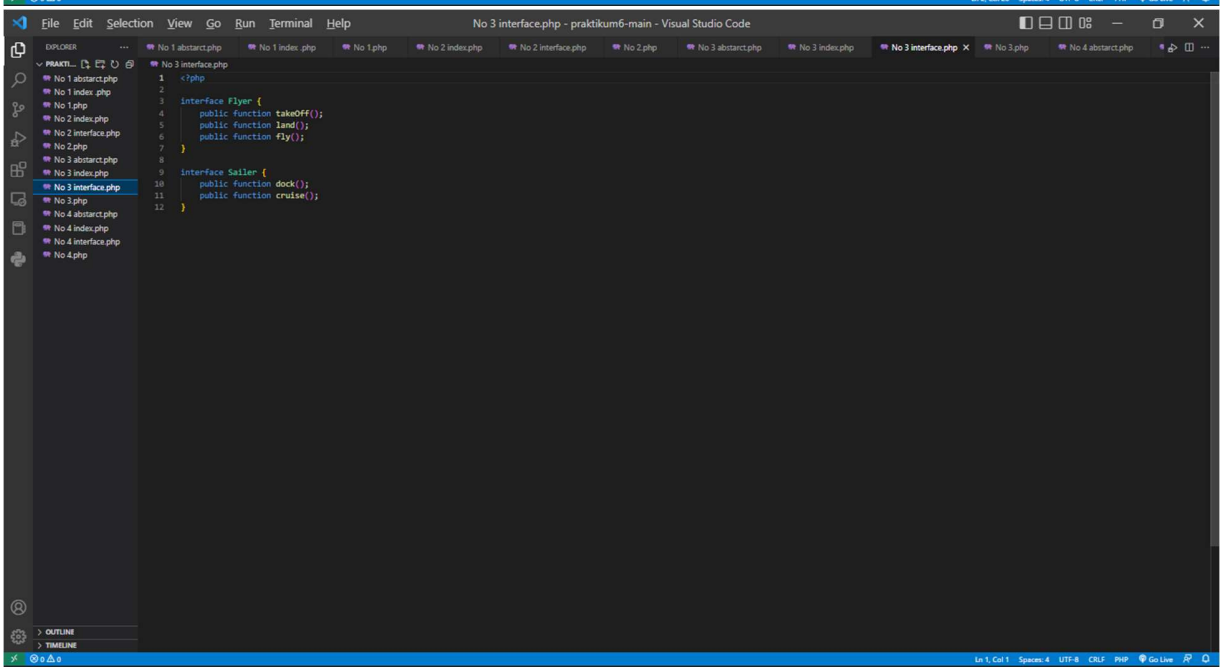
Analisa: Program ini merupakan implementasi dari Polymhorpism dengan menggunakan Interface Flyer. Oleh karena itu, setiap kelas yang mengimplementasikan antarmuka Flyer harus memiliki metode peluncuran, pendaratan, pendaratan, dan penerbangan.

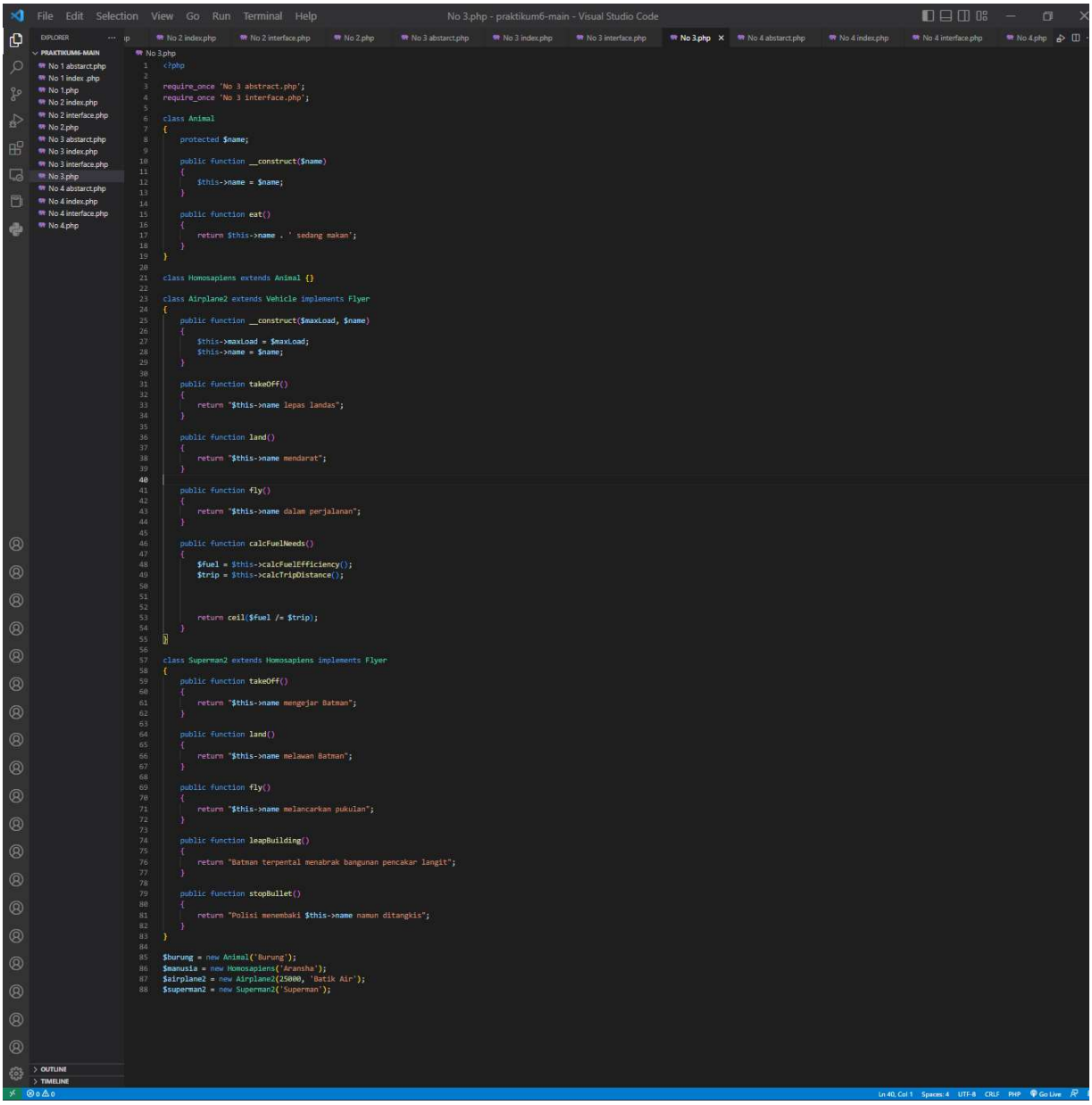
3. Source Code



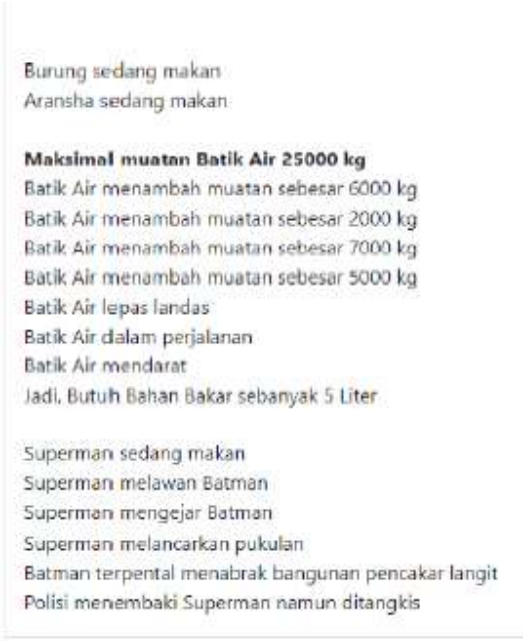

File Edit Selection View Go Run Terminal Help
No 3 index.php - praktikum6-main - Visual Studio Code
EXPLORER
PRAKTI...
No 1 abstarct.php
No 1 index .php
No 1.php
No 2 index.php
No 2 interface.php
No 2.php
No 3 abstarct.php
No 3 index.php
No 3 interface.php
No 3.php
No 4 abstarct.php
No 4 index.php
No 4 interface.php
No 4.php
OUTLINE
TIMELINE
No 3 index.php
1 <?php
2 require_once 'No 3.php';
3 ?>
4
5 <!DOCTYPE html>
6 <html lang="id">
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"
11 integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
12
13 <title>Praktikum 6</title>
14 </head>
15 <body>
16
17
18
19 <h4 class="text-center"><u>Soal 3</u></h4>
20

21 <?= $burung->eat(); ?>
22 <?= $manusia->eat(); ?>
23
24
25 <?= $airplane2->getMaxLoad() . ' kg'; ?>
26 <?= $airplane2->addBox(6000) . ' kg'; ?>
27 <?= $airplane2->addBox(2000) . ' kg'; ?>
28 <?= $airplane2->addBox(7000) . ' kg'; ?>
29 <?= $airplane2->addBox(5000) . ' kg'; ?>
30 <?= $airplane2->takeOff(); ?>
31 <?= $airplane2->fly(); ?>
32 <?= $airplane2->land(); ?>
33
34 <?php
35 echo "Jadi, Butuh Bahan Bakar sebanyak " . $airplane2->calcFuelNeeds() . ' Liter'. '
';
36 ?>
37
38 <?= $superman2->eat(); ?>
39 <?= $superman2->land(); ?>
40 <?= $superman2->takeOff(); ?>
41 <?= $superman2->fly(); ?>
42 <?= $superman2->leapBuilding(); ?>
43 <?= $superman2->stopBullet(); ?>
44
45
46
47 </body>
48
49 </html>
Ln 2, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live


File Edit Selection View Go Run Terminal Help
No 3 interface.php - praktikum6-main - Visual Studio Code
EXPLORER
PRAKTI...
No 1 abstarct.php
No 1 index .php
No 1.php
No 2 index.php
No 2 interface.php
No 2.php
No 3 abstarct.php
No 3 index.php
No 3 interface.php
No 3.php
No 4 abstarct.php
No 4 index.php
No 4 interface.php
No 4.php
OUTLINE
TIMELINE
No 3 interface.php
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 }
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live

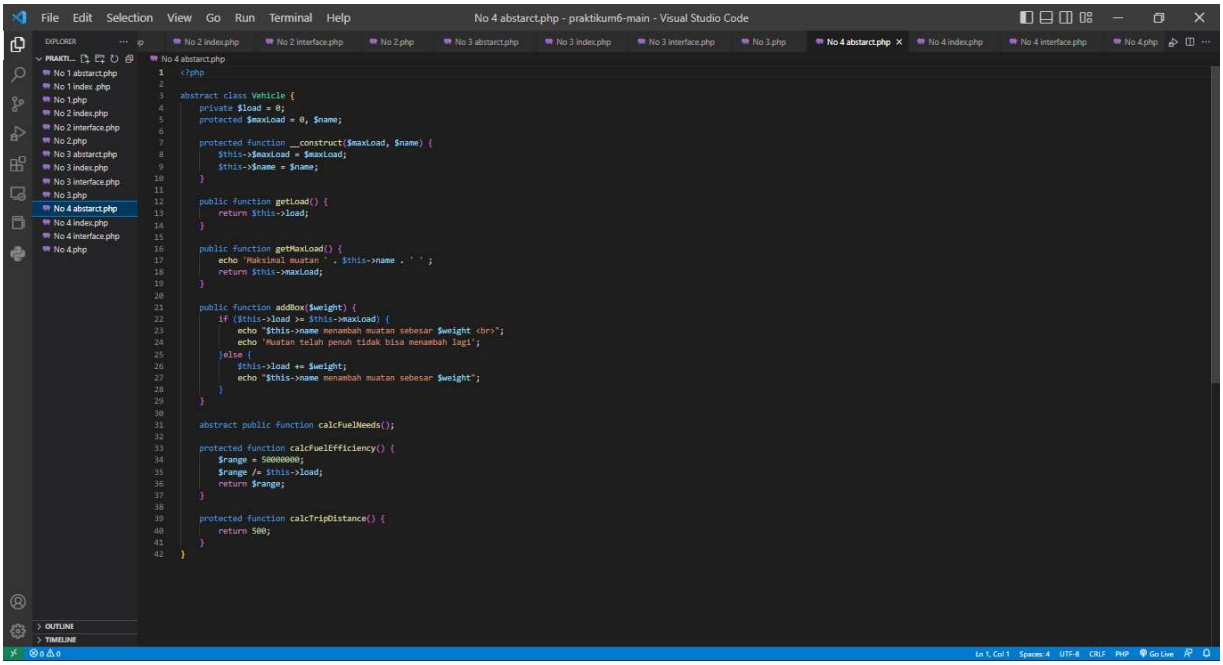

File Edit Selection View Go Run Terminal Help
No 3.php - praktikum6-main - Visual Studio Code
EXPLORER
PRAKTIKUM6-MAIN
No 1 abstarct.php
No 1 index .php
No 1.php
No 2 index.php
No 2 interface.php
No 2.php
No 3 abstarct.php
No 3 index.php
No 3 interface.php
No 3.php
No 4 abstarct.php
No 4 index.php
No 4 interface.php
No 4.php
OUTLINE
TIMELINE
No 2 index.php
No 2 interface.php
No 2.php
No 3 abstarct.php
No 3 index.php
No 3 interface.php
No 3.php
No 4 abstarct.php
No 4 index.php
No 4 interface.php
No 4.php
No 3.php
<?php
require_once 'No 3 abstract.php';
require_once 'No 3 interface.php';
class Animal
{
protected $name;
public function __construct($name)
{
$this->name = $name;
}
public function eat()
{
return $this->name . ' sedang makan';
}
}
class Homosapiens extends Animal {}
class Airplane2 extends Vehicle implements Flyer
{
public function __construct($maxLoad, $name)
{
$this->maxLoad = $maxLoad;
$this->name = $name;
}
public function takeOff()
{
return "$this->name lepas landas";
}
public function land()
{
return "$this->name mendarat";
}
public function fly()
{
return "$this->name dalam perjalanan";
}
public function calcFuelNeeds()
{
$fuel = $this->calcFuelEfficiency();
$trip = $this->calcTripDistance();
return ceil($fuel /= $trip);
}
}
class Superman2 extends Homosapiens implements Flyer
{
public function takeOff()
{
return "$this->name mengejar Batman";
}
public function land()
{
return "$this->name melawan Batman";
}
public function fly()
{
return "$this->name melancarkan pukulan";
}
public function leapBuilding()
{
return "Batman terpental menabrak bangunan pencakar langit";
}
public function stopBullet()
{
return "Polisi menembaki $this->name namun ditangkis";
}
}
$burung = new Animal('Burung');
$manusia = new Homosapiens('Aransha');
$airplane2 = new Airplane2(25000, 'Batik Air');
$superman2 = new Superman2('Superman');
Ln 40, Col 1
Spaces: 4
UTF-8
CRLF
PHP
Go Live

Ouput :

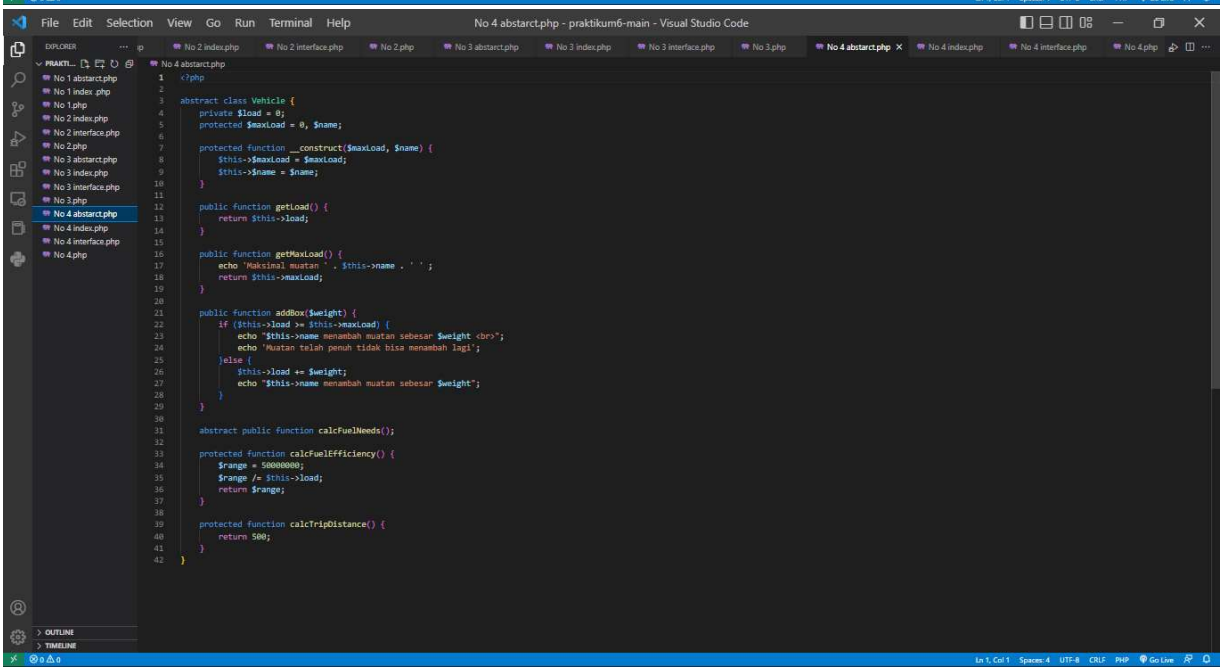


Analisa : Program ini memiliki antarmuka selebaran dan kelas kendaraan abstrak. Kelas Pesawat merupakan implementasi dari antarmuka Flyer dan turunan dari Kendaraan. Oleh karena itu, kelas Airplane harus memiliki metode untuk menghitung Kebutuhan Bahan Bakar, lepas landas, mendarat, dan terbang. Implementasi 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.

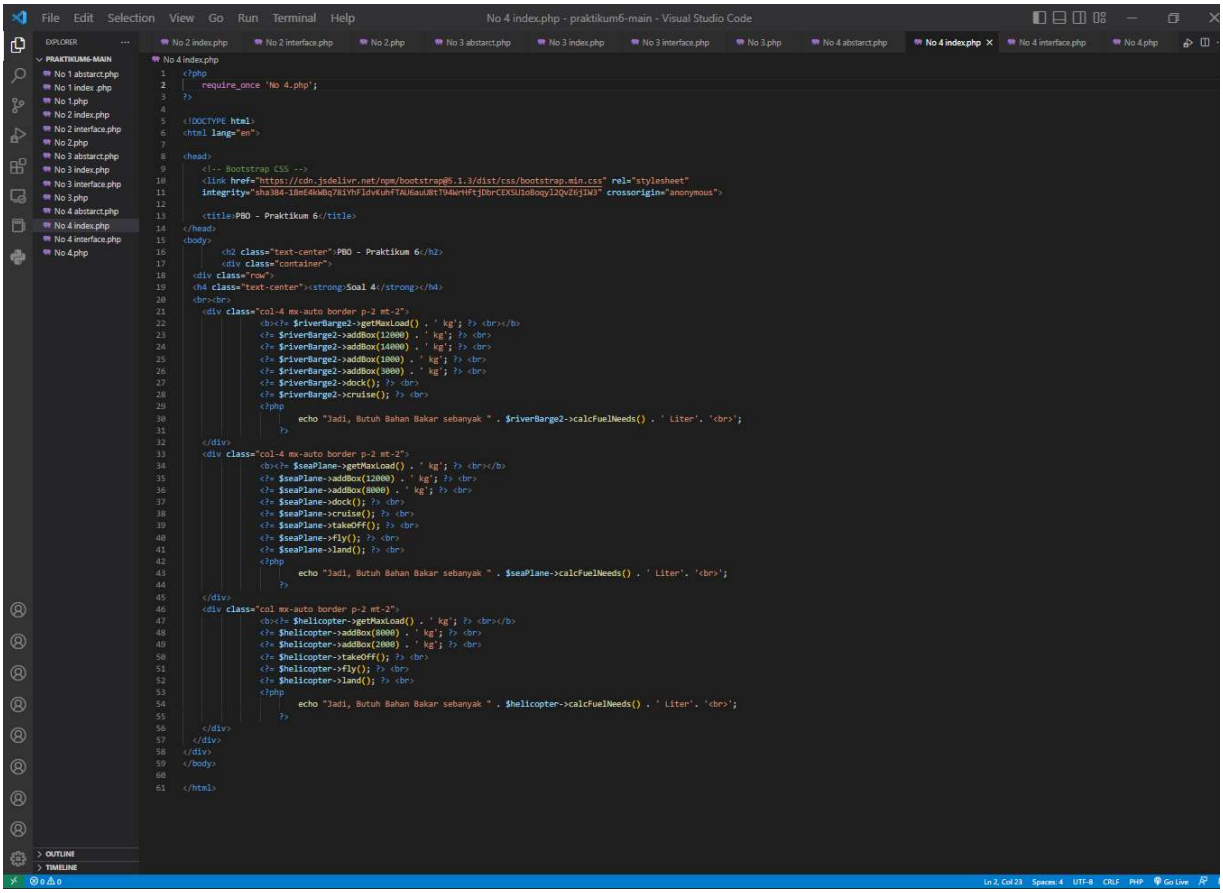
4. Source Code



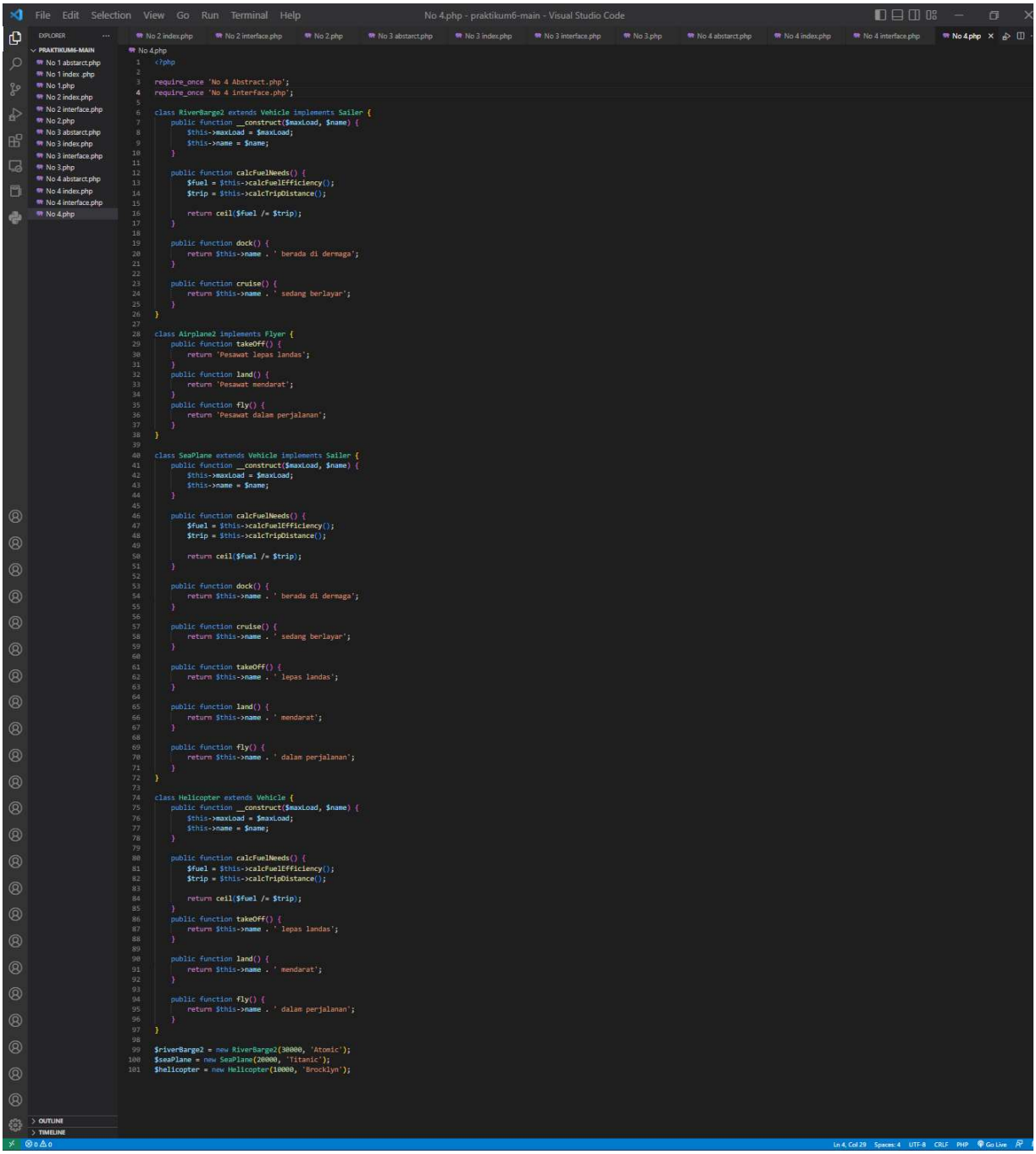

```php
<?php

abstract class Vehicle {
    private $load = 0;
    protected $maxLoad = 0, $name;

    protected function __construct($maxLoad, $name) {
        $this->$maxLoad = $maxLoad;
        $this->$name = $name;
    }

    public function getLoad() {
        return $this->load;
    }

    public function getMaxLoad() {
        echo 'Maksimal muatan ' . $this->name . ' ' ;
        return $this->maxLoad;
    }

    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name menambah muatan sebesar $weight <br>";
            echo 'Muatan telah penuh tidak bisa menambah lagi';
        }else {
            $this->load += $weight;
            echo "$this->name menambah muatan sebesar $weight";
        }
    }

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 50000000;
        $range /= $this->load;
        return $range;
    }

    protected function calcTripDistance() {
        return 500;
    }
}
```




```php
<?php

abstract class Vehicle {
    private $load = 0;
    protected $maxLoad = 0, $name;

    protected function __construct($maxLoad, $name) {
        $this->$maxLoad = $maxLoad;
        $this->$name = $name;
    }

    public function getLoad() {
        return $this->load;
    }

    public function getMaxLoad() {
        echo 'Maksimal muatan ' . $this->name . ' ' ;
        return $this->maxLoad;
    }

    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name menambah muatan sebesar $weight <br>";
            echo 'Muatan telah penuh tidak bisa menambah lagi';
        }else {
            $this->load += $weight;
            echo "$this->name menambah muatan sebesar $weight";
        }
    }

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 50000000;
        $range /= $this->load;
        return $range;
    }

    protected function calcTripDistance() {
        return 500;
    }
}
```




```php
<?php
    require_once 'No 4.php';
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <title>PBO - Praktikum 6</title>
</head>
<body>
        <h2 class="text-center">PBO - Praktikum 6</h2>
        <div class="container">
  <div class="row">
  <h4 class="text-center"><strong>Soal 4</strong></h4>
  <br><br>
    <div class="col-4 mx-auto border p-2 mt-2">
                <b><?= $riverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $riverBarge2->addBox(12000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(14000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(1000) . ' kg'; ?> <br>
                <?= $riverBarge2->addBox(3000) . ' kg'; ?> <br>
                <?= $riverBarge2->dock(); ?> <br>
                <?= $riverBarge2->cruise(); ?> <br>
                <?php
                    echo "Jadi, Butuh Bahan Bakar sebanyak " . $riverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
                ?>
    </div>
    <div class="col-4 mx-auto border p-2 mt-2">
                <b><?= $seaPlane->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $seaPlane->addBox(12000) . ' kg'; ?> <br>
                <?= $seaPlane->addBox(8000) . ' kg'; ?> <br>
                <?= $seaPlane->dock(); ?> <br>
                <?= $seaPlane->cruise(); ?> <br>
                <?= $seaPlane->takeOff(); ?> <br>
                <?= $seaPlane->fly(); ?> <br>
                <?= $seaPlane->land(); ?> <br>
                <?php
                    echo "Jadi, Butuh Bahan Bakar sebanyak " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
                ?>
    </div>
    <div class="col mx-auto border p-2 mt-2">
                <b><?= $helicopter->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $helicopter->addBox(8000) . ' kg'; ?> <br>
                <?= $helicopter->addBox(2000) . ' kg'; ?> <br>
                <?= $helicopter->takeOff(); ?> <br>
                <?= $helicopter->fly(); ?> <br>
                <?= $helicopter->land(); ?> <br>
                <?php
                    echo "Jadi, Butuh Bahan Bakar sebanyak " . $helicopter->calcFuelNeeds() . ' Liter'. '<br>';
                ?>
    </div>
  </div>
</div>
</body>

</html>
```



```php
<?php

require_once 'No 4 Abstract.php';
require_once 'No 4 interface.php';

class RiverBarge2 extends Vehicle implements Sailer {
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }

    public function dock() {
        return $this->name . ' berada di dermaga';
    }

    public function cruise() {
        return $this->name . ' sedang berlayar';
    }
}

class Airplane2 implements Flyer {
    public function takeOff() {
        return 'Pesawat lepas landas';
    }
    public function land() {
        return 'Pesawat mendarat';
    }
    public function fly() {
        return 'Pesawat dalam perjalanan';
    }
}

class SeaPlane extends Vehicle implements Sailer {
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }

    public function dock() {
        return $this->name . ' berada di dermaga';
    }

    public function cruise() {
        return $this->name . ' sedang berlayar';
    }

    public function takeOff() {
        return $this->name . ' lepas landas';
    }

    public function land() {
        return $this->name . ' mendarat';
    }

    public function fly() {
        return $this->name . ' dalam perjalanan';
    }
}

class Helicopter extends Vehicle {
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
    public function takeOff() {
        return $this->name . ' lepas landas';
    }

    public function land() {
        return $this->name . ' mendarat';
    }

    public function fly() {
        return $this->name . ' dalam perjalanan';
    }
}

$riverBarge2 = new RiverBarge2(30000, 'Atomic');
$seaPlane = new SeaPlane(20000, 'Titanic');
$helicopter = new Helicopter(10000, 'Brocklyn');
```

Ouput :

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