

**LAPORAN PRAKTIKUM 3 INHERITANCE**  
**Pemrograman Berorientasi Objek**

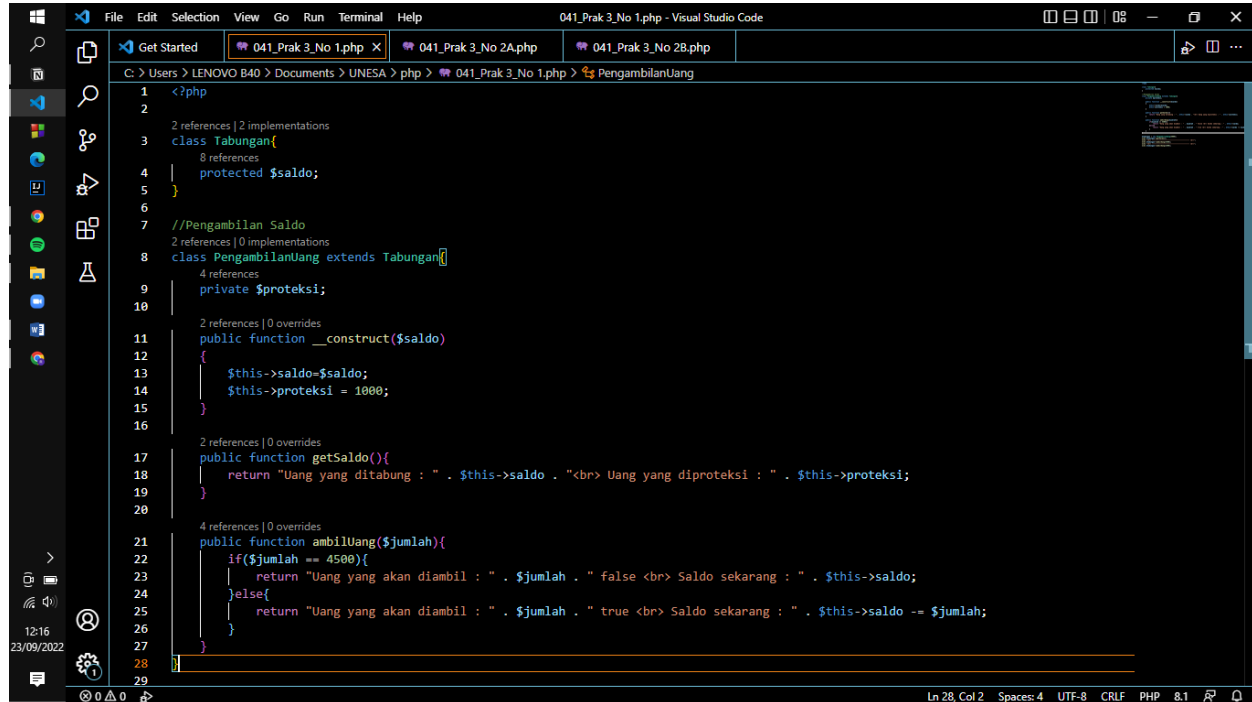


**Disusun Oleh:**  
**Refila Dyah Ghizanda Wardoyo (21091397041)**

**D4 Manajemen Informatika**  
**Fakultas Vokasi**  
**Universitas Negeri Surabaya**

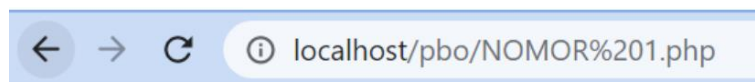
1. Transformasikan diagram UML class diagram dibawah ini kedalam program File PHP

- Codes



```
1 <?php
2
3 class Tabungan{
4     protected $saldo;
5 }
6
7 //Pengambilan Saldo
8 class PengambilanUang extends Tabungan{
9     private $proteksi;
10
11     public function __construct($saldo)
12     {
13         $this->saldo=$saldo;
14         $this->proteksi = 1000;
15     }
16
17     public function getSaldo(){
18         return "Uang yang ditabung : " . $this->saldo . "<br> Uang yang diproteksi : " . $this->proteksi;
19     }
20
21     public function ambilUang($jumlah){
22         if($jumlah == 4500){
23             return "Uang yang akan diambil : " . $jumlah . " false <br> Saldo sekarang : " . $this->saldo;
24         }else{
25             return "Uang yang akan diambil : " . $jumlah . " true <br> Saldo sekarang : " . $this->saldo -- $jumlah;
26         }
27     }
28
29 }
```

- Output



Uang yang ditabung : 5000  
Uang yang diproteksi : 1000

-----  
Uang yang akan diambil : 4500 false  
Saldo sekarang : 5000

-----  
Uang yang akan diambil : 2500 true  
Saldo sekarang : 2500

2. a) Buatlah kelas-kelas berdasarkan UML class diagram dibawah ini

```
1 <?php
2
3 class Kendaraan {
4     protected $jmlRoda, $warna;
5     6 references | 0 overrides
6     function setJmlRoda($jmlRoda) {
7         $this->jmlRoda = $jmlRoda;
8     }
9     6 references | 0 overrides
10    public function getJmlRoda() {
11        return $this->jmlRoda;
12    }
13    6 references | 0 overrides
14    public function setWarna($warna) {
15        $this->warna = $warna;
16    }
17    6 references | 0 overrides
18    public function getWarna() {
19        return $this->warna;
20    }
21 }
22
23 class Mobil extends Kendaraan {
24     protected $bahanBakar, $kapasitasMesin;
25     4 references | 0 overrides
26    public function setBahanBakar($bahanBakar) {
27        $this->bahanBakar = $bahanBakar;
28    }
29    4 references | 0 overrides
30    public function getBahanBakar() {
31        return $this->bahanBakar;
32    }
33    4 references | 0 overrides
34    public function setKapasitasMesin($kapasitasMesin) {
35
36    }
37 }
38
39 $this->kapasitasMesin = $kapasitasMesin;
40
41 public function getKapasitasMesin() {
42     return $this->kapasitasMesin;
43 }
44
45 class Sepeda extends Kendaraan {
46     protected $jmlSadel, $jmlGir;
47     2 references | 0 overrides
48    public function setJmlSadel($jmlSadel) {
49        $this->jmlSadel = $jmlSadel;
50    }
51    2 references | 0 overrides
52    public function getJmlSadel() {
53        return $this->jmlSadel;
54    }
55    2 references | 0 overrides
56    public function setJmlGir($jmlGir) {
57        $this->jmlGir = $jmlGir;
58    }
59    2 references | 0 overrides
60    public function getJmlGir() {
61        return $this->jmlGir;
62    }
63 }
64
65 class Truk extends Mobil {
66     private $muatanMaks;
67     2 references | 0 overrides
68    public function setMuatanMaks($muatanMaks) {
69        $this->muatanMaks = $muatanMaks;
70    }
71 }
```

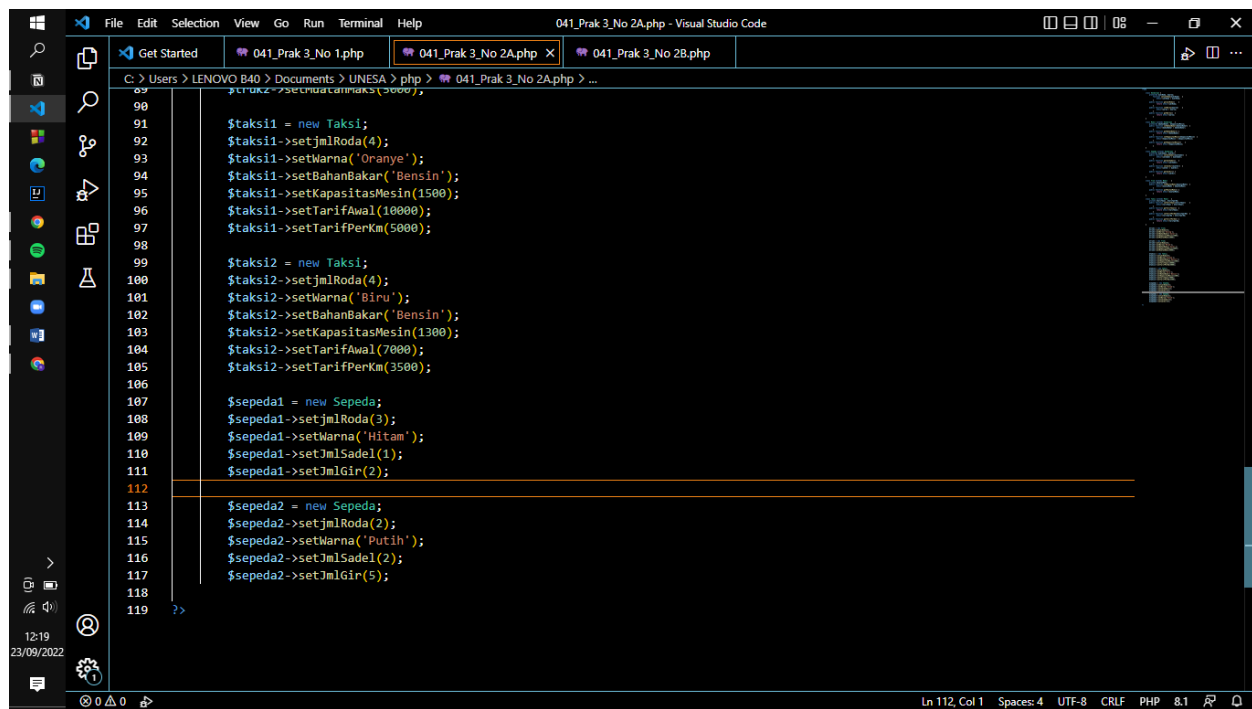
Visual Studio Code interface showing two PHP files: 041\_Prak 3\_No 2A.php and 041\_Prak 3\_No 2B.php.

**041\_Prak 3\_No 2A.php**

```
53 public function setMuatanMaks($muatanMaks) {
54     $this->muatanMaks = $muatanMaks;
55 }
56 public function getMuatanMaks() {
57     return $this->muatanMaks;
58 }
59
60
61 2 references | 0 overrides
62 class Taksi extends Mobil {
63     private $tarifAwal, $tarifPerKm;
64     2 references | 0 overrides
65     public function setTarifAwal($tarifAwal) {
66         $this->tarifAwal = $tarifAwal;
67     }
68     2 references | 0 overrides
69     public function getTarifAwal() {
70         return $this->tarifAwal;
71     }
72     2 references | 0 overrides
73     public function setTarifPerKm($tarifPerKm) {
74         $this->tarifPerKm = $tarifPerKm;
75     }
76     2 references | 0 overrides
77     public function getTarifPerKm() {
78         return $this->tarifPerKm;
79     }
80 }
81 $truk1 = new Truk;
82 $truk1->setJmlRoda(4);
83 $truk1->setWarna('Kuning');
84 $truk1->setBahanBakar('Solar');
85 $truk1->setKapasitasMesin(1500);
86 $truk1->setMuatanMaks(1000);
```

**041\_Prak 3\_No 2B.php**

```
79 $truk1->setWarna('Kuning');
80 $truk1->setBahanBakar('Solar');
81 $truk1->setKapasitasMesin(1500);
82 $truk1->setMuatanMaks(1000);
83
84 $truk2 = new Truk;
85 $truk2->setJmlRoda(6);
86 $truk2->setWarna('Merah');
87 $truk2->setBahanBakar('Solar');
88 $truk2->setKapasitasMesin(2000);
89 $truk2->setMuatanMaks(5000);
90
91 $taksi1 = new Taksi;
92 $taksi1->setJmlRoda(4);
93 $taksi1->setWarna('Oranye');
94 $taksi1->setBahanBakar('Bensin');
95 $taksi1->setKapasitasMesin(1500);
96 $taksi1->setTarifAwal(10000);
97 $taksi1->setTarifPerKm(5000);
98
99 $taksi2 = new Taksi;
100 $taksi2->setJmlRoda(4);
101 $taksi2->setWarna('Biru');
102 $taksi2->setBahanBakar('Bensin');
103 $taksi2->setKapasitasMesin(1300);
104 $taksi2->setTarifAwal(7000);
105 $taksi2->setTarifPerKm(3500);
106
107 $sepeda1 = new Sepeda;
108 $sepeda1->setJmlRoda(3);
109 $sepeda1->setWarna('Hitam');
110 $sepeda1->setJmlSadel(1);
111 $sepeda1->setJmlGir(2);
112
113 $sepeda2 = new Sepeda;
```



b) Selanjutnya buatlah kelas Tes.php yang membuat obyek-obyek serta mengeset nilai variabel seperti pada Tabel 1 dan tampilkan data per obyek

```
1 <?php
2 | require_once 'Soal No 2 A.php';
3 | ?>
4
5 <!DOCTYPE html>
6 <html lang="en">
7 <head>
8 | <title>Praktikum Modul 3</title>
9 </head>
10 <body>
11 <table>
12 |
13 | <tr>
14 | | <td>No</td>
15 | | <td>Object</td>
16 | | <td>Jumlah Roda</td>
17 | | <td>Warna</td>
18 | | <td>Bahan Bakar</td>
19 | | <td>Kapasitas Mesin</td>
20 | | <td>Muatan Maksimal</td>
21 | | <td>Tarif Awal</td>
22 | | <td>Tarif Per Km</td>
23 | | <td>Jumlah Sadel</td>
24 | | <td>Jumlah Gir</td>
25 | </tr>
26 | <tr>
27 | | <td>1</td>
28 | | <td>Truk 1</td>
29 | | <td><?= $truk1->getJmlRoda(); ?></td>
30 | | <td><?= $truk1->getWarna(); ?></td>
31 | | <td><?= $truk1->getBahanBakar(); ?></td>
32 | | <td><?= $truk1->getKapasitasMesin(); ?></td>
33 | | <td><?= $truk1->getMuatanMaks(); ?></td>
34 | | <td></td>
35 | | <td></td>
36 | </tr>
37 | <tr>
38 | | <td>2</td>
39 | | <td>Truk 2</td>
40 | | <td><?= $truk2->getJmlRoda(); ?></td>
41 | | <td><?= $truk2->getWarna(); ?></td>
42 | | <td><?= $truk2->getBahanBakar(); ?></td>
43 | | <td><?= $truk2->getKapasitasMesin(); ?></td>
44 | | <td><?= $truk2->getMuatanMaks(); ?></td>
45 | | <td></td>
46 | | <td></td>
47 | </tr>
48 | <tr>
49 | | <td>3</td>
50 | | <td>Taksi 1</td>
51 | | <td><?= $taksi1->getJmlRoda(); ?></td>
52 | | <td><?= $taksi1->getWarna(); ?></td>
53 | | <td><?= $taksi1->getBahanBakar(); ?></td>
54 | | <td><?= $taksi1->getKapasitasMesin(); ?></td>
55 | | <td><?= $taksi1->getTarifAwal(); ?></td>
56 | | <td><?= $taksi1->getTarifPerKm(); ?></td>
57 | | <td></td>
58 | | <td></td>
59 | </tr>
60 | <tr>
61 | | <td>4</td>
62 | | <td>Taksi 2</td>
63 | | <td><?= $taksi2->getJmlRoda(); ?></td>
64 | | <td><?= $taksi2->getWarna(); ?></td>
65 | | <td><?= $taksi2->getBahanBakar(); ?></td>
66 | | <td><?= $taksi2->getKapasitasMesin(); ?></td>
67 | | <td></td>
68 | | <td></td>
69 | </tr>
70 | </table>
```

Visual Studio Code interface showing two screenshots of the file `041_Prak 3_No 2B.php`.

The top screenshot shows the code from line 68 to 102. The code is an HTML table with two rows of data. The first row contains information for 'Sepeda 1' and the second row for 'Sepeda 2'. The table has columns for 'Warna', 'Roda', 'Kapasitas', 'Tarif Awal', and 'Tarif Per Km'. The code uses PHP functions like `$staksi2->getWarna()`, `$staksi2->getBahanBakar()`, `$staksi2->getKapasitasMesin()`, `$staksi2->getTarifAwal()`, `$staksi2->getTarifPerKm()`, `$sepeda1->getJmlRoda()`, `$sepeda1->getWarna()`, `$sepeda1->getJmlSadel()`, and `$sepeda1->getJmlGir()` to populate the table.


The bottom screenshot shows the code from line 103 to 105. The code continues the HTML table structure, closing the table and the body tags.

```
68 <td><?=$staksi2->getWarna(); ?></td>
69 <td><?=$staksi2->getBahanBakar(); ?></td>
70 <td><?=$staksi2->getKapasitasMesin(); ?></td>
71 <td><?=$staksi2->getTarifAwal(); ?></td>
72 <td><?=$staksi2->getTarifPerKm(); ?></td>
73 <td><?=$staksi2->getTarifPerKm(); ?></td>
74 <td><?=$staksi2->getTarifPerKm(); ?></td>
75 <td><?=$staksi2->getTarifPerKm(); ?></td>
76 </tr>
77 <tr>
78 <td>5</td>
79 <td>Sepeda 1</td>
80 <td><?=$sepeda1->getJmlRoda(); ?></td>
81 <td><?=$sepeda1->getWarna(); ?></td>
82 <td><?=$sepeda1->getWarna(); ?></td>
83 <td><?=$sepeda1->getWarna(); ?></td>
84 <td><?=$sepeda1->getWarna(); ?></td>
85 <td><?=$sepeda1->getWarna(); ?></td>
86 <td><?=$sepeda1->getWarna(); ?></td>
87 <td><?=$sepeda1->getJmlSadel(); ?></td>
88 <td><?=$sepeda1->getJmlGir(); ?></td>
89 </tr>
90 <tr>
91 <td>6</td>
92 <td>Sepeda 2</td>
93 <td><?=$sepeda2->getJmlRoda(); ?></td>
94 <td><?=$sepeda2->getWarna(); ?></td>
95 <td><?=$sepeda2->getWarna(); ?></td>
96 <td><?=$sepeda2->getWarna(); ?></td>
97 <td><?=$sepeda2->getWarna(); ?></td>
98 <td><?=$sepeda2->getWarna(); ?></td>
99 <td><?=$sepeda2->getWarna(); ?></td>
100 <td><?=$sepeda2->getJmlSadel(); ?></td>
101 <td><?=$sepeda2->getJmlGir(); ?></td>
102 </tr>
```

The bottom screenshot shows the code from line 103 to 105. The code continues the HTML table structure, closing the table and the body tags.

```
103 </table>
104 </body>
105 </html>
```

- Output



No	Object	Jumlah Roda	Warna	Bahan Bakar	Kapasitas Mesin	Muatan Maksimal	Tarif Awal	Tarif Per KM	Jumlah Sadel	Jumlah Gir
1	Truk 1	4	Kuning	Solar	1500	1000	-	-	-	-
2	Truk 2	6	Merah	Solar	2000	5000	-	-	-	-
3	Taksi 1	4	Oranye	Bensin	1500	-	10000	5000	-	-
4	Taksi 2	4	Biru	Bensin	1300	-	7000	3500	-	-
5	Sepeda 1	3	Hitam	-	-	-	-	-	1	2
6	Sepeda 2	2	Putih	-	-	-	-	-	2	5

### 3. Analisa Laporan

#### Analisa nomor 1

Dalam kasus tersebut, saya memanfaatkan aturan overriding. Saldo yang diinisialisasi di awal akan dimasukkan ke method `setSaldo` yang ada di parent class. Uang proteksi disini bersifat sementara dan akan hilang ketika tidak digunakan lagi. Oleh karena itu, ketika `ambilUang(2500)` dijalankan, maka saldo sekarangny adalah  $5000 - 2500 = 2500$  bukan 1500. Hal ini bisa terjadi karena method `getSaldo()` yang ada di subclass akan meng-override method `getSaldo()` yang ada di parent class.

#### Analisa nomor 2

Program pada soal nomer 2 merupakan contoh program yang menerapkan konsep inheritance, overriding method dan keyword `super`. Pertama ada class `Kendaraan` yaitu class yang sebagai induk. Class `Kendaraan` mempunyai variable roda dan warna. Kemudian anak dari class `Kendaraan` tersebut ada dua yaitu class `Mobil` dan class `Sepeda`. Jadi kedua anak class tersebut mempunyai variable roda dan warna yang diturunkan dari class `Kendaraan` tadi. Penulisan class seperti biasa menggunakan `extends`. Syntax-nya sebagai berikut `public class Mobil extends Kendaraan` dan `public class Sepeda extends Kendaraan`. Class `Mobil` memiliki variable bahan bakar dan kapasitas mesin dan Class `Sepeda` mempunyai variable sadel dan gir. Kemudian class mobil menurunkan sifatnya kepada class `Truk` dan `Taksi`. Jadi kedua class ini mempunyai data variable dari mobil dan kendaraan. Penulisan class seperti biasa menggunakan `extends`. Syntax-nya sebagai berikut `public class Truk extends Mobil` dan `public class Taksi extends Mobil`. Class `Truk` mempunyai variable muatan maksimal dan class `Taksi` mempunyai variable tarif awal dan tarif per kilometer. Class `Sepeda` tidak menurunkan sifatnya karena kendaraan dalam UML yang lain tidak