

Latihan praktek oop mvc

Nama: Alden Christian Totonafu Waruwu

NIM : G.211.21.0087

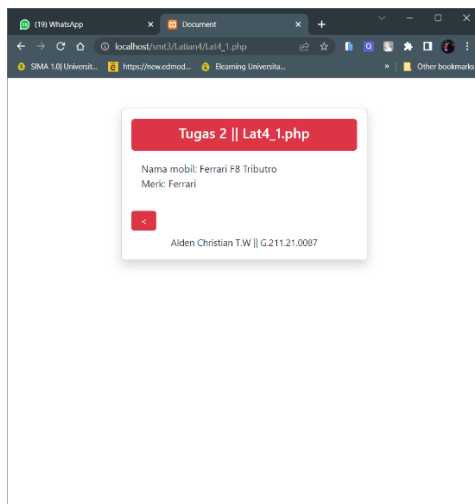
1. Lat4_1.PHP

A. Setelah Menggunakan Metode overload getinfo 5a

```
1 <?php
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <title>Document</title>
9 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css">
10 <script defer src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>
11 </head>
12
13 <body>
14 <?php
15 //class Mobil
16 class Mobil
17 {
18     public $nama;
19     public $merk;
20     function getinfo($a)
21     {
22         echo "Nama mobil : " . $a->nama . "<br/>";
23         echo "Merk : " . $a->merk . "<br/>";
24     }
25 }
26
27
28 <div class="container mt-5 bg-gray-100">
29 <div class="card p-3 bg-3 shadow-sm auto" style="width: 480px;">
30 <div class="card-header text-center text-bg-danger border-1 rounded">
31 <h3>Tugas 2 || Lat4_1.php </h3>
32 </div>
33 <div class="card-body">
34 <?php
35 $a = new Mobil();
36 $a->nama = "Ferrari F8 Tributo";
37 $a->merk = "Ferrari";
38
39 $a->getinfo($a);
40
41 </div>
42 </div>
43 </div>
44 </body>
45 </html>
```

```
1 <?php
2
3 $a = new Mobil();
4 $a->nama = "Ferrari F8 Tributo";
5 $a->merk = "Ferrari";
6
7 $a->getinfo($a);
8
9 </div>
10 </div>
11 </div>
12 </div>
13 </div>
14 </div>
15 </div>
16 </div>
17 </div>
18 </div>
19 </div>
20 </div>
21 </div>
22 </div>
23 </div>
24 </div>
25 </div>
26 </div>
27 </div>
28 </div>
29 </div>
30 </div>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
```

B. Tampilan



C. Kesimpulan

1. Cara membuat class pada PHP <?php

```

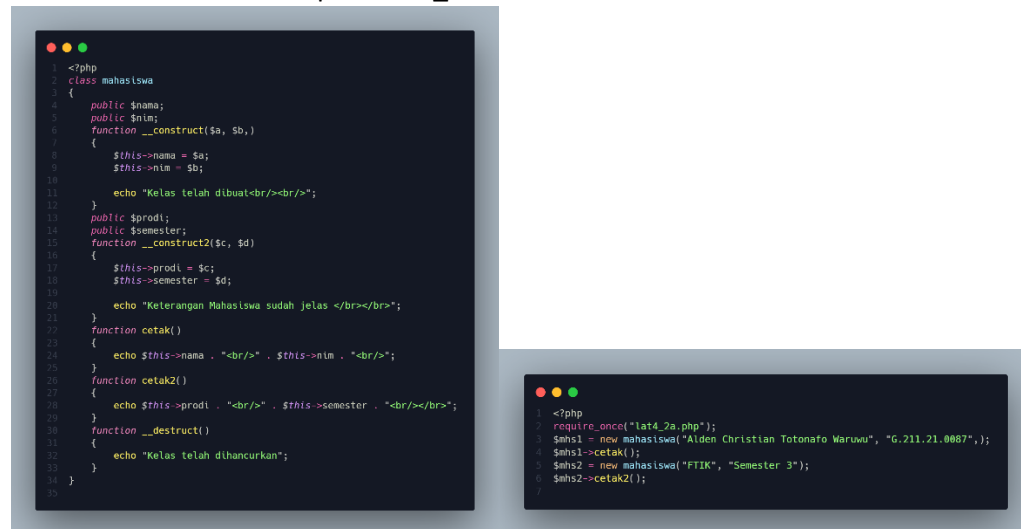
Class nama_class {
}
?>

```

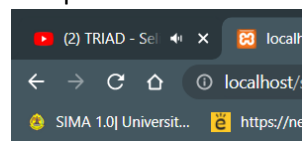
2. Cara penulisan property
Modifier \$nama_properti;
3. Penulisan method
Modifier function nama_method(){
Isi_method;}
4. Cara inisiasi object
\$nama_object = new nama_class();
5. Cara mengisi property atau mendefinisikan property
\$nama_object->properties="aaa";
6. Cara memanggil/menjalankan method pada suatu Class
\$nama_object->nama_methode();

2. Lat4_2a dan Lat4_2B

- A. Menambahkan construct pada Lat4_2a



- B. Tampilan



Kelas telah dibuat

Alden Christian
G.211.21.0087
FTIK
Semester 3

Kelas telah dihancurkan

- C. Kesimpulan : Constructor dalam OOP PHP tidak dapat di Override

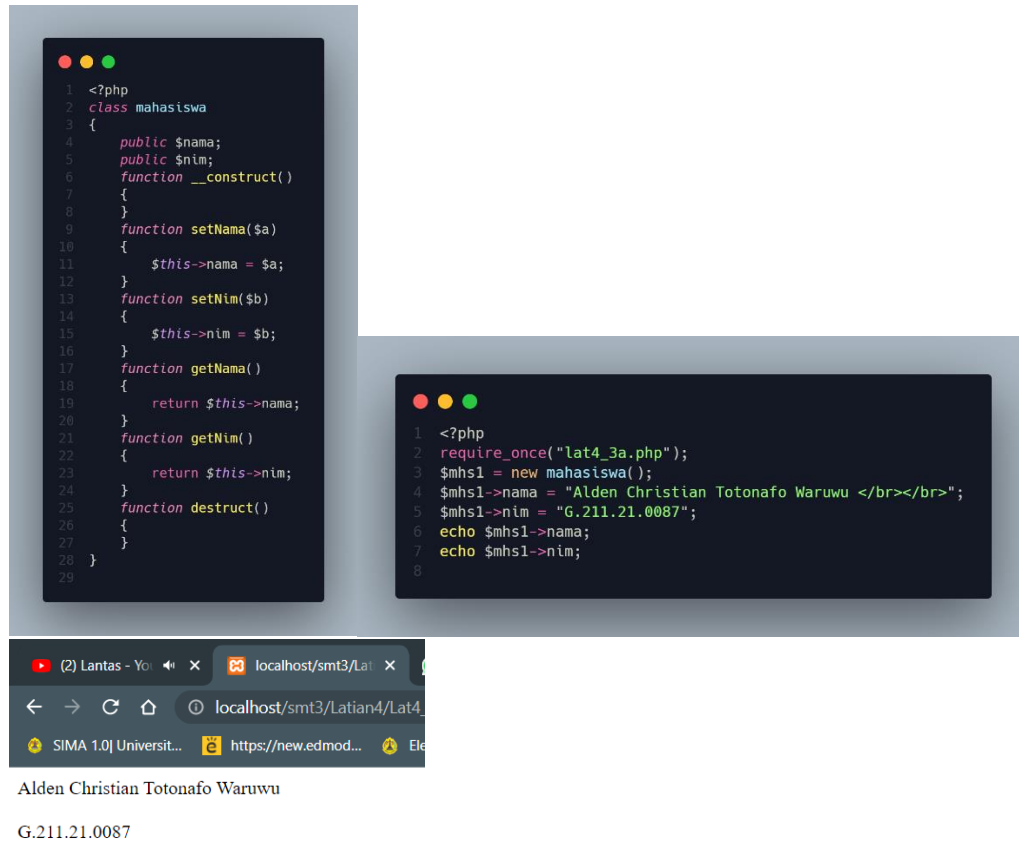
3. Lat4_3a dan Lat4_3b

- A. Program tersebut mengalami beberapa error, Hal itu terjadi dikarenakan Property yang bermodifikasi private hanya bisa digunakan pada class mahasiswa sendiri.
- B. Modifier protected dan Public

b.1 Protected

b.2 Public

Tampilan:



The image shows a PHP class definition for 'mahasiswa' and its usage in a script. The class has two public properties, '\$nama' and '\$nim', and several methods: '__construct()', 'setNama()', 'setNim()', 'getNama()', 'getNim()', and 'destruct()'. The script 'lat4_3a.php' creates an instance of the class, sets its name to 'Alden Christian Tonafo Waruwu' and its NIM to 'G.211.21.0087', and then echoes these values. The browser output shows the name and NIM on separate lines.

```
1 <?php
2 class mahasiswa
3 {
4     public $nama;
5     public $nim;
6     function __construct()
7     {
8     }
9     function setNama($a)
10    {
11        $this->nama = $a;
12    }
13    function setNim($b)
14    {
15        $this->nim = $b;
16    }
17    function getNama()
18    {
19        return $this->nama;
20    }
21    function getNim()
22    {
23        return $this->nim;
24    }
25    function destruct()
26    {
27    }
28 }
29
```

```
1 <?php
2 require_once("lat4_3a.php");
3 $mhs1 = new mahasiswa();
4 $mhs1->nama = "Alden Christian Tonafo Waruwu </br></br>";
5 $mhs1->nim = "G.211.21.0087";
6 echo $mhs1->nama;
7 echo $mhs1->nim;
8
```

Alden Christian Tonafo Waruwu
G.211.21.0087

C. Kesimpulan:

I. Penggunaan Modifier

Modifier keterangan public

Untuk mendefinisikan data atau metode yang akan terlihat dari luar oleh siapapun dan dimanapun.

A. Private

Untuk Mendefinisikan data atau metode yang akan terlihat pada class/object itu sendiri.

B. Protected

Untuk mendefinisikan data atau metode untuk tidak terlihat dari luar (seperti private), tetapi akan dapat diakses oleh "anak" dari class tersebut.

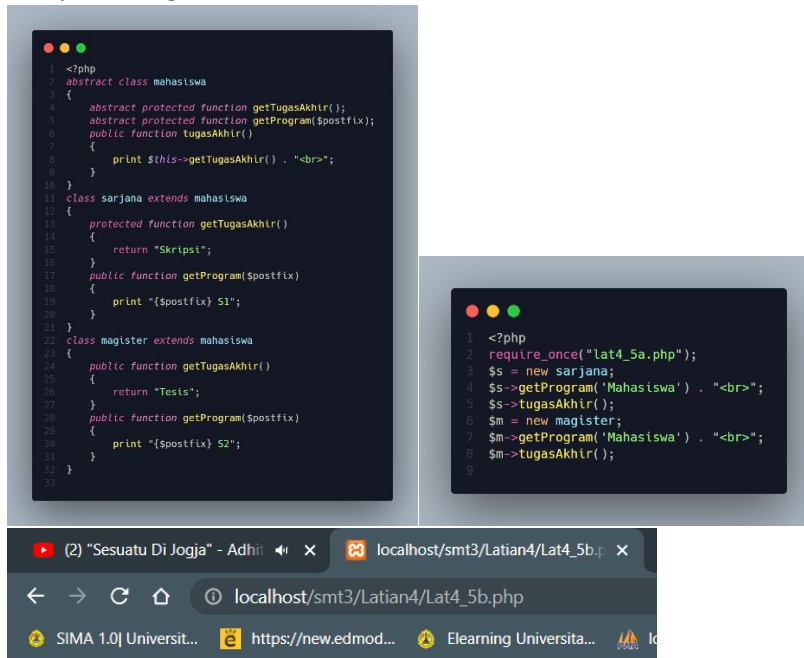
- II. Modifier Protected dan Private, Properties bisa dipanggil dengan mengimplementasikan setter-getter.

4. Lat4_4a dan Lat4_4b

- A. Tampilan
- B. Kesimpulan: Pada PHP OOP, Class asisten (child) bisa memanggil method dari mahasiswa (Parent).

5. Lat4_5a dan Lat_5b

- A. Tampilan Program



```
1 <?php
2 abstract class mahasiswa
3 {
4     abstract protected function getTugasAkhir();
5     abstract protected function getProgram($postfix);
6     public function tugasAkhir()
7     {
8         print $this->getTugasAkhir() . "<br>";
9     }
10 }
11 class sarjana extends mahasiswa
12 {
13     protected function getTugasAkhir()
14     {
15         return "Skripsi";
16     }
17     public function getProgram($postfix)
18     {
19         print "{$postfix} S1";
20     }
21 }
22 class magister extends mahasiswa
23 {
24     public function getTugasAkhir()
25     {
26         return "Tesis";
27     }
28     public function getProgram($postfix)
29     {
30         print "{$postfix} S2";
31     }
32 }
33
```

```
1 <?php
2 require_once("Lat4_5a.php");
3 $s = new sarjana;
4 $s->getProgram('Mahasiswa') . "<br>";
5 $s->tugasAkhir();
6 $m = new magister;
7 $m->getProgram('Mahasiswa') . "<br>";
8 $m->tugasAkhir();
9
```

Mahasiswa S1Skripsi
Mahasiswa S2Tesis

- B. Kesimpulan
- class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.

6. Lat4_6

- A. Maksud dari program diatas adalah penggunaan object interface
- B. Tampilan

```
Lat4_6.php > bar
1 <?php
2 interface a
3 {
4     public function foo();
5 }
6 interface b
7 {
8     public function bar();
9 }
10 interface c extends a, b
11 {
12     public function baz();
13 }
14 class d implements c
15 {
16     public function foo()
17     {
18     }
19     public function bar()
20     {
21     }
22 }
23
```

Fatal error: Class d contains 1 abstract method and must therefore be declared abstract or implement the remaining methods (c::baz) in C:\xampp\htdocs\TUGAS 2 FRAMEWORK OOP MVC\Lat4_6.php on line 14

Error, karena pada object interface, ketika kita mengimplementasikan object tersebut, seluruh method pada interface harus diimplementasikan seluruhnya. Karena class d mengimplementasikan interface c, maka method-method pada interface c harus diimplementasikan seluruhnya.

C. Setelah menambahkan class baru yaitu class e

```
Lat4_6.php > foo
1 <?php
2 interface a
3 {
4     public function foo();
5 }
6 interface b
7 {
8     public function bar();
9 }
10 interface c extends a, b
11 {
12     public function baz();
13 }
14 class d implements c
15 {
16     public function foo()
17     {
18     }
19     public function bar()
20     {
21     }
22     public function baz()
23     {
24     }
25 }
26 class e
27 {
28     public function foo()
29     {
30     }
31     public function bar()
32     {
33     }
34     public function baz()
35     {
36     }
37 }
```

Error, method foo dan bar terdapat pada interface a dan b, pada penerapannya sebuah class baru tidak dapat mengimplementasikan method yang sama pada dua interface.

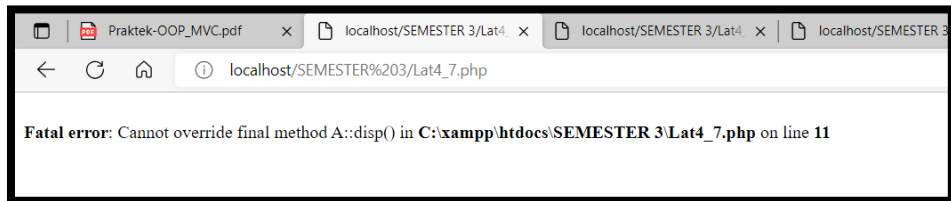
D. Kesimpulan

- 1) Interface didefinisikan dengan "Interface" keyword, mirip dengan deklarasi class biasa, hanya saja definisi atau detail method tidak dituliskan.
- 2) Seluruh method yang dideklarasikan pada interface harus memiliki modifier " public ".
- 3) Untuk mengimplementasikan sebuah interface, kita dapat menggunakan "implement" keyword.

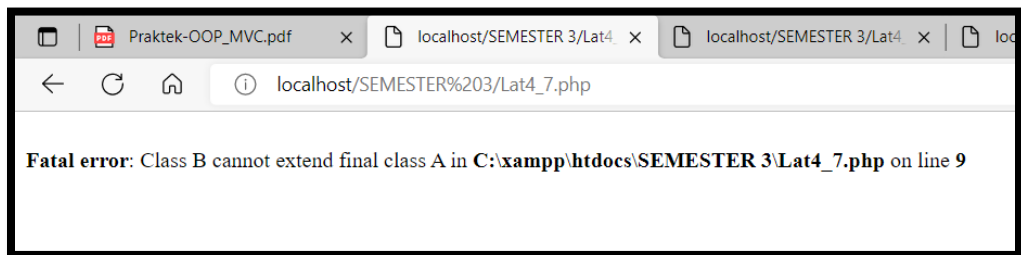
- 4) Seluruh method yang ada pada interface harus diimplementasikan seluruhnya. Sebuah class bisa mengimplementasikan lebih dari satu interface.
- 5) Class tidak bisa mengimplementasikan dua interface yang mempunyai nama method yang sama.
- 6) Interface bisa diwariskan seperti class menggunakan “extends”.
- 7) Class yang mengimplementasikan interface harus menggunakan method-method yang ada pada interface tersebut dengan nama dan spesifikasi yang sama persis.

7. Lat4_7

- a. Tampilan program (program tersebut mengalami error)



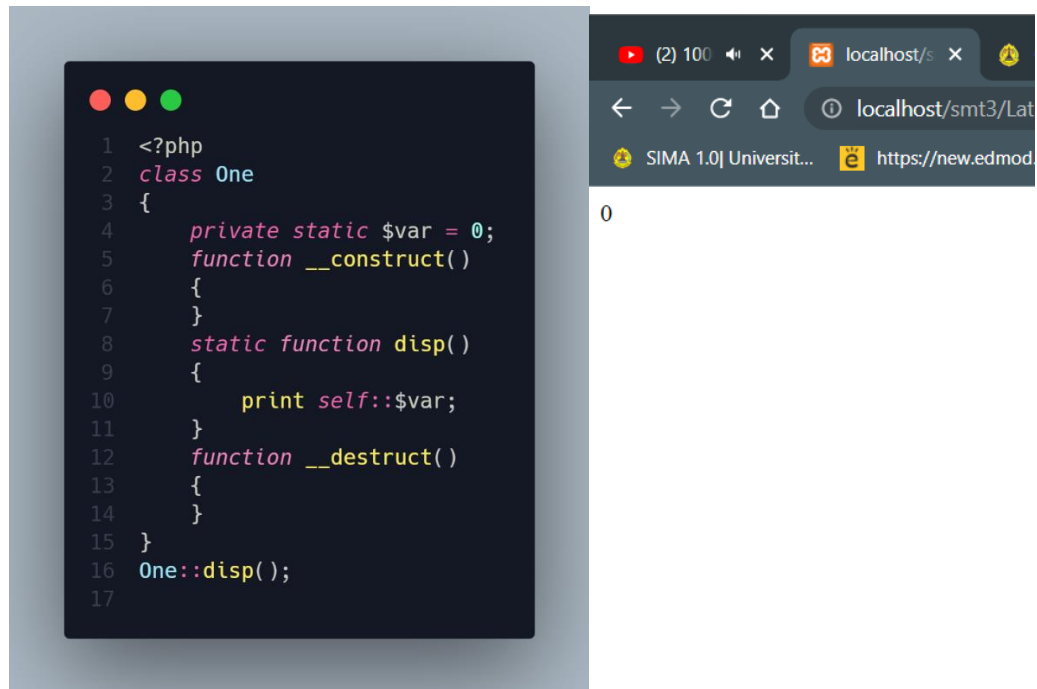
- b. Setelah memindahkan kata final dari baris 5 ke baris 2 (program tersebut tetap mengalami error), error pada program tersebut dikarenakan...



- c. Kesimpulan
1. final, akan mencegah proses overriding method pada class anak (sub-class)
 2. Apabila metode kita berikan status final, maka metode tersebut tidak akan bisa dioverride, begitu juga pada class, apabila kita berikan status “final” pada deklarasi class maka class tersebut tidak bisa diperpanjang (diwariskan).

8. Lat4_8

- A. Tampilan



B. Kesimpulan

1.Lat4_8.php mengimplementasikan penggunaan property static dengan modifierprivate.Properti statis dideklarasikan dengan menggunakan katakunci statis sebelum modifier- Sintaks:

modifier static \$nama_property = nilai;

2.Sifat statis dapat diakses tanpa perlu sebuah contoh objek dari kelas, menggunakannama kelas bersama dengan ::- Sintaks:

ClassName :: \$nama_property method_name();

Perhatikan bahwa properti statis menggunakan tanda dollar (\$).Properti statis tidak dapat diakses melalui obyek menggunakan operator **panah "->"**

9. Latian_9

a. Database

perpustakaan	id_buku	judul	pengarang	penerbit	tahun_terbit	isbn	jumlah_buku	tgl_input	file
New	13	Laut Berencana	Leila S. Chudori	Gramedia	1997		30	2017-12-10	03102022193447
anggota	14	Antares	Rweinda	Loveable	2020		30	2021-01-01	03102022193646
buku	15	Dikta dan Hukum	Dhian Farah	Loveable	2021		30	2022-05-01	03102022193840
transaksi	16	Dragon Ball	Akira Toriyama	Elex media komputindo	1995		30	2004-02-10	13122019052904
user	17	Sword Art Online	Kawahara Reki	A-1 Pictures	2009		30	2019-12-01	13122019073332
phpmyadmin									
test									

b. Model

```
1 <?php
2
3 class Koneksi {
4     var $host = 'localhost';
5     var $user = 'root';
6     var $pass = '';
7     var $db_name = 'perpustakaan'; // nama database
8
9     public function __construct() {
10         $this->konek = mysqli_connect($this->host, $this->user, $this->pass, $this->db_name);
11     }
12     //
13 }
14 ?>
```

c. Controller

```
1 <?php
2
3 require_once '../config/perpustakaan.php';
4
5 class Buku extends perpustakaan {
6     public function getDataPetugas($id) {
7         $stmt = mysqli_query($this->konek, "SELECT * FROM tb_petugas WHERE id_petugas = '" . $id . "'");
8
9         return $stmt;
10     }
11 }
```

d. View

e. Tampilan

The screenshot displays the Miracle Library web application interface. The browser's address bar shows the URL `localhost/perpustakaan/index_admin.php?page=buku`. The application has a dark sidebar on the left with the 'Miracle Library' logo and three menu items: 'Data Buku', 'Data Peminjam', and 'Transaksi Peminjaman'. The main content area features a 'Tambah Buku' button and a table titled 'Table Buku'. The table lists 5 books with columns for No, Judul, Pengarang, Penerbit, ISBN, Jumlah Buku, and Aksi. Each book entry has 'Ubah' (Edit) and 'Hapus' (Delete) buttons. Below the table, it indicates 'Showing 1 to 5 of 5 entries' with pagination controls for 'Previous', '1', and 'Next'.

Miracle Library

Logout

MIRACLE

Data Buku

Data Peminjam

Transaksi Peminjaman

Tambah Buku

Table Buku

10 records per page

Search:

No	Judul	Pengarang	Penerbit	ISBN	Jumlah Buku	Aksi
1	Laut Bercerita	Leila S. Chudori	Gramedia		30	Ubah Hapus
2	Antares	Rweinda	Loveable		30	Ubah Hapus
3	Dikta dan Hukum	Dhian Farah	Loveable		30	Ubah Hapus
4	Dragon Ball	Akira Toriyama	Elex media komputindo		30	Ubah Hapus
5	Sword Art Online	Kawahara Reki	A-1 Pictures		30	Ubah Hapus

Showing 1 to 5 of 5 entries

Previous 1 Next