

PEMOGRAMAN FRAMEWORK



CODEIGNITER 3 STUDI KASUS SI PERPUSTAKAAN

(Semua Script Di Tes Dengan Menggunakan PHP 7.2)

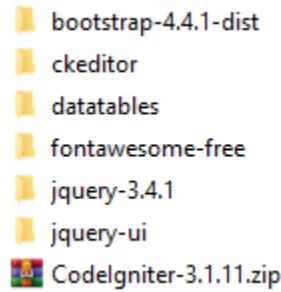
TAMUS TAHIR

TAMUS TAHIR

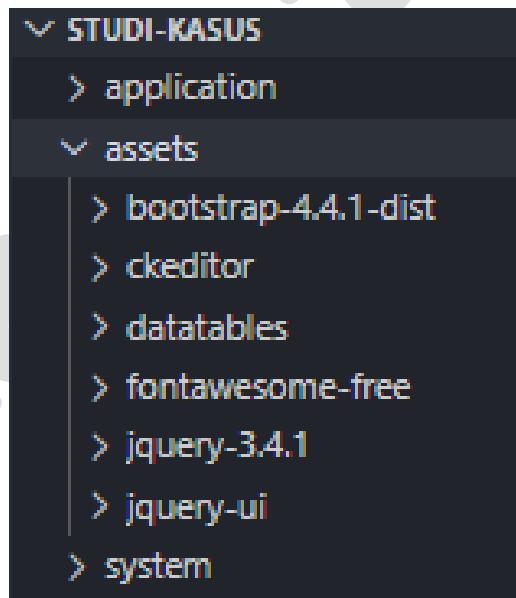
2020

INSTALASI

1. Pada folder bahan kalian akan melihat beberapa file yang akan kita pakai pada tutorial kali ini



2. Extract codeigniter → ubah nama foldernya menjadi studi-kasus → copy ke htdocs
3. Pada folder studi kasus buat folder baru di dalamnya dengan nama assets
4. Copy bootstrap, ckeditor, jquery dan font awesome ke folder assets
5. Buka di text editor



SETTINGAN AWAL

1. Buat file .htaccess

```
1 RewriteEngine On
2 RewriteCond %{REQUEST_FILENAME} !-f
3 RewriteCond %{REQUEST_FILENAME} !-d
4 RewriteRule ^(.*)$ index.php/$1 [L]
```

The screenshot shows the VS Code interface with the Explorer sidebar open, displaying a folder structure for 'STUDI-KASUS' containing 'application', 'assets', 'system', 'user_guide', '.editorconfig', '.gitignore', '.htaccess', and 'composer.json'. The '.htaccess' file is selected in the Explorer, and its contents are displayed in the main editor area.

2. Set autoload.php

- Library → jika load database akan error karena database belum dibuat

```
60 */
61 $autoload['libraries'] = array('session', 'form_validation');
```

The screenshot shows the VS Code interface with the Explorer sidebar open, displaying a folder structure for 'STUDI-KASUS' containing 'application', 'cache', 'config', 'autoload.php', and 'config.php'. The 'autoload.php' file is selected in the Explorer, and its contents are displayed in the main editor area.

- Helper

```
91 */
92 $autoload['helper'] = array('url');
```

The screenshot shows the VS Code interface with the Explorer sidebar open, displaying a folder structure for 'STUDI-KASUS' containing 'application', 'cache', 'config', 'autoload.php', and 'config.php'. The 'autoload.php' file is selected in the Explorer, and its contents are displayed in the main editor area.

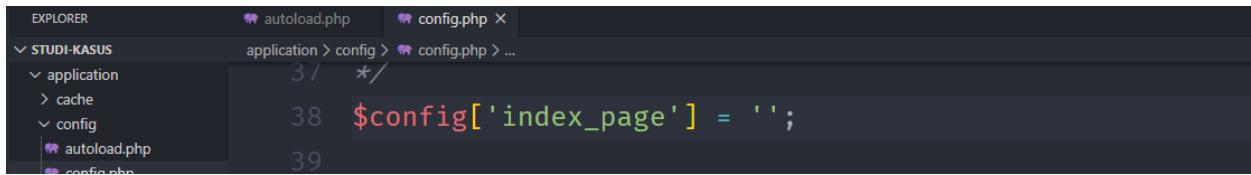
3. Set config.php

- Base_url

```
25 */
26 $config['base_url'] = 'http://localhost/studi-kasus/';
```

The screenshot shows the VS Code interface with the Explorer sidebar open, displaying a folder structure for 'STUDI-KASUS' containing 'application', 'cache', 'config', 'autoload.php', and 'config.php'. The 'config.php' file is selected in the Explorer, and its contents are displayed in the main editor area.

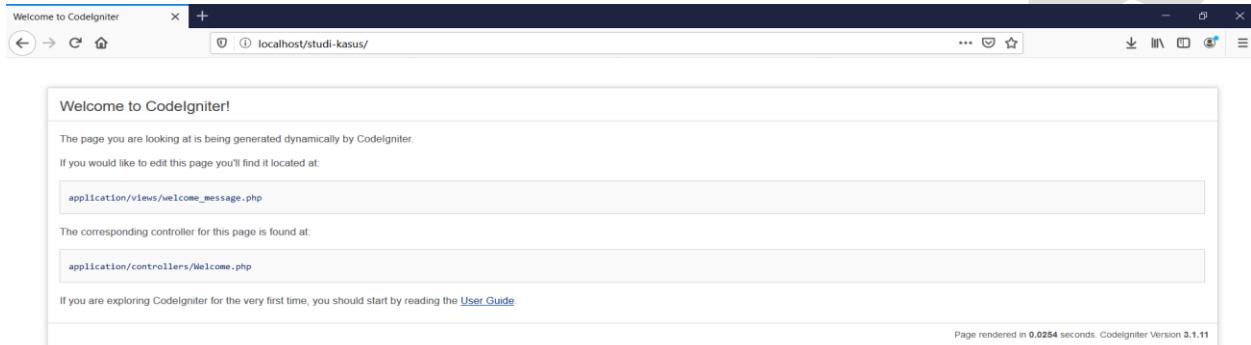
b. Index_page



The screenshot shows a code editor with an Explorer sidebar on the left. The Explorer sidebar lists a project structure under 'STUDI-KASUS': application, cache, config, autoload.php, and config.php. The main area displays the contents of config.php. The code is as follows:

```
autoload.php config.php
STUDI-KASUS application > config > config.php > ...
37      */
38 $config['index_page'] = '';
39 
```

HASIL



TEMPLATING

1. Ketika memulai sebuah project biasanya saya akan memulai dari user management
2. Buat controller user



The screenshot shows the User.php controller file in a code editor. The code defines a User class that extends CI_Controller. The index() method sets the title to 'Data User' and loads the 'user/index' view with the \$data array.

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class User extends CI_Controller
{
    public function index()
    {
        $data['title'] = 'Data User';
        $this->load->view('user/index', $data);
    }
}
```

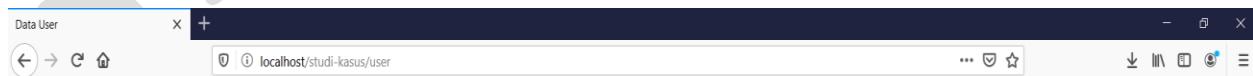
3. View user



The screenshot shows the index.php view file in a code editor. The file contains HTML and PHP code for a user index page. It includes a head section with meta tags and a body section with a main heading.

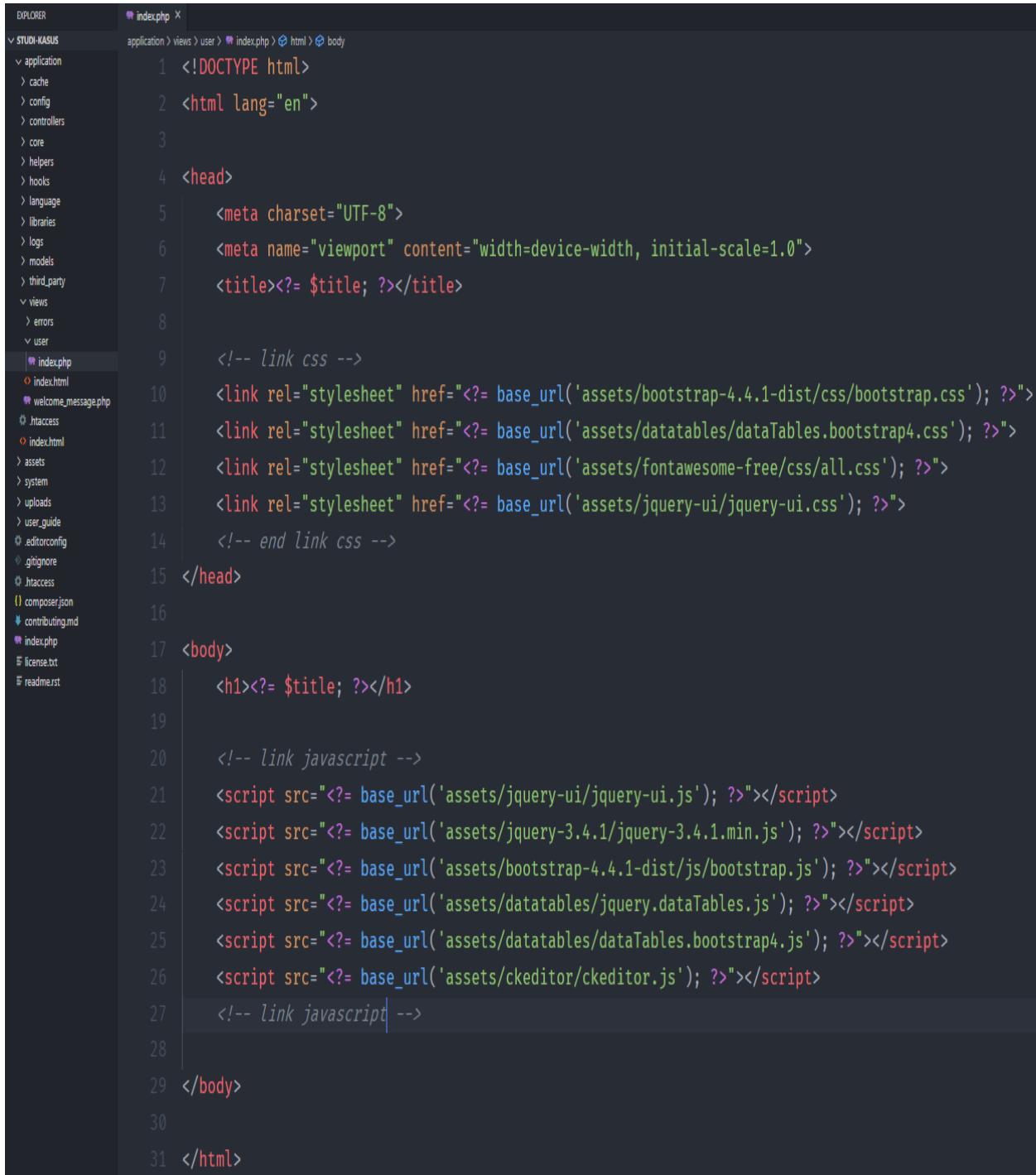
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title><?= $title; ?></title>
</head>
<body>
    <h1><?= $title; ?></h1>
</body>
</html>
```

4. Browser ➡ <http://localhost/studi-kasus/user>



Data User

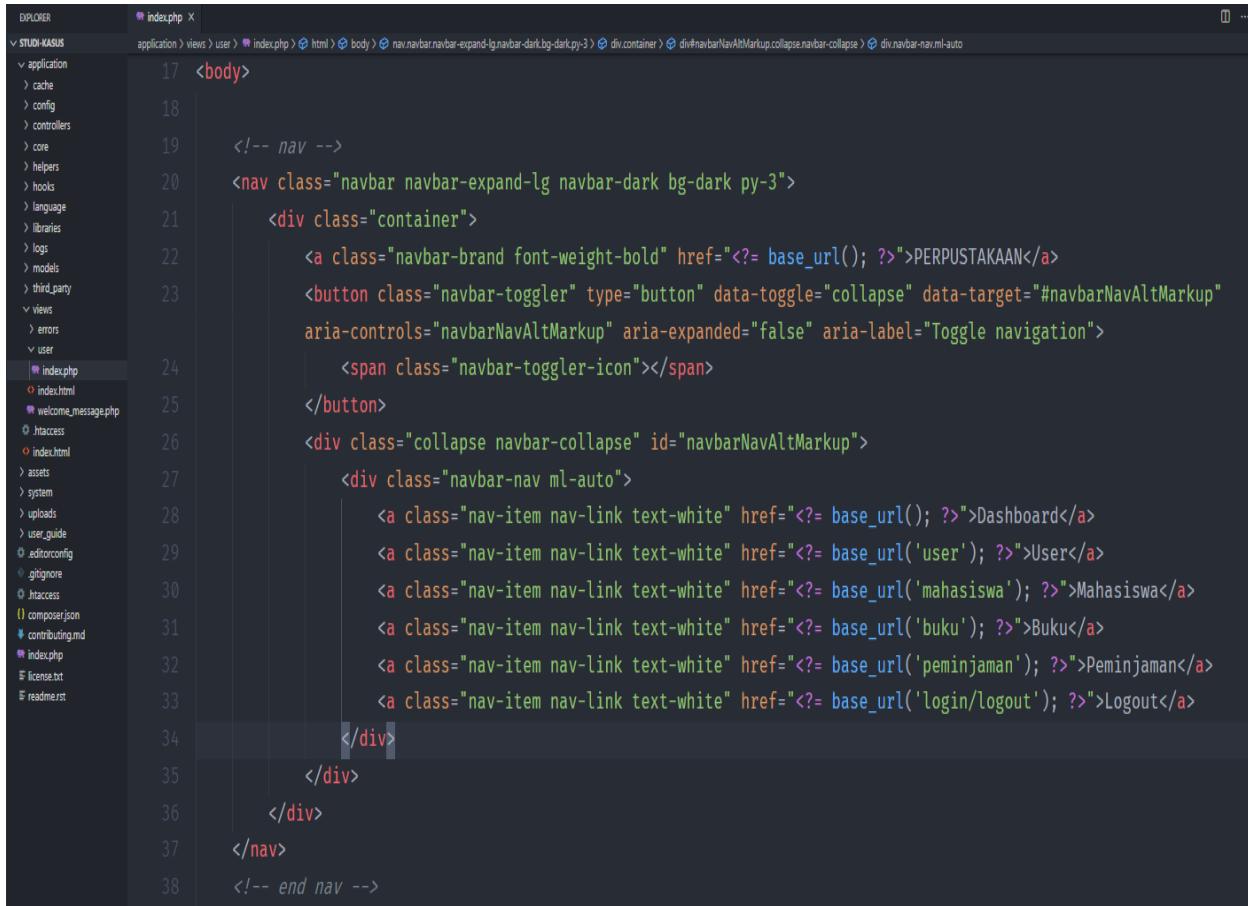
5. Memanggil library pada assets



```
index.php
application > views > user > index.php > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title><?= $title; ?></title>
8
9      <!-- link css -->
10     <link rel="stylesheet" href=<?= base_url('assets/bootstrap-4.4.1-dist/css/bootstrap.css'); ?>>
11     <link rel="stylesheet" href=<?= base_url('assets/datatables/dataTables.bootstrap4.css'); ?>>
12     <link rel="stylesheet" href=<?= base_url('assets/fontawesome-free/css/all.css'); ?>>
13     <link rel="stylesheet" href=<?= base_url('assets/jquery-ui/jquery-ui.css'); ?>>
14     <!-- end link css -->
15 </head>
16
17 <body>
18     <h1><?= $title; ?></h1>
19
20     <!-- link javascript -->
21     <script src=<?= base_url('assets/jquery-ui/jquery-ui.js'); ?>></script>
22     <script src=<?= base_url('assets/jquery-3.4.1/jquery-3.4.1.min.js'); ?>></script>
23     <script src=<?= base_url('assets/bootstrap-4.4.1-dist/js/bootstrap.js'); ?>></script>
24     <script src=<?= base_url('assets/datatables/jquery.dataTables.js'); ?>></script>
25     <script src=<?= base_url('assets/datatables/dataTables.bootstrap4.js'); ?>></script>
26     <script src=<?= base_url('assets/ckeditor/ckeditor.js'); ?>></script>
27     <!-- link javascript -->
28
29 </body>
30
31 </html>
```

6. Untuk mengecek apakah pemanggilannya sudah benar bisa klik kanan pada browser dan pilih view page source. Kemudian bisa membuka link dari pemanggilan filenya.
 - ➔ Jika benar filenya akan terbuka
 - ➔ Jika salah maka akan menampilkan script html 404 Page Not Found

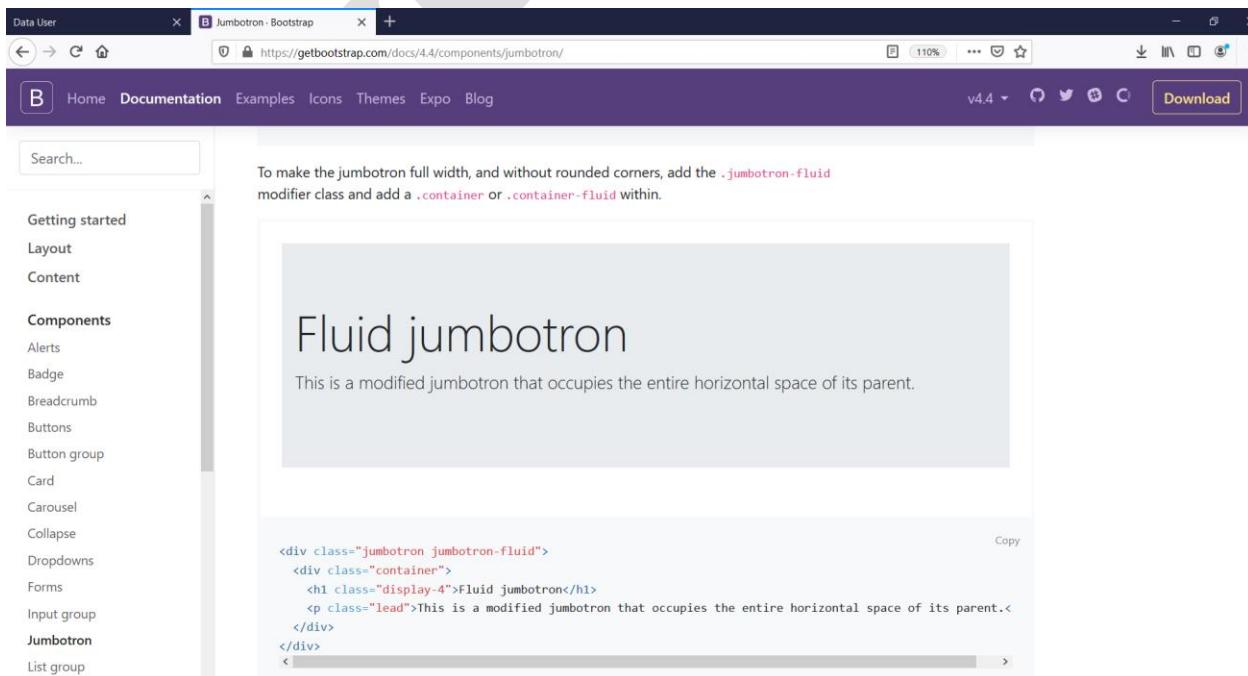
7. Navigasi ➔ sudah pernah dijelaskan pada tutorial bootstrap



A screenshot of a code editor showing the navigation bar code. The file is named index.php and is located in the application/views/user directory. The code uses Bootstrap's navbar components, including a container, collapse, and various nav-item links. The code is color-coded for syntax highlighting.

```
<!-- nav -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark py-3">
  <div class="container">
    <a class="navbar-brand font-weight-bold" href="= base_url(); ?">>PERPUSTAKAAN</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
      <div class="navbar-nav ml-auto">
        <a class="nav-item nav-link text-white" href="= base_url(); ?">>Dashboard</a>
        <a class="nav-item nav-link text-white" href="= base_url('user') ; ?">>User</a>
        <a class="nav-item nav-link text-white" href="= base_url('mahasiswa') ; ?">>Mahasiswa</a>
        <a class="nav-item nav-link text-white" href="= base_url('buku') ; ?">>Buku</a>
        <a class="nav-item nav-link text-white" href="= base_url('peminjaman') ; ?">>Peminjaman</a>
        <a class="nav-item nav-link text-white" href="= base_url('login/logout') ; ?">>Logout</a>
      </div>
    </div>
  </div>
</nav>
<!-- end nav -->
```

8. Judul Halaman ➔ seachr pada bootstrap jumbotron ➔ fluid jumbotron



A screenshot of a web browser displaying the Fluid jumbotron component documentation from getbootstrap.com. The page shows a large, full-width jumbotron with rounded corners removed. The title is "Fluid jumbotron" and the text below it reads: "This is a modified jumbotron that occupies the entire horizontal space of its parent." Below the visual example is the corresponding HTML code.

To make the jumbotron full width, and without rounded corners, add the `.jumbotron-fluid` modifier class and add a `.container` or `.container-fluid` within.

Fluid jumbotron

This is a modified jumbotron that occupies the entire horizontal space of its parent.

```
<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <h1 class="display-4">Fluid jumbotron</h1>
    <p class="lead">This is a modified jumbotron that occupies the entire horizontal space of its parent.</p>
  </div>
</div>
```



EXPLORER index.php

STUDI-KASUS

- application
 - cache
 - config
 - controllers
 - core
 - helpers
 - hooks
 - language
 - libraries
 - logs
 - models
 - third_party
- views
 - errors
 - user
- index.php

```
application > views > user > index.php > html > body > nav.navbar.navbar-expand-lg.navbar-dark.bg-dark.py-3 > div.container > div#navbarNavA
```

39
40 <!-- judul -->
41

42

43

<?= \$title; ?>

44

45

46 <!-- end judul -->

9. Footer



EXPLORER index.php

STUDI-KASUS

- application
 - cache
 - config
 - controllers
 - core
 - helpers
 - hooks
 - language
 - libraries
 - logs
 - models
 - third_party
- views
 - errors
 - cli
 - html
 - index-backup.php
 - index.html
 - index.php
 - welcome_message.php
 - assets
 - system
 - uploads
 - user_guide
 - editorconfig
 - ignore
 - htaccess
- composer.json
- contributing.md
- index.php
- license.txt
- readme.md

```
application > views > user > index.php > html > body
```

46 <!-- end judul -->
47
48 <!-- footer -->
49
50

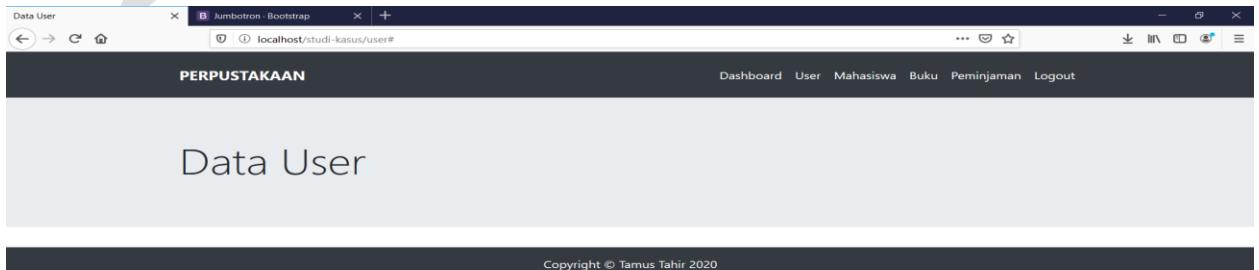
51

Copyright © Tamus Tahir 2020

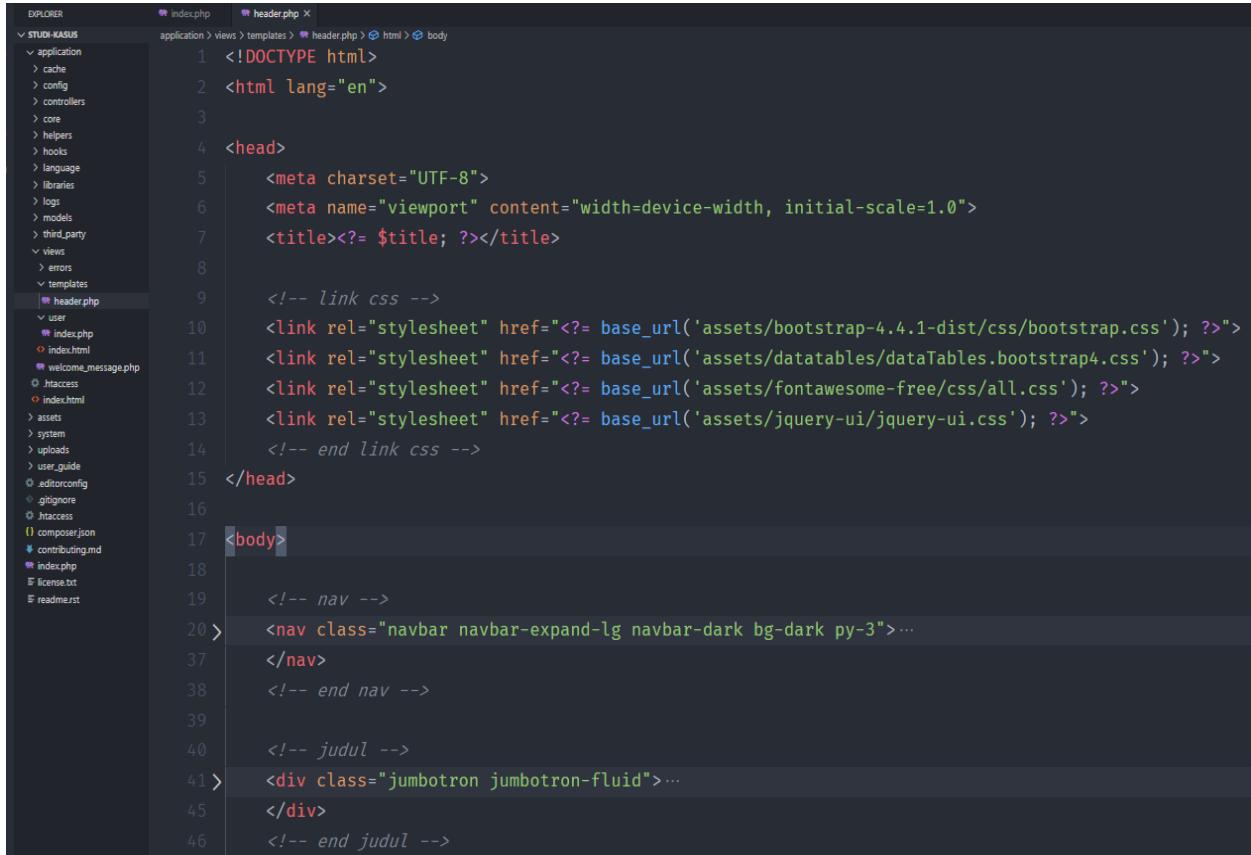
52

53
54 <!-- end footer -->
55
56 <!-- link javascript -->
57 <script src="= base_url('assets/jquery-ui/jquery-ui.js'); ?>"/></script><br/58 <script src="= base_url('assets/jquery-3.4.1/jquery-3.4.1.min.js'); ?>"/></script><br/59 <script src="= base_url('assets/bootstrap-4.4.1-dist/js/bootstrap.js'); ?>"/></script><br/60 <script src="= base_url('assets/datatables/jquery.dataTables.js'); ?>"/></script><br/61 <script src="= base_url('assets/datatables/dataTables.bootstrap4.js'); ?>"/></script><br/62 <script src="= base_url('assets/ckeditor/ckeditor.js'); ?>"/></script><br/63 <!-- link javascript -->
64
65

10. Browser



11. Sekarang pecah filenya menjadi beberapa bagian
12. Buat folder baru pada view dengan nama templates
13. Header.php

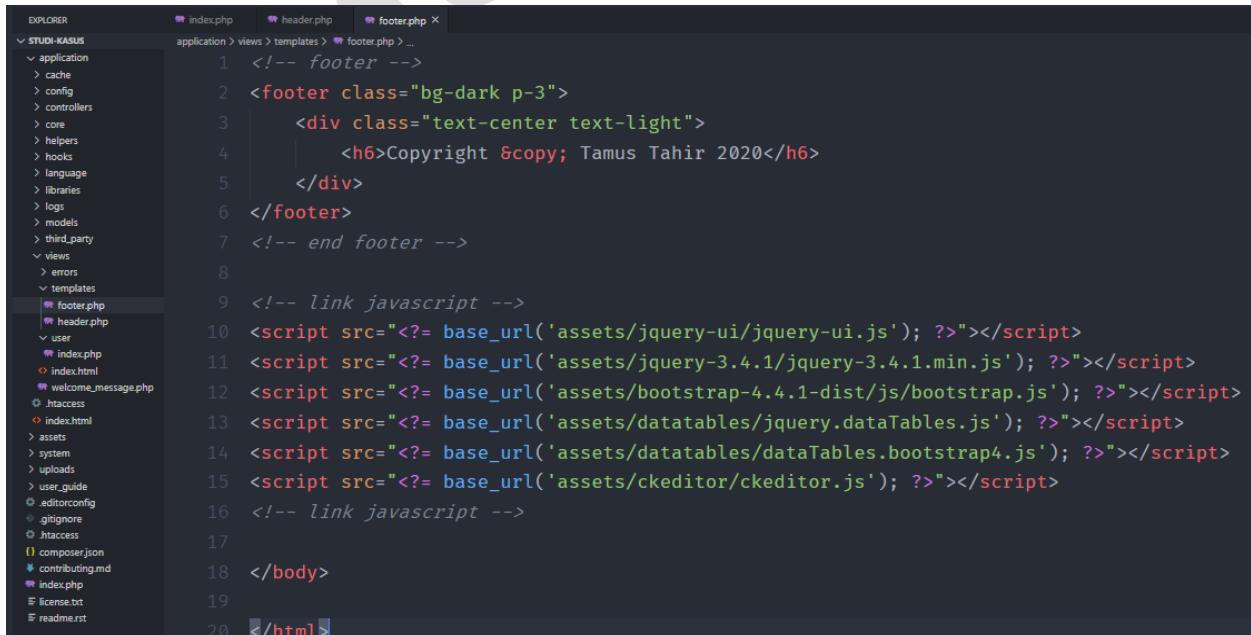


```

EXPLORER index.php header.php
STUDI-KASUS application > views > templates > header.php > body
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title><?= $title; ?></title>
8
9      <!-- link css -->
10     <link rel="stylesheet" href=<?= base_url('assets/bootstrap-4.4.1-dist/css/bootstrap.css'); ?>">
11     <link rel="stylesheet" href=<?= base_url('assets/datatables/dataTables.bootstrap4.css'); ?>">
12     <link rel="stylesheet" href=<?= base_url('assets/fontawesome-free/css/all.css'); ?>">
13     <link rel="stylesheet" href=<?= base_url('assets/jquery-ui/jquery-ui.css'); ?>">
14     <!-- end link css -->
15 </head>
16
17 <body>
18
19     <!-- nav -->
20 >     <nav class="navbar navbar-expand-lg navbar-dark bg-dark py-3">...
21     </nav>
22     <!-- end nav -->
23
24     <!-- judul -->
25 >     <div class="jumbotron jumbotron-fluid">...
26     </div>
27     <!-- end judul -->

```

14. Footer.php



```

EXPLORER index.php header.php footer.php
STUDI-KASUS application > views > templates > footer.php > ...
1  <!-- footer -->
2  <footer class="bg-dark p-3">
3      <div class="text-center text-light">
4          <h6>Copyright &copy; Tamus Tahir 2020</h6>
5      </div>
6  </footer>
7  <!-- end footer -->
8
9  <!-- link javascript -->
10 <script src=<?= base_url('assets/jquery-ui/jquery-ui.js'); ?>"></script>
11 <script src=<?= base_url('assets/jquery-3.4.1/jquery-3.4.1.min.js'); ?>"></script>
12 <script src=<?= base_url('assets/bootstrap-4.4.1-dist/js/bootstrap.js'); ?>"></script>
13 <script src=<?= base_url('assets/datatables/jquery.dataTables.js'); ?>"></script>
14 <script src=<?= base_url('assets/datatables/dataTables.bootstrap4.js'); ?>"></script>
15 <script src=<?= base_url('assets/ckeditor/ckeditor.js'); ?>"></script>
16 <!-- link javascript -->
17
18 </body>
19
20 </html>

```

15. Panggil file template tersebut pada controller user

The screenshot shows a code editor with the User.php file open. The code defines a User controller that extends CI_Controller. It contains an index() method which loads the header, user/index template, and footer. The code editor interface includes tabs for index.php, header.php, footer.php, and User.php.

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class User extends CI_Controller
{
    public function index()
    {
        $data['title'] = 'Data User';
        $this->load->view('templates/header', $data);
        $this->load->view('user/index', $data);
        $this->load->view('templates/footer', $data);
    }
}
```

16. Sekarang view user bisa dikosongkan

The screenshot shows a code editor with the index.php file open. The file is currently empty, indicated by a single blank line. The code editor interface includes tabs for index.php, header.php, footer.php, and User.php.

```
1
```

17. Browser



CRUD USER

CREATE USER

1. Buat database dengan nama perpustakaan
2. Buat tabel dengan nama user dengan struktur:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_user	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	level	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
5	nama	varchar(128)	utf8mb4_general_ci		No	None			Change Drop More

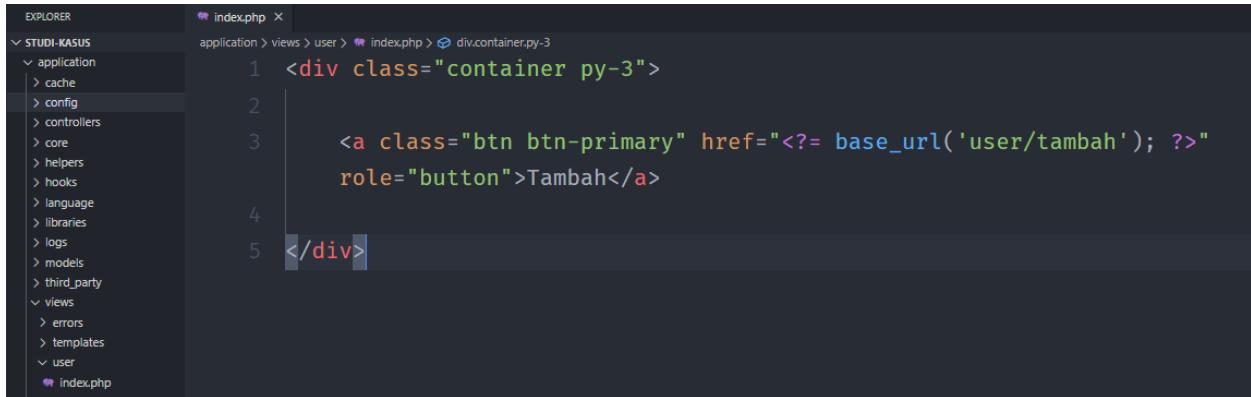
3. Koneksi database
 - a. Load library database

```
EXPLORER index.php autoload.php header.php footer.php User.php
STUDI-KASUS application > config > autoload.php > ...
60 */
61 $autoload['libraries'] = array('session', 'form_validation', 'database');
```

- b. Setting database

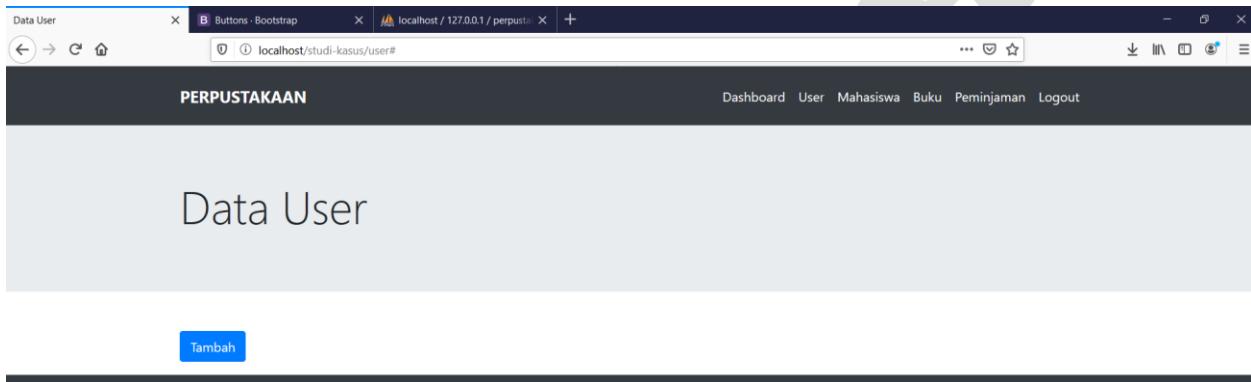
```
EXPLORER index.php autoload.php database.php header.php footer.php User.php
STUDI-KASUS application > config > database.php > ...
76 $db['default'] = array(
77     'dsn'      => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'perpustakaan',
```

4. Buat tombol tambah pada view user → cari di bootstrap → button



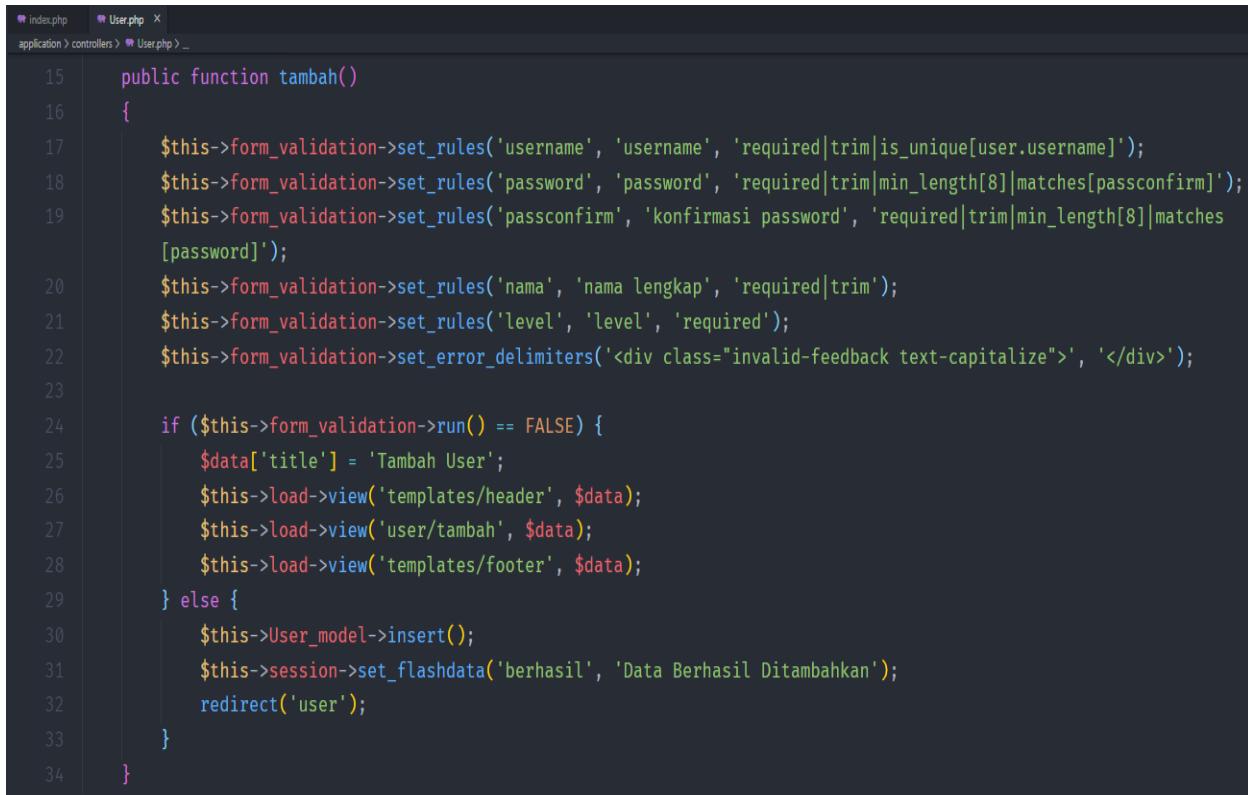
```
EXPLORER index.php ×
STUDI-KASUS application > views > user > index.php > div.container.py-3
1  <div class="container py-3">
2
3    <a class="btn btn-primary" href="= base_url('user/tambah'); ?&gt;" role="button"&gt;Tambah&lt;/a&gt;
4
5  &lt;/div&gt;</pre
```

5. Browser



6. Method tambah controller user

- ⇒ Is_unique → nilai field harus unik (tidak boleh ada yang sama), dengan 2 parameter nama table dan field yang akan diperiksa
- ⇒ Trim → menghilangkan karakter spasi di awal dan akhir string
- ⇒ Min_length → Panjang minimal karakter
- ⇒ Matches → nilainya harus sama
- ⇒ Set error delimiters → jika terjadi error kita bisa mengirikan sesuatu, untuk kali ini kita mengirimkan html dengan class css bootsrap
- ⇒ Invalid feedback → membuat tulisan error pada form berwarna merah



```

15     public function tambah()
16     {
17         $this->form_validation->set_rules('username', 'username', 'required|trim|is_unique[user.username]');
18         $this->form_validation->set_rules('password', 'password', 'required|trim|min_length[8]|matches[passconfirm]');
19         $this->form_validation->set_rules('passconfirm', 'konfirmasi password', 'required|trim|min_length[8]|matches
20             [password]');
21         $this->form_validation->set_rules('nama', 'nama lengkap', 'required|trim');
22         $this->form_validation->set_rules('level', 'level', 'required');
23         $this->form_validation->set_error_delimiters('<div class="invalid-feedback text-capitalize">', '</div>');
24
25         if ($this->form_validation->run() == FALSE) {
26             $data['title'] = 'Tambah User';
27             $this->load->view('templates/header', $data);
28             $this->load->view('user/tambah', $data);
29             $this->load->view('templates/footer', $data);
30         } else {
31             $this->User_model->insert();
32             $this->session->set_flashdata('berhasil', 'Data Berhasil Ditambahkan');
33             redirect('user');
34         }
35     }

```

7. Contstruct



```

1 <?php
2 defined('BASEPATH') or exit('No direct script access allowed');
3
4 class User extends CI_Controller
5 {
6
7
8     public function __construct()
9     {
10        parent::__construct();
11        $this->load->model('User_model');
12    }
13

```

8. Buat file baru user_model.php pada folder modal

- ⇒ Password hash → encrypt password dengan algoritma hash
- ⇒ 2 parameter, passwordnya dan metodenya

EXPLORER

- STUDI-KASUS
- application
 - > cache
 - > config
 - > controllers
 - > core
 - > helpers
 - > hooks
 - > language
 - > libraries
 - > logs
 - models
 - index.html
 - User_model.php
 - > third_party
 - views
 - .htaccess
 - index.html
 - assets
 - > bootstrap-4.4.1-dist
 - > ckeditor
 - > databases
 - > fontawesome-free
 - > jquery-3.4.1
 - > system
 - > user_guide
 - editordconfig
 - gitignore
 - .htaccess
 - composer.json
 - contributing.md
 - index.php
 - license.txt
 - readme.rst

User_model.php

```

1 <?php
2 defined('BASEPATH') or exit('No direct script access allowed');
3
4 class User_model extends CI_Model
5 {
6
7     public function insert()
8     {
9
10         $data = [
11             'username' => $this->input->post('username'),
12             'password' => password_hash($this->input->post('password'), PASSWORD_DEFAULT),
13             'nama' => $this->input->post('nama'),
14             'level' => $this->input->post('level')
15         ];
16
17         $this->db->insert('user', $data);
18     }

```

9. View tambah user

EXPLORER

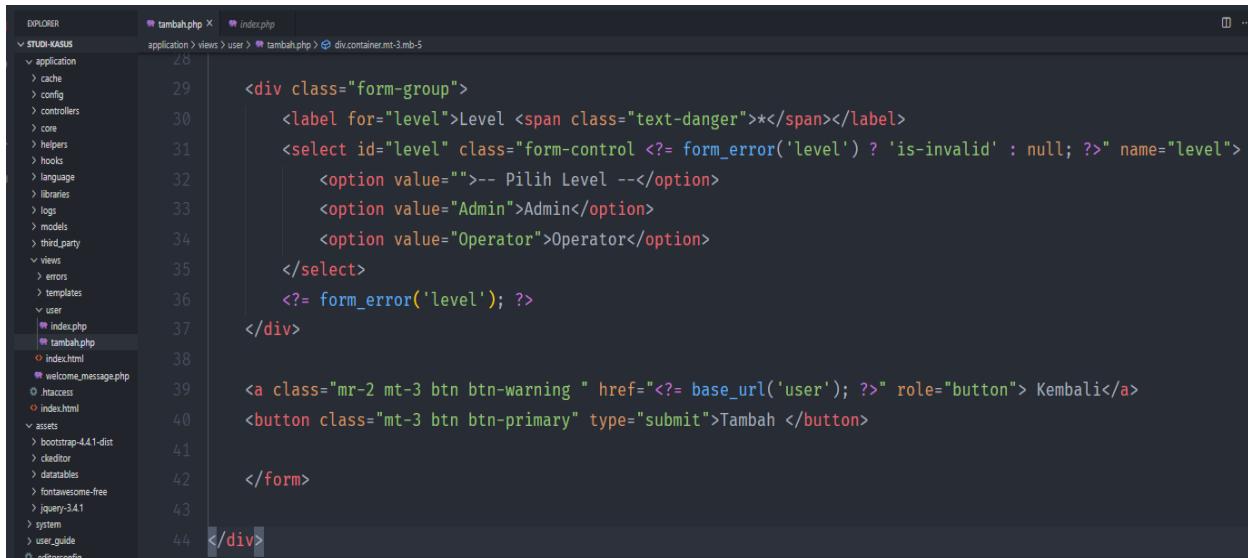
- STUDI-KASUS
- application
 - > cache
 - > config
 - > controllers
 - > core
 - > helpers
 - > hooks
 - > language
 - > libraries
 - > logs
 - models
 - > third_party
 - views
 - > errors
 - > templates
 - user
 - index.php
 - tambah.php
 - index.html
 - welcome_message.php
 - .htaccess
 - index.html
 - assets
 - > bootstrap-4.4.1-dist
 - > ckeditor
 - > databases
 - > fontawesome-free
 - > jquery-3.4.1
 - > system
 - > user_guide
 - editordconfig
 - gitignore
 - .htaccess
 - composer.json
 - contributing.md
 - index.php
 - license.txt
 - readme.rst

tambah.php

```

1 <div class="container mt-3 mb-5">
2
3     <?= form_open(); ?>
4
5     <div class="form-group">
6         <label for="username">Username <span class="text-danger"></span></label>
7         <input type="text" class="form-control <?= form_error('username') ? 'is-invalid' : null; ?>" id="username" name="username" autocomplete="off">
8         <?= form_error('username'); ?>
9     </div>
10
11    <div class="form-group">
12        <label for="password">Password <span class="text-danger"></span></label>
13        <input type="password" class="form-control <?= form_error('password') ? 'is-invalid' : null; ?>" id="password" name="password" autocomplete="off">
14        <?= form_error('password'); ?>
15    </div>
16
17    <div class="form-group">
18        <label for="passconfirm">Konfirmasi Password <span class="text-danger"></span></label>
19        <input type="password" class="form-control <?= form_error('passconfirm') ? 'is-invalid' : null; ?>" id="passconfirm" name="passconfirm" autocomplete="off">
20        <?= form_error('passconfirm'); ?>
21    </div>
22
23    <div class="form-group">
24        <label for="nama">Nama Lengkap <span class="text-danger"></span></label>
25        <input type="text" class="form-control <?= form_error('nama') ? 'is-invalid' : null; ?>" id="nama" name="nama" autocomplete="off">
26        <?= form_error('nama'); ?>
27    </div>

```



```

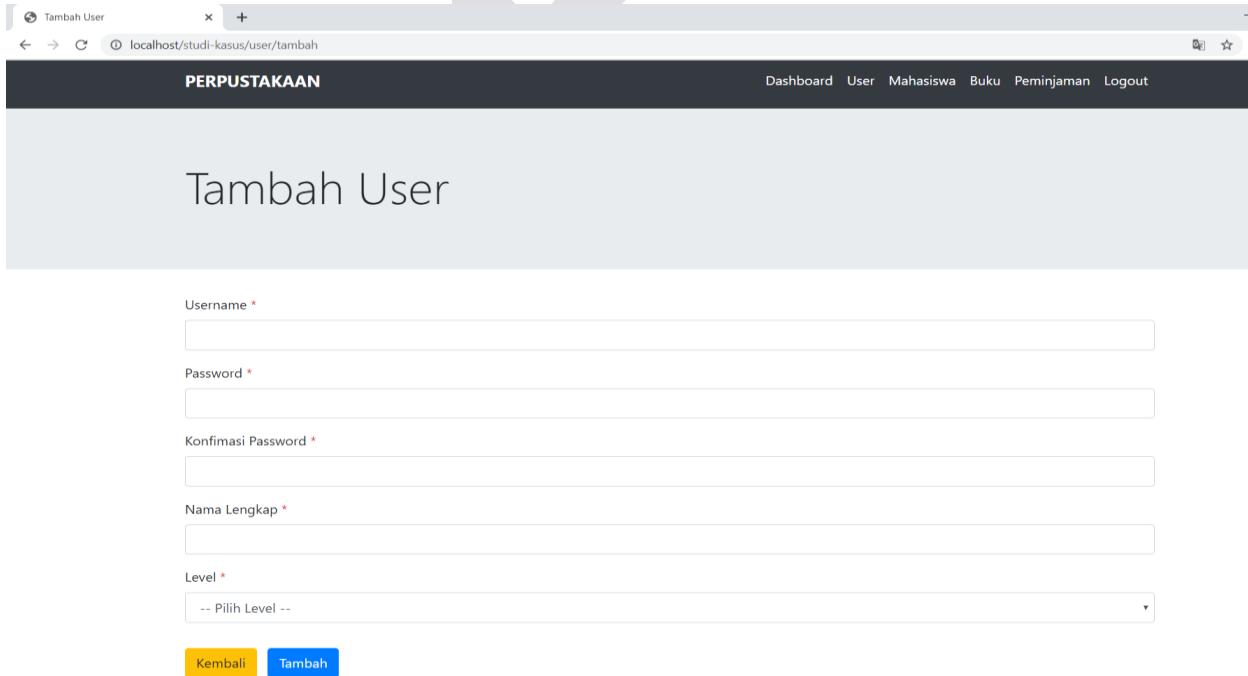
STUDI-KASUS
application > views > user > tambah.php > div.container.mt-3.mb-5
28
29     <div class="form-group">
30         <label for="level">Level <span class="text-danger">*</span></label>
31         <select id="level" class="form-control <?= form_error('level') ? 'is-invalid' : null; ?>" name="level">
32             <option value="">-- Pilih Level --</option>
33             <option value="Admin">Admin</option>
34             <option value="Operator">Operator</option>
35         </select>
36         <?= form_error('level') ?>
37     </div>
38
39     <a class="mr-2 mt-3 btn btn-warning" href="= base_url('user'); ?&gt;" role="button"&gt; Kembali &lt;/a&gt;
40     &lt;button class="mt-3 btn btn-primary" type="submit"&gt;Tambah &lt;/button&gt;
41
42 &lt;/form&gt;
43
44 &lt;/div&gt;
</pre

```

10. Membuat kondisi dimana jika formnya error kita akan mencetak sebuah class dengan nama is-invalid yang berfungsi membuat input pada form berwarna merah

```
<?= form_error('username') ? 'is-invalid' : null; ?>
```

11. Browser



PERPUSTAKAAN

Tambah User

Username *

Password *

Konfirmasi Password *

Nama Lengkap *

Level *

-- Pilih Level --

Kembali Tambah

12. Kosongkan form dan pilih tambah

Username *

The Username Field Is Required.

Password *

The Password Field Is Required.

Konfirmasi Password *

The Konfirmasi Password Field Is Required.

Nama Lengkap *

The Nama Lengkap Field Is Required.

Level *

-- Pilih Level --

The Level Field Is Required.

[Kembali](#) [Tambah](#)

13. Index user, cetak notifikasi jika berhasil ➔ cari di bootstrap ➔ alert

The screenshot shows a code editor with a file named 'index.php' open. The code is as follows:

```
<div class="container py-3">
    <a class="btn btn-primary" href="= base_url('user/tambah'); ?" role="button">Tambah</a>
    <?php if ($this->session->flashdata('berhasil')) : ?>
        <div class="alert alert-primary my-3" role="alert">
            <?= $this->session->flashdata('berhasil'); ?>
        </div>
    <?php endif ?>
</div>
```

14. Pengujian

⇒ Jika password dan konfirmasi password tidak sama

Password *

The Password Field Does Not Match The Konfirmasi Password Field.

⇒ Jika password atau konfirmasi password panjang karakternya kurang dari 8

Password *

The Password Field Must Be At Least 8 Characters In Length.

⇒ Input terlebih dahulu 1 data, kemudian input data baru dengan username yang sama

Username *

The Username Field Must Contain A Unique Value.

⇒ Jika berhasil



⇒ Buatlah 2 akun dengan level yang berbeda

READ

1. Controller user method index

```
EXPLORER User.php
STUDI-KASUS application > controllers > User.php > PHP Intelephense > User
User.php
public function index()
{
    $data['title'] = 'Data User';
    $data['user'] = $this->User_model->get();

    $this->load->view('templates/header', $data);
    $this->load->view('user/index', $data);
    $this->load->view('templates/footer', $data);
}
```

2. Modal user

```
EXPLORER User.php User_model.php
STUDI-KASUS application > models > User_model.php > PHP Intelephense > User_model
User_model.php
1 <?php
2 defined('BASEPATH') or exit('No direct script access allowed');
3
4 class User_model extends CI_Model
5 {
6     public function get()
7     {
8         return $this->db->get('user')->result();
9     }
10
```

3. View user

The image shows two screenshots of a code editor, likely PhpStorm, displaying PHP code related to user management.

Top Screenshot: This screenshot shows the beginning of a table definition. The code includes the opening `<div class="table-responsive my-3">` and the start of a `<table>` element. The table has five columns defined by `<th>` elements with scopes "col", "Username", "Level", "Nama", and "Aksi".

```
<div class="table-responsive my-3">
    <table class="table table-striped table-bordered text-center">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Username</th>
                <th scope="col">Level</th>
                <th scope="col">Nama</th>
                <th scope="col">Aksi</th>
            </tr>
        </thead>
```

Bottom Screenshot: This screenshot shows the continuation of the table definition. It includes the `<tbody>` section, which contains multiple `<tr>` rows. Each row uses PHP loops (`<?php $i = 1; ?>` and `<?php foreach ($user as $u) : ?>`) to generate table rows. The `<td>` elements within each row correspond to the columns defined in the header. The loop increments `$i` and ends with `<?php endforeach; ?>`.

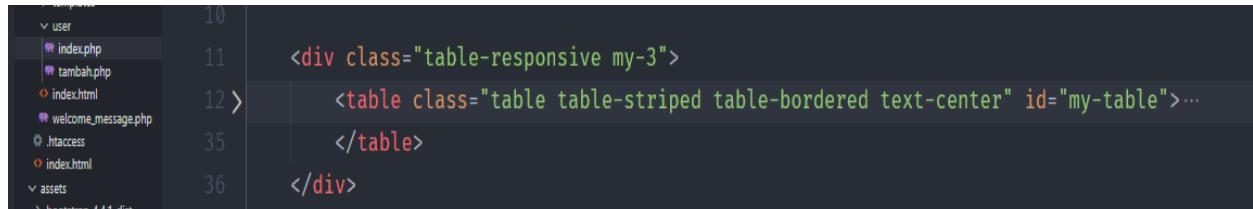
```
<tbody>
    <?php $i = 1; ?>
    <?php foreach ($user as $u) : ?>
        <tr>
            <th scope="row"><?= $i; ?></th>
            <td><?= $u->username; ?></td>
            <td><?= $u->level; ?></td>
            <td><?= $u->nama; ?></td>
            <td></td>
        </tr>
        <?php $i++ ?>
    <?php endforeach; ?>
</tbody>
</table>
</div>
```

4. Browser

The image shows a screenshot of a web browser window titled "Tables - Bootstrap". The URL in the address bar is `localhost/studi-kasus/user`. The page content is a table titled "Data User" with the following data:

#	Username	Level	Nama	Aksi
1	admin	Admin	Monkey D Luffy	
2	operator	Operator	Roronoa Zoro	

5. Datatables → untuk menggunakannya kita perlu memanggil function datatable
 ⇒ Buat id pada table

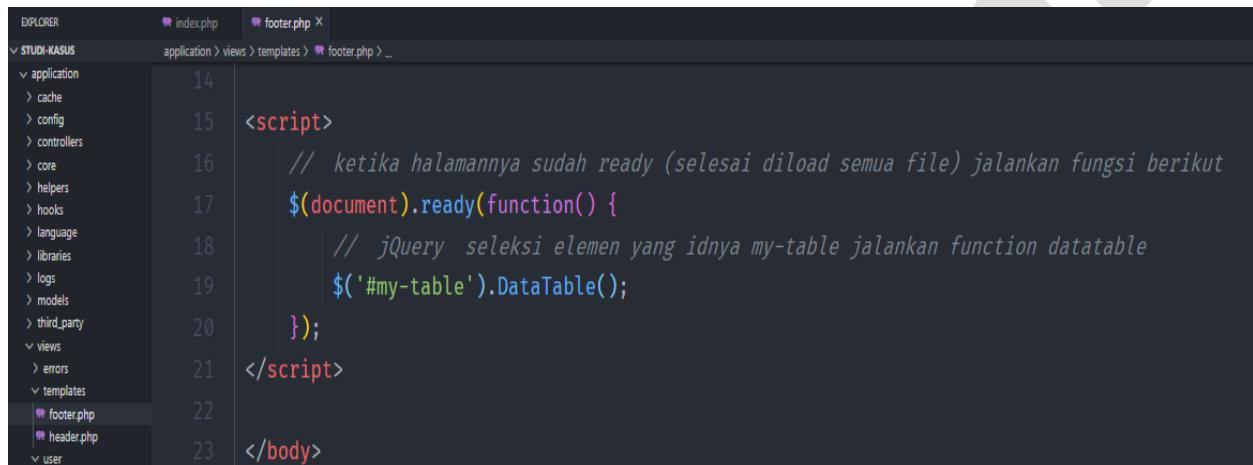


```

10
11     <div class="table-responsive my-3">
12 >     <table class="table table-striped table-bordered text-center" id="my-table">...
13 >         <thead>
14 >             <tr>
15 >                 <th>#</th>
16 >                 <th>Username</th>
17 >                 <th>Level</th>
18 >                 <th>Nama</th>
19 >                 <th>Aksi</th>
20 >             </tr>
21 >         </thead>
22 >         <tbody>
23 >             <tr>

```

- ⇒ Jalankan function datatable dengan target id tersebut dengan menggunakan javascript
 ⇒ Footer.php

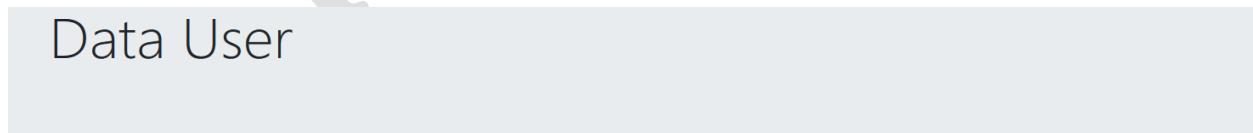


```

EXPLORER          index.php  footer.php X
STUDI-KASUS
application > views > templates > footer.php > ...
14
15 <script>
16 // ketika halamannya sudah ready (selesai diload semua file) jalankan fungsi berikut
17 $(document).ready(function() {
18     // jQuery seleksi elemen yang idnya my-table jalankan function datatable
19     $('#my-table').DataTable();
20 });
21 </script>
22
23 </body>

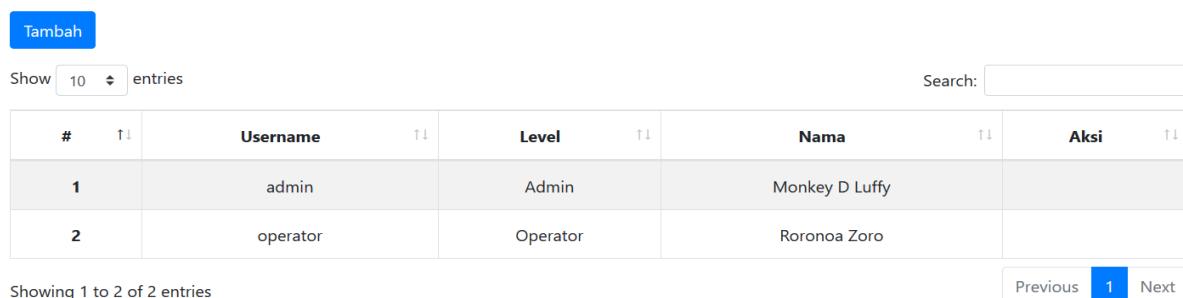
```

6. Browser, Tabelnya sudah memiliki fungsi
 ⇒ live search
 ⇒ Pagination
 ⇒ Show data
 ⇒ Ordering



Data User

#	Username	Level	Nama	Aksi
1	admin	Admin	Monkey D Luffy	
2	operator	Operator	Roronoa Zoro	



#	Username	Level	Nama	Aksi
1	admin	Admin	Monkey D Luffy	
2	operator	Operator	Roronoa Zoro	

Show 10 entries Search:

Showing 1 to 2 of 2 entries

Previous 1 Next

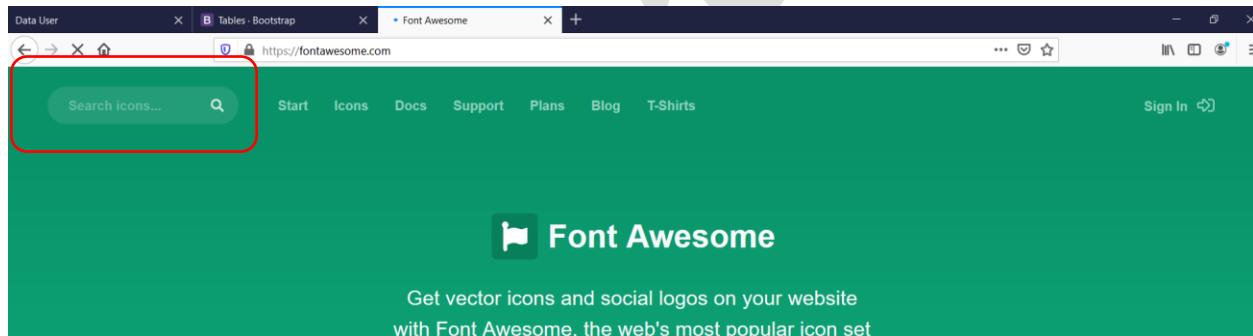
EDIT

1. Pada table user buat link ubah dan hapus

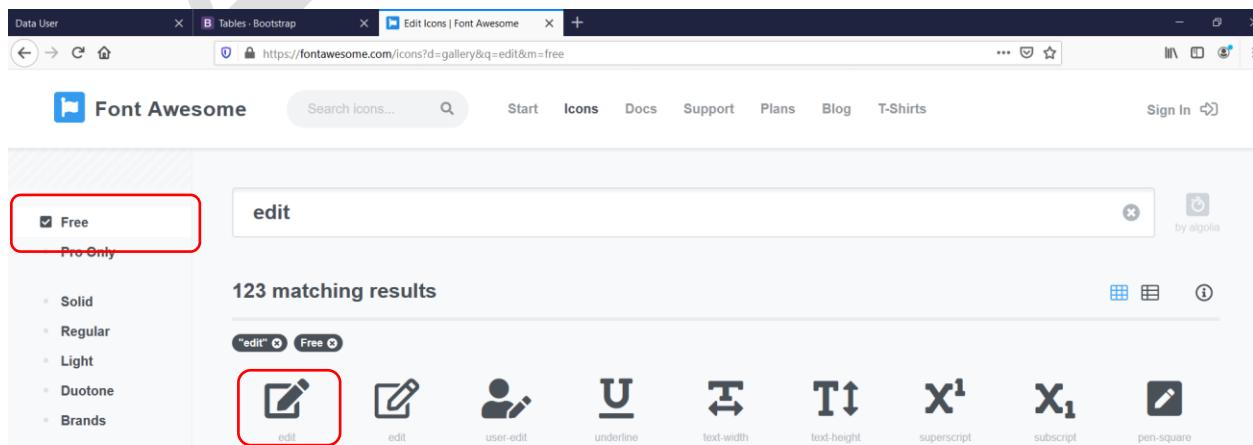
```
> logs 29
> models 30
> third_party 31
< views 32
  > errors 33
  < templates 34
    < footer.php 35
    < header.php 36
  < user 37
    < index.php 38
    < tambah.php
    < index.html
    < welcome_message.php
  < .htaccess
  < index.html
< assets 39
  < bootstrap-4.4.1-dist 40
  < ckeditor 41
  < datatables 42
  < fontawesome-free 43
  < jquery-3.4.1 44
  < system 45
  < user_guide 46
```

```
<td><?= $u->nama; ?></td>
<td>
  <a class="btn btn-outline-warning" href=<?= base_url('user/ubah/' .
$u->id_user); ?>" role="button">
    </a>
  <a class="btn btn-outline-danger" href=<?= base_url('user/hapus/' .
$u->id_user); ?>" onclick="return confirm('Anda Yakin?')"
    role="button">
    </a>
</td>
```

2. Untuk tulisannya kita akan menggunakan icon
3. Cari icon pada font awesome: <https://fontawesome.com/>



4. Pada search insert edit → pilih free → pilih salah satu icon



5. Copy iconnya di dalam link ubah

The screenshot shows the Font Awesome website at https://fontawesome.com/icons/edit?style=solid. The page title is "edit". A search bar contains "Search icons...". Below it are navigation links: Start, Icons, Docs, Support, Plans, and B. A red box highlights the icon code <i class="fas fa-edit"></i>.

6. Cari juga icon untuk hapus → trash → copy iconnya ke dalam link hapus

The screenshot shows a code editor with a sidebar of file navigation. The main area displays PHP code for a user management system. Two specific lines of code are highlighted with boxes: one for the 'ubah' (edit) button and another for the 'hapus' (delete) button. Both lines contain the icon code <i class="fa fa-...></i>.

7. Browser

The screenshot shows a table of user data with columns: #, Username, Level, Nama, and Aksi. The Aksi column contains two buttons: a yellow edit icon and a red trash icon. The table has 2 entries. At the bottom, there is a navigation bar with "Previous", a page number "1", and "Next".

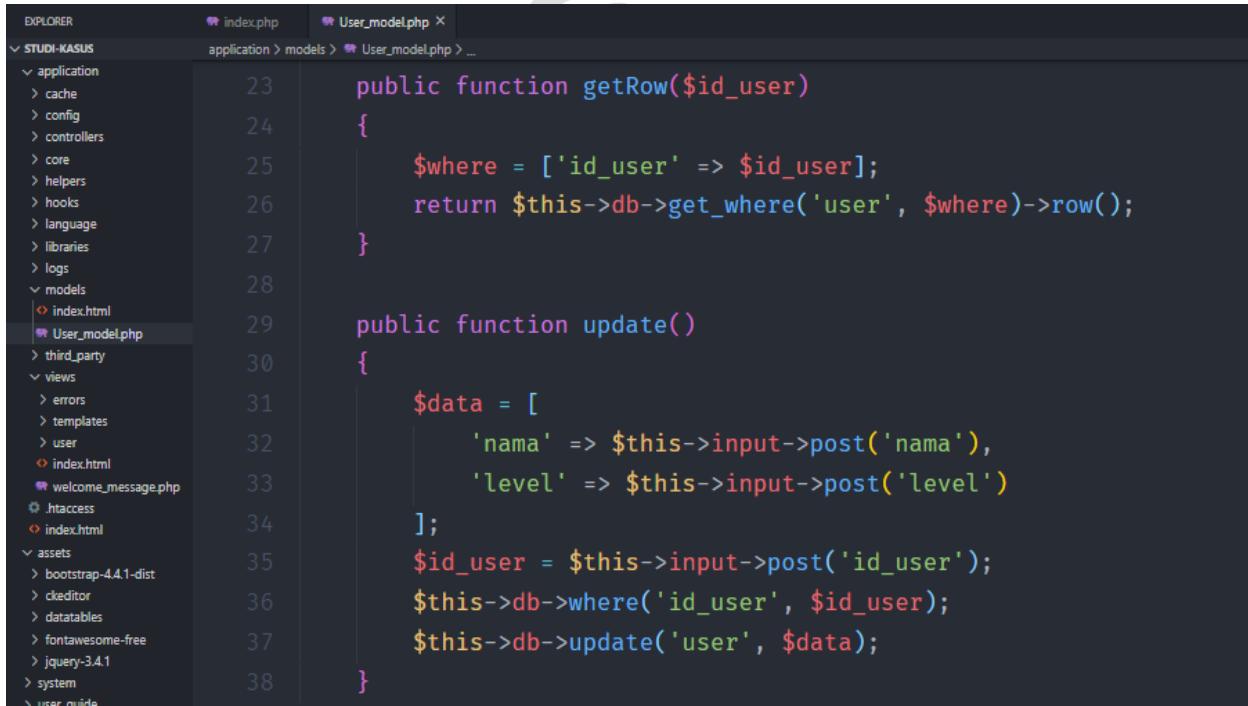
#	Username	Level	Nama	Aksi
1	admin	Admin	Monkey D Luffy	
2	operator	Operator	Roronoa Zoro	

8. Method ubah controller user



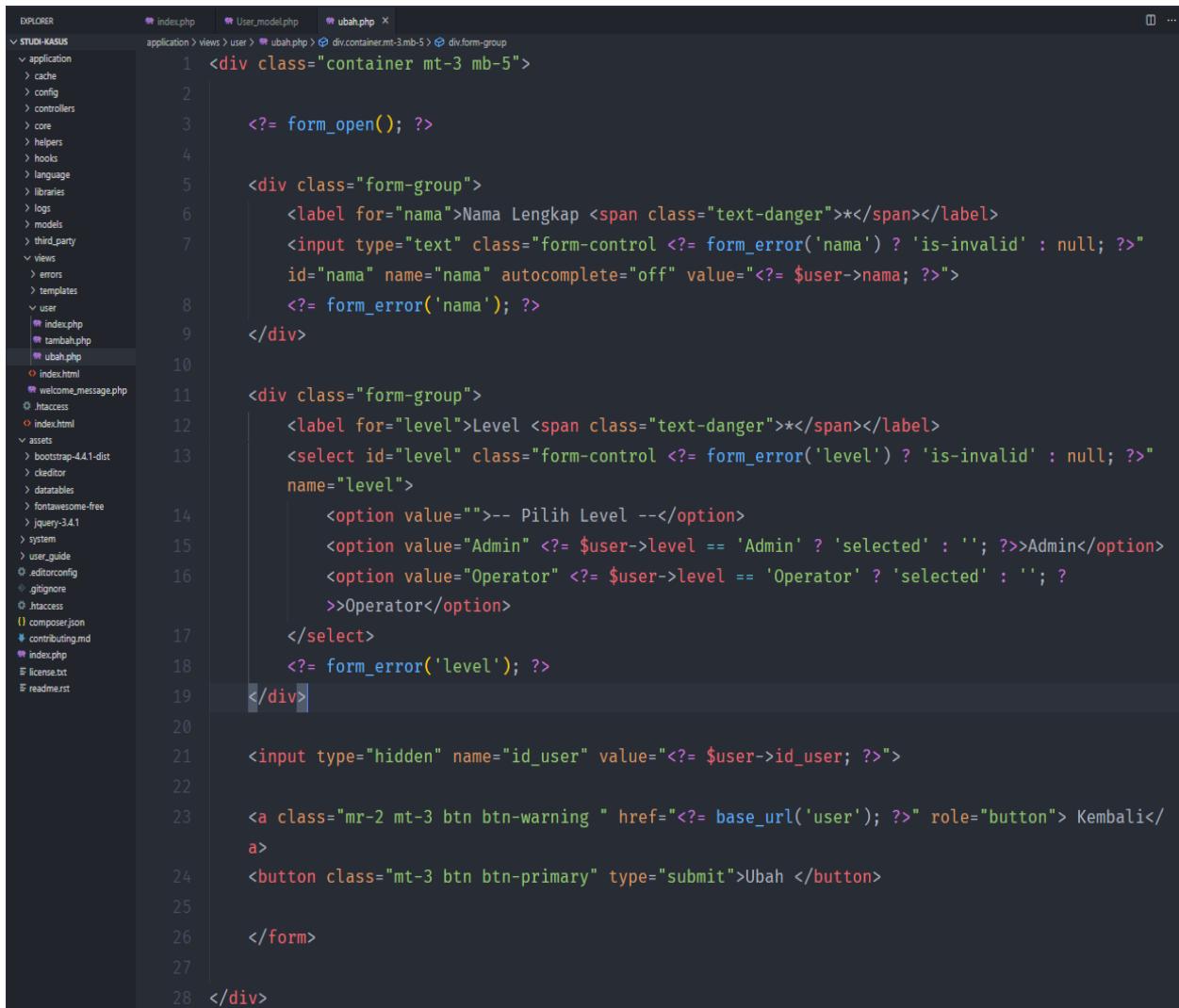
```
EXPLORER index.php User.php
STUDI-KASUS application > controllers > User.php > ...
46     public function ubah($id_user)
47     {
48         $this->form_validation->set_rules('nama', 'nama lengkap', 'required|trim');
49         $this->form_validation->set_rules('level', 'level', 'required');
50         $this->form_validation->set_error_delimiters('<div class="invalid-feedback
51             text-capitalize">', '</div>');
52
53         if ($this->form_validation->run() == FALSE) {
54             $data['title'] = 'Ubah User';
55             $data['user'] = $this->User_model->getRow($id_user);
56
57             $this->load->view('templates/header', $data);
58             $this->load->view('user/ubah', $data);
59             $this->load->view('templates/footer', $data);
60         } else {
61             $this->User_model->update();
62             $this->session->set_flashdata('berhasil', 'Data Berhasil Diubah');
63             redirect('user');
64         }
65     }
66 }
```

9. Method getRow dan method update user model



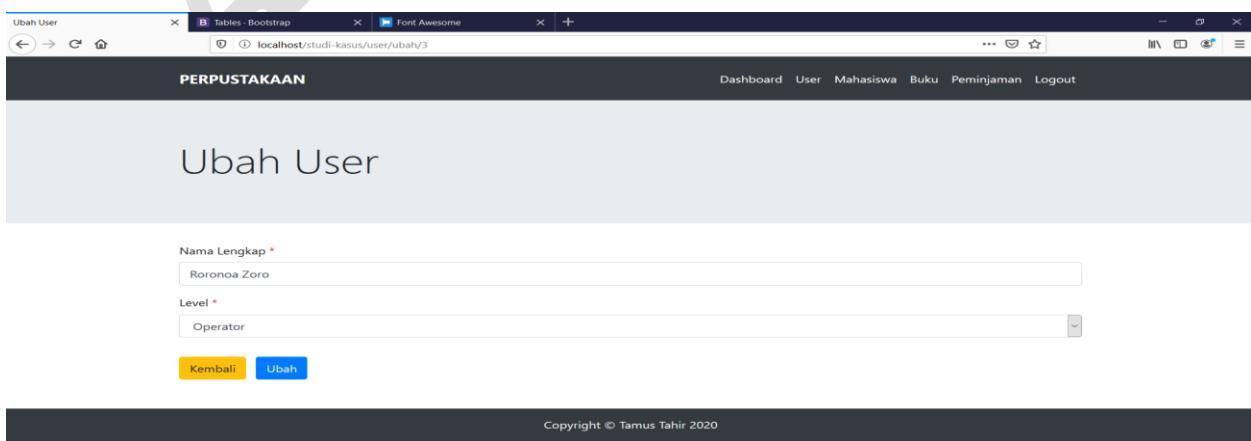
```
EXPLORER index.php User_model.php
STUDI-KASUS application > models > User_model.php > ...
23     public function getRow($id_user)
24     {
25         $where = ['id_user' => $id_user];
26         return $this->db->get_where('user', $where)->row();
27     }
28
29     public function update()
30     {
31         $data = [
32             'nama' => $this->input->post('nama'),
33             'level' => $this->input->post('level')
34         ];
35         $id_user = $this->input->post('id_user');
36         $this->db->where('id_user', $id_user);
37         $this->db->update('user', $data);
38     }
39 }
```

10. View ubah user



```
index.php User_model.php ubah.php
application views > user > ubah.php > div.container.mt-3.mb-5 > div.form-group
1 <div class="container mt-3 mb-5">
2
3 <?= form_open(); ?>
4
5 <div class="form-group">
6     <label for="nama">Nama Lengkap <span class="text-danger">*</span></label>
7     <input type="text" class="form-control <?= form_error('nama') ? 'is-invalid' : null; ?>" id="nama" name="nama" autocomplete="off" value="<?= $user->nama; ?>">
8     <?= form_error('nama'); ?>
9 </div>
10
11 <div class="form-group">
12     <label for="level">Level <span class="text-danger">*</span></label>
13     <select id="level" class="form-control <?= form_error('level') ? 'is-invalid' : null; ?>" name="level">
14         <option value="">-- Pilih Level --</option>
15         <option value="Admin" <?= $user->level == 'Admin' ? 'selected' : ''; ?>>Admin</option>
16         <option value="Operator" <?= $user->level == 'Operator' ? 'selected' : ''; ?>>Operator</option>
17     </select>
18     <?= form_error('level'); ?>
19 </div>
20
21 <input type="hidden" name="id_user" value="<?= $user->id_user; ?>">
22
23 <a class="mr-2 mt-3 btn btn-warning" href="<?= base_url('user'); ?>" role="button"> Kembali </a>
24
25 <button class="mt-3 btn btn-primary" type="submit">Ubah </button>
26
27 </form>
28 </div>
```

11. Browser



Ubah User

localhost/studi-kasus/user/ubah/3

PERPUSTAKAAN

Dashboard User Mahasiswa Buku Peminjaman Logout

Ubah User

Nama Lengkap *

Roronoa Zoro

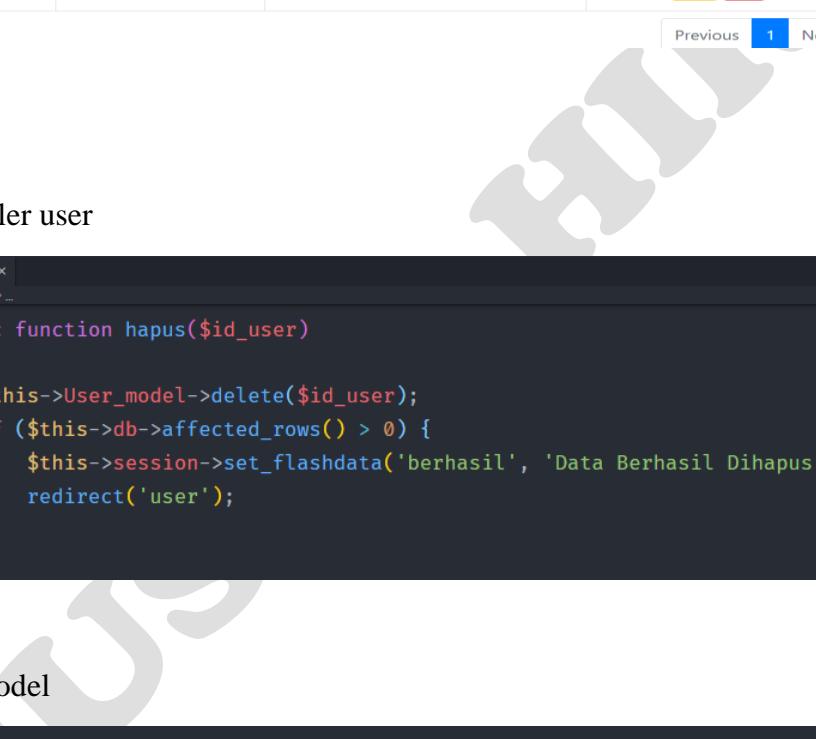
Level *

Operator

Kembali Ubah

Copyright © Tamus Tahir 2020

12. Pengujian ubah data



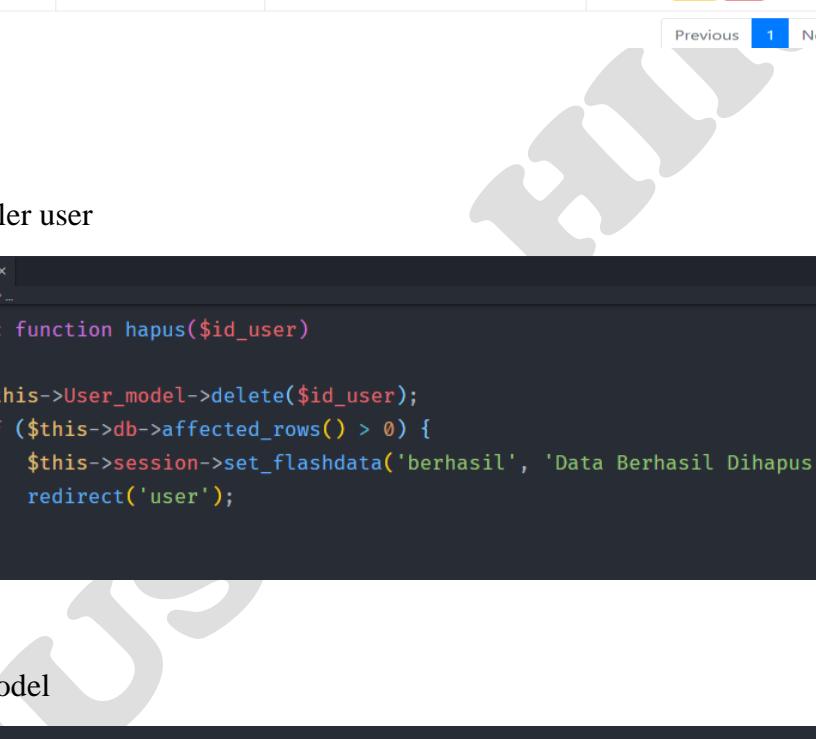
Data Berhasil Diubah					
Show 10 entries		Search:			
#	Username	Level	Nama	Aksi	
1	admin	Admin	Monkey D Luffy		
2	operator	Operator	Vinsmoke Sanji		

Showing 1 to 2 of 2 entries

Previous 1 Next

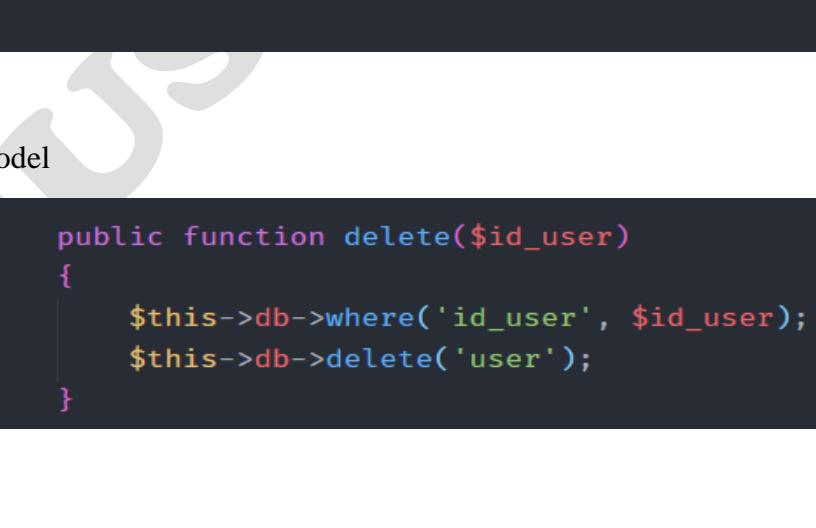
DELETE

1. Method hapus controller user



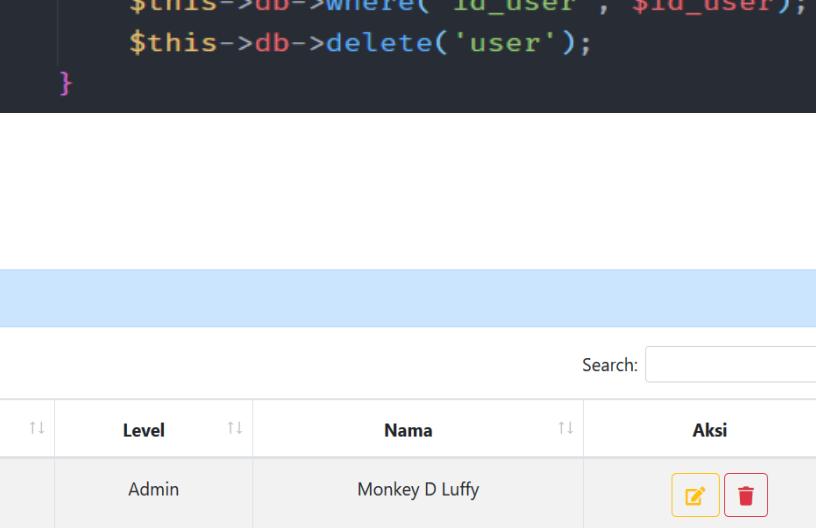
```
EXPLORER          User_model.php  User.php x
STUDI-KASUS      application > controllers > User.php > ...
application
  > cache
  > config
  > controllers
    > index.html
    > User.php
    > Welcome.php
  > core
  > helpers
  > hooks
  > language
  > libraries
  > logs
  > models
    > index.html
66   public function hapus($id_user)
67   {
68     $this->User_model->delete($id_user);
69     if ($this->db->affected_rows() > 0) {
70       $this->session->set_flashdata('berhasil', 'Data Berhasil Dihapus');
71       redirect('user');
72     }
73   }
```

2. Method delete user model



```
> language
> libraries
> logs
> models
  > index.html
  > User_model.php
  > third_party
> views
> errors
> templates
40   public function delete($id_user)
41   {
42     $this->db->where('id_user', $id_user);
43     $this->db->delete('user');
44   }
```

3. Pengujian



Data Berhasil Dihapus					
Show 10 entries		Search:			
#	Username	Level	Nama	Aksi	
1	admin	Admin	Monkey D Luffy		

LOGIN || LOGOUT

LOGIN

1. Controller Login ➔



The screenshot shows a code editor with a dark theme. The file is named 'Login.php' and is located in the 'controllers' folder of a CodeIgniter application. The code is as follows:

```
index.php Login.php X
application > controllers > Login.php > PHP Intelsphense > Login > index

1 <?php
2 defined('BASEPATH') or exit('No direct script access allowed');
3
4 class Login extends CI_Controller
5 {
6     public function index()
7     {
8         $this->form_validation->set_rules('username', 'username', 'required|trim');
9         $this->form_validation->set_rules('password', 'password', 'required|trim');
10        $this->form_validation->set_error_delimiters('<div class="invalid-feedback text-capitalize">', '</div>');
11
12        if ($this->form_validation->run() == false) {
13            $data['title'] = 'Login';
14            $this->load->view('login/index', $data);
15        } else {
16            $username = $this->input->post('username');
17            $password = $this->input->post('password');
18
19            /*
20             disini saya tidak membuat model karena kita hanya membutuhkan 1 kali pemanggilan database
21             kita bisa saja tidak membutuhkan model untuk mengambil data dari database
22             tapi sebaiknya sesuai dengan aturan codeigniter yaitu menggunakan model
23            */
24
25            // mengambil data user berdasarkan username yang diinput
26            $user = $this->db->query("SELECT * FROM user WHERE username = '$username'")->row();
27        }
    }
```

```

27
28         // jika data usernya ada
29         if ($user) {
30             // cek apakah passwordnya benar
31             if (password_verify($password, $user->password)) {
32                 // jika benar berikan session sesuai data user
33                 $data = [
34                     'id_user' => $user->id_user,
35                     'level' => $user->level
36                 ];
37                 $this->session->set_userdata($data);
38                 redirect('dashboard');
39             } else {
40                 // jika passwordnya salah
41                 $this->session->set_flashdata('gagal', 'Password Salah');
42                 redirect('login');
43             }
44         } else {
45             // jika data usernya tidak ada
46             $this->session->set_flashdata('gagal', 'Username Salah');
47             redirect('login');
48         }
49     }
50 }
51 }
```

2. View login/index.php

```

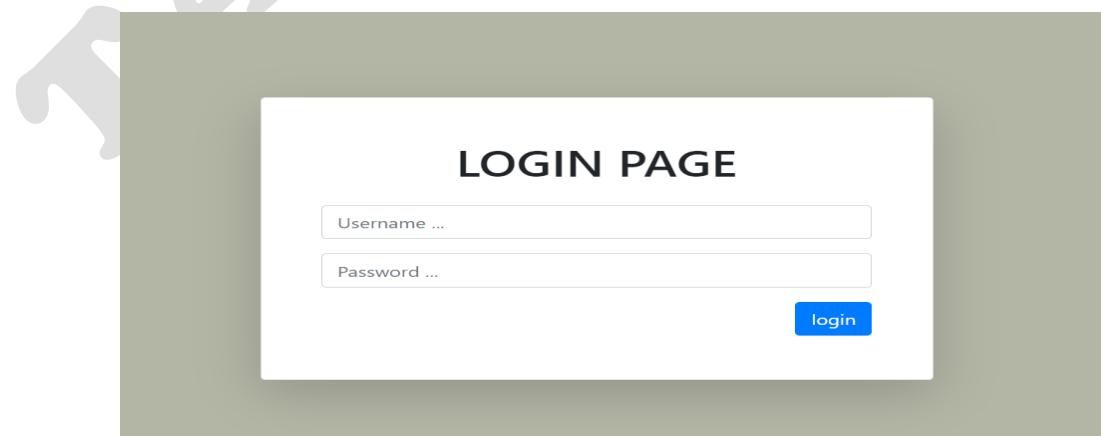
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title><?= $title; ?></title>
8
9     <link rel="stylesheet" href="<?= base_url('assets/bootstrap-4.4.1-dist/css/bootstrap.css'); ?>">
10
11    <style>
12        html,
13        body {
14            height: 100%;
15            margin: 0;
16            padding: 0;
17        }
18
19        body {
20            display: flex;
21            align-items: center;
22            justify-content: center;
23            background: #b3b5a5;
24        }
25    </style>
26
27 </head>
28
```

```

C:\xampp\views>login>index.php>html
29 <body>
30   <div class="container">
31     <div class="row justify-content-center">
32       <div class="col-md-6">
33         <div class="card p-5 shadow-lg ">
34
35           <h1 class="text-center mb-4">LOGIN PAGE</h1>
36
37           <?php if ($this->session->flashdata('gagal')) : ?>
38             <div class="alert alert-danger my-3" role="alert">
39               <?= $this->session->flashdata('gagal'); ?>
40             </div>
41           <?php endif ?>
42
43           <?= form_open(); ?>
44           <div class="form-group">
45             <input type="text" class="form-control <?= form_error('username') ? 'is-invalid' :
46               null; ?>" name="username" placeholder="Username ...">
47               <?= form_error('username'); ?>
48           </div>
49           <div class="form-group">
50             <input type="password" class="form-control <?= form_error('password') ? 'is-invalid' :
51               null; ?>" name="password" placeholder="Password ...">
52               <?= form_error('password'); ?>
53           </div>
54           <div class="text-right">
55             <button type="submit" class="btn btn-primary">login</button>
56           </div>
57         </div>
58       </div>
59     </div>
60   </body>
61 </html>

```

3. Browser ➔ <http://localhost/studi-kasus/login>



4. Pengujian
⇒ Jika form tidak diisi

LOGIN PAGE

Username ... !
The Username Field Is Required.

Password ... !
The Password Field Is Required.

login

- ⇒ Jika username salah

Username Salah

Username ...

- ⇒ Jika username benar dan password salah

Password Salah

Username ...

- ⇒ Jika username dan password benar maka akan diarahkan ke halaman dashboard
⇒ Ini akan error karena kita belum buat halaman dashboard

LOGOUT

1. Method logout controller login

```
EXPLORER          Login.php ×
STUDI-KASUS
  application > controllers > Login.php > ...
    public function logout()
    {
        $this->session->sess_destroy();
        redirect('login');
    }

```

DASHBOARD

1. Controller dashboard

The screenshot shows the code editor interface with the following details:

- Left Sidebar (EXPLORER):** Shows the project structure under "STUDI-KASUS".
 - application**: cache, config, controllers (Dashboard.php, index.html, Login.php, User.php, Welcome.php)
 - core, helpers, hooks, language, libraries, logs, models, third_party, views
 - .htaccess, index.html
 - assets, system, user_guide, .editorconfig, .gitignore, .htaccess
 - composer.json, contributing.md, index.php, license.txt, readme.rst
- Top Bar:** Shows the current file is "Dashboard.php" and the full path is "application > controllers > Dashboard.php".
- Code Editor Area:** Displays the PHP code for the Dashboard controller.

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Dashboard extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('User_model');
    }

    public function index()
    {
        $data['title'] = 'Dashboard';
        // ketika login akan diberikan session userdata id user
        $id_user = $this->session->userdata('id_user');
        // id user ini digunakan untuk mengambil data user
        $data['user'] = $this->User_model->getRow($id_user);

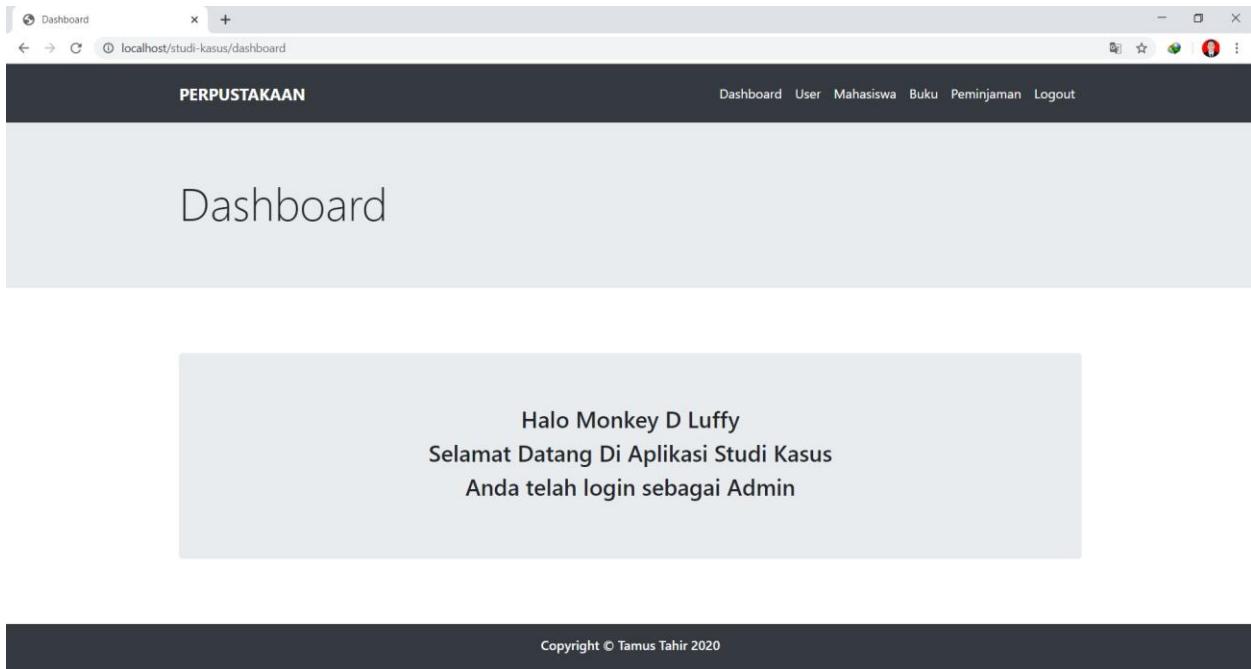
        $this->load->view('templates/header', $data);
        $this->load->view('dashboard/index', $data);
        $this->load->view('templates/footer', $data);
    }
}
```

2. View → bootstrap jumbotron

The screenshot shows a code editor interface with several tabs at the top: 'EXPLORER', 'Dashboard.php', and 'index.php x'. The 'index.php' tab is active, displaying PHP code. On the left, there's a file tree under the 'STUDI-KASUS' project. The 'views/dashboard/index.php' file is selected. The code itself is a simple HTML template with some PHP logic:

```
<div class="container py-5">
    <div class="jumbotron text-center">
        <h3>Halo <?= $user->nama; ?></h3>
        <h3>Selamat Datang Di Aplikasi Studi Kasus</h3>
        <h3>Anda telah login sebagai <?= $user->level; ?></h3>
    </div>
</div>
```

3. Browser



4. Ubah controller default ke dashboard

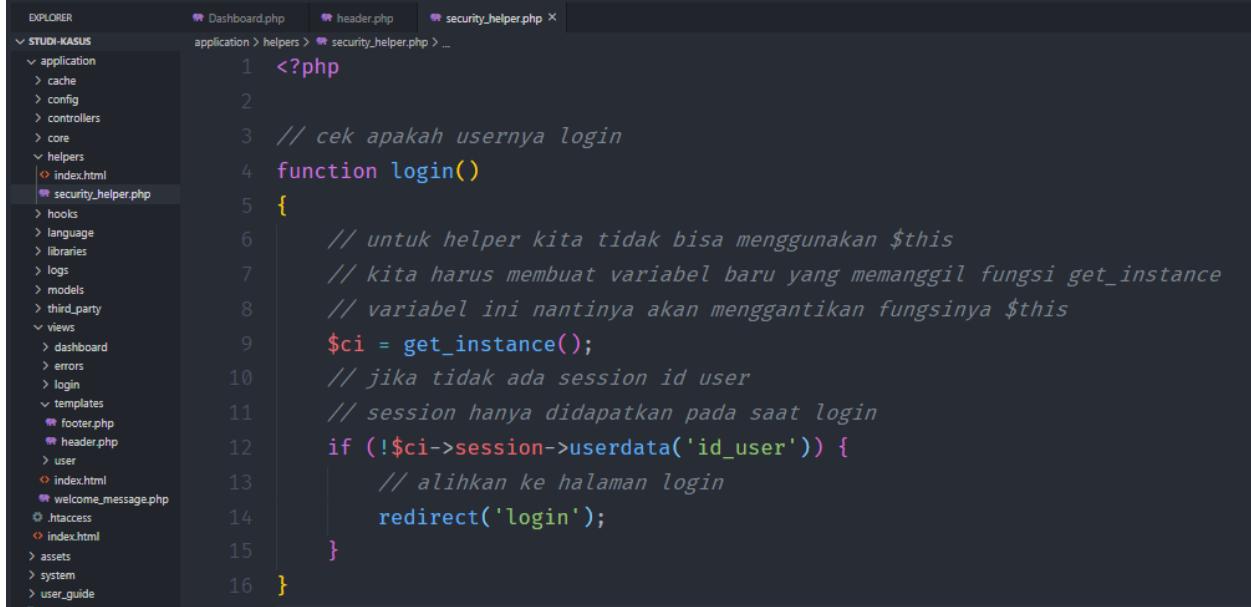
The screenshot shows a code editor with the following details:

- Explorer:** STUDI-KASUS
- File:** routes.php
- Content:** routes.php (application > config > routes.php)

```
46 / When you set this option to TRUE, it will replace ALL
47 / controller and method URI segments.
48 /
49 / Examples: my-controller/index -> my_controller/index
50 /           my-controller/my-method -> my_controller/my_met
51 */
52 $route['default_controller'] = 'dashboard';
53 $route['404_override'] = '';
54 $route['translate_uri_dashes'] = FALSE;
```

SECURITY

1. Buat helper baru dengan nama security_helper.php



```
1 <?php
2
3 // cek apakah usernya login
4 function login()
5 {
6     // untuk helper kita tidak bisa menggunakan $this
7     // kita harus membuat variabel baru yang memanggil fungsi get_instance
8     // variabel ini nantinya akan menggantikan fungsinya $this
9     $ci = get_instance();
10    // jika tidak ada session id user
11    // session hanya didapatkan pada saat login
12    if (!$ci->session->userdata('id_user')) {
13        // alihkan ke halaman login
14        redirect('login');
15    }
16 }
```

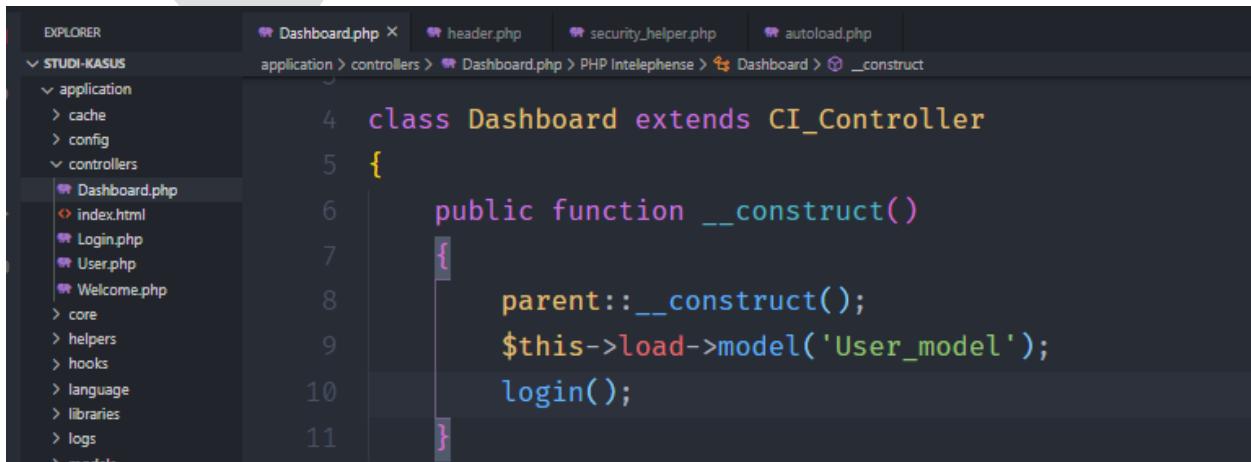
2. Load helper



```
91 */
92 $autoload['helper'] = array('url', 'security');
93
```

3. Panggil function tersebut pada controller yang akan kita periksa

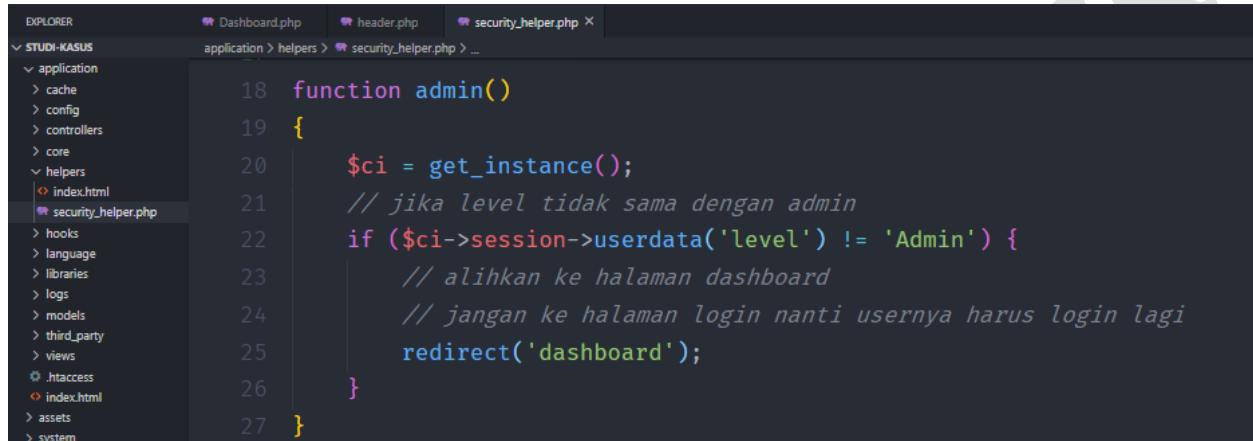
4. Controller dashboard



```
4 class Dashboard extends CI_Controller
5 {
6     public function __construct()
7     {
8         parent::__construct();
9         $this->load->model('User_model');
10        login();
11    }
}
```

5. Pengujian pada browser
 - ⇒ Jika anda sedang login, logout terlebih dahulu
 - ⇒ Akses browser: <http://localhost/studi-kasus/dashboard>
 - ⇒ Jika securitynya berhasil maka anda akan di alihkan ke halaman login

6. Untuk controller user hanya boleh diakses oleh user yang levelnya admin
7. Buat security untuk cek level pada helper

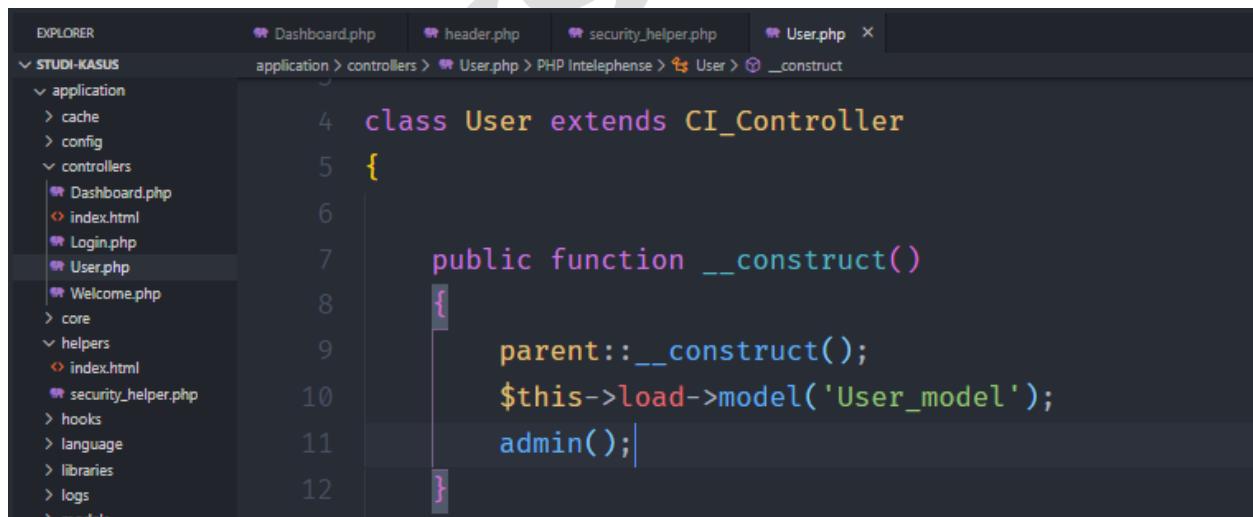


```

EXPLORER Dashboard.php header.php security_helper.php
STUDI-KASUS application helpers security_helper.php ...
18 function admin()
19 {
20     $ci = get_instance();
21     // jika level tidak sama dengan admin
22     if ($ci->session->userdata('level') != 'Admin') {
23         // alihkan ke halaman dashboard
24         // jangan ke halaman login nanti usernya harus login lagi
25         redirect('dashboard');
26     }
27 }

```

8. Panggil fungsinya pada controller user



```

EXPLORER Dashboard.php header.php security_helper.php User.php ...
STUDI-KASUS application controllers User.php ...
4 class User extends CI_Controller
5 {
6
7     public function __construct()
8     {
9         parent::__construct();
10        $this->load->model('User_model');
11        admin();
12    }

```

9. Pengujian pada browser
 - ⇒ Login dengan level operator dan akses halaman user
 - ⇒ Login dengan level admin dan akses halaman user

10. Karena halaman user tidak bisa diakses oleh operator, kita akan hilangkan pada nav jika login sebagai operator
11. Views / templates / header.php

```
> dashboard           27
> errors             28
> login              29
> mahasiswa          30
< templates          31
  < footer.php        32
  < header.php        33
    > user              34
      < index.html       35
      < welcome_message.php
      < .htaccess         36
      < index.html       37
< assets             38
  > bootstrap-4.4.1-dist
```

```
<div class="navbar-nav ml-auto">
  <a class="nav-item nav-link text-white" href="= base_url(); ?">>Dashboard</a>

  <?php if ($this->session->userdata('level') == 'Admin') : ?>
    <a class="nav-item nav-link text-white" href="= base_url('user'); ?">>User</a>
  <?php endif ?>

  <a class="nav-item nav-link text-white" href="= base_url('mahasiswa'); ?">>Mahasiswa</a>
```

12. Browser



CRUD MAHASISWA

DATABASE

- Buat tabel baru dengan nama mahasiswa dengan struktur sebagai berikut

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_mahasiswa	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama	varchar(128)	utf8mb4_general_ci		No	None			Change Drop More
3	nim	int(11)			No	None			Change Drop More

CREATE || READ

- Controller mahasiswa

The screenshot shows a code editor interface with a dark theme. On the left, there's an 'EXPLORER' sidebar showing a file structure for a 'STUDI-KASUS' project. The 'models' folder is currently selected. In the main area, a file named 'Mahasiswa.php' is open, showing PHP code for a controller:

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Mahasiswa extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('Mahasiswa_model');
        login();
    }

    public function index()
    {
        $data['title'] = 'Data Mahasiswa';
        $data['mahasiswa'] = $this->Mahasiswa_model->get();

        $this->load->view('templates/header', $data);
        $this->load->view('mahasiswa/index', $data);
        $this->load->view('templates/footer', $data);
    }
}
```



EXPLORER Mahasiswa.php

application > controllers > Mahasiswa.php > PHP Intelephense > Mahasiswa > index

```
24     public function tambah()
25     {
26
27         $this->form_validation->set_rules('nama', 'nama', 'required|trim');
28         $this->form_validation->set_rules('nim', 'nim', 'required|trim');
29         $this->form_validation->set_error_delimiters('<div class="invalid-feedback text-capitalize">', '</div>');
30
31         if ($this->form_validation->run() == FALSE) {
32             $data['title'] = 'Tambah Mahasiswa';
33             $this->load->view('templates/header', $data);
34             $this->load->view('mahasiswa/tambah', $data);
35             $this->load->view('templates/footer', $data);
36         } else {
37             $this->Mahasiswa_model->insert();
38             $this->session->set_flashdata('berhasil', 'Data Berhasil Ditambahkan');
39             redirect('mahasiswa');
40         }
41     }
42 }
```

2. Mahasiswa model



EXPLORER Mahasiswa_model.php

application > models > Mahasiswa_model.php > PHP Intelephense > Mahasiswa_model > get

```
1 <?php
2 defined('BASEPATH') or exit('No direct script access allowed');
3
4 class Mahasiswa_model extends CI_Model
5 {
5
6     public function get()
7     {
8         return $this->db->get('mahasiswa')->result();
9     }
10
11    public function insert()
12    {
13
14        $data = [
15            'nama' => $this->input->post('nama'),
16            'nim' => $this->input->post('nim')
17        ];
18        $this->db->insert('mahasiswa', $data);
19    }
20 }
```

3. View mahasiswa index

The screenshot shows a code editor interface with two tabs open: 'Mahasiswa_model.php' and 'index.php'. The left sidebar displays a file tree for a project named 'STUDI-KASUS' under 'application'. Both tabs show PHP code.

Mahasiswa_model.php:

```
1 <div class="container py-3">
2
3     <a class="btn btn-primary" href="= base_url('mahasiswa/tambah'); ?" role="button">Tambah</a>
4
5     <?php if ($this->session->flashdata('berhasil')) : ?>
6         <div class="alert alert-primary my-3" role="alert">
7             <?= $this->session->flashdata('berhasil'); ?>
8         </div>
9     <?php endif ?>
10
11    <div class="table-responsive my-3">
12        <table class="table table-striped table-bordered text-center" id="my-table">
13            <thead>
14                <tr>
15                    <th scope="col">#</th>
16                    <th scope="col">Nama</th>
17                    <th scope="col">NIM</th>
18                    <th scope="col">Aksi</th>
19                </tr>
20            </thead>
```

index.php:

```
21    <tbody>
22        <?php $i = 1; ?>
23        <?php foreach ($mahasiswa as $u) : ?>
24            <tr>
25                <th scope="row"><?= $i; ?></th>
26                <td><?= $u->nama; ?></td>
27                <td><?= $u->nim; ?></td>
28                <td>
29                    <a class="btn btn-outline-warning" href="= base_url('mahasiswa/ubah/'
30                        . $u-&gt;id_mahasiswa); ?" role="button">
31                        <i class="fa fa-edit"></i>
32                    </a>
33                    <a class="btn btn-outline-danger" href="= base_url('mahasiswa/hapus/'
34                        . $u-&gt;id_mahasiswa); ?" onclick="return confirm('Anda Yakin?')"
35                        role="button">
36                        <i class="fa fa-trash"></i>
37                    </a>
38                </td>
39            </tr>
40        <?php endforeach; ?>
41    </tbody>
42    </table>
43 </div>
```

4. Browser

The screenshot shows a web application interface. At the top, there are several tabs: 'localhost / 127.0.0.1 / perpus', 'Data Mahasiswa', 'Jumbotron - Bootstrap', and 'Font Awesome'. The main content area has a dark header with the text 'PERPUSTAKAAN'. Below the header is a navigation menu with links: 'Dashboard', 'User', 'Mahasiswa', 'Buku', 'Peminjaman', and 'Logout'. The main content area is titled 'Data Mahasiswa'. It contains a table with four columns: '#', 'Nama', 'NIM', and 'Aksi'. The table is currently empty, displaying the message 'No data available in table'. Above the table, there is a search bar labeled 'Search:' and a button labeled 'Tambah'. Below the table, there are buttons for 'Previous' and 'Next'. At the bottom of the page, there is a footer with the text 'Copyright © Tamus Tahir 2020'.

5. View mahasiswa tambah

```
EXPLORER
STUDI-KASUS
  application
    > cache
    > config
    > controllers
    > core
    > helpers
    > hooks
    > language
    > libraries
    > logs
    > models
    > third_party
  views
    > dashboard
    > errors
    > login
    > mahasiswa
      index.php
      tambah.php
    > templates
    > user
      index.html
      welcome_message.php
      .htaccess
      index.html
    > assets
    > system
    > user_guide
    > .editorconfig
    > .gitignore
    > .travis.yml
    composer.json
    contributing.md
    index.php
    license.txt
    readme.rst

tambah.php
application > views > mahasiswa > tambah.php > div.container.mt-3 mb-5 > div.form-group
1  <div class="container mt-3 mb-5">
2
3    <?= form_open(); ?>
4
5    <div class="form-group">
6      <label for="nama">Nama Lengkap <span class="text-danger"></span></label>
7      <input type="text" class="form-control <?= form_error('nama') ? 'is-invalid' : null; ?>" id="nama" name="nama" autocomplete="off">
8        <?= form_error('nama'); ?>
9    </div>
10
11   <div class="form-group">
12     <label for="nim">NIM <span class="text-danger"></span></label>
13     <input type="text" class="form-control <?= form_error('nim') ? 'is-invalid' : null; ?>" id="nim" name="nim" autocomplete="off">
14       <?= form_error('nim'); ?>
15   </div>
16
17   <a class="mr-2 mt-3 btn btn-warning" href="<?= base_url('mahasiswa'); ?>" role="button">
18     Kembali</a>
19   <button class="mt-3 btn btn-primary" type="submit">Tambah </button>
20
21 </div>
```

6. Browser

Tambah Mahasiswa

Nama Lengkap *

NIM *

Kembali Tambah

Copyright © Tamus Tahir 2020

7. Pengujian ➔ tambahkanlah beberapa data (coba lebih dari 10 supaya bisa melihat pagination dari datatables)

Data Mahasiswa

Tambah

Data Berhasil Ditambahkan

#	Nama	NIM	Aksi
1	Eren Yeager	2147483647	

Show 10 entries Search:

Showing 1 to 1 of 1 entries

Previous 1 Next

LANJUTKAN UNTUK
UBAH DAN HAPUS

CRUD BUKU

DATABASE

- Buat tabel dengan nama buku, dengan struktur table sebagai berikut:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_buku 🔑	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	judul	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	deskripsi	mediumtext	utf8mb4_general_ci		No	None			Change Drop More
4	jumlah	int(11)			No	None			Change Drop More
5	gambar	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

CREATE || READ

- Controller buku

The screenshot shows a code editor window with the file `Buku.php` open. The left sidebar displays the project structure under `STUDI-KASUS`, showing files like `Buku.php`, `Dashboard.php`, `index.html`, `Login.php`, `Mahasiswa.php`, `User.php`, and `Welcome.php`. The right pane contains the PHP code for the `Buku` controller:

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Buku extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('Buku_model');
        login();
    }

    public function index()
    {
        $data['title'] = 'Data Buku';
        $data['buku'] = $this->Buku_model->get();

        $this->load->view('templates/header', $data);
        $this->load->view('buku/index', $data);
        $this->load->view('templates/footer', $data);
    }
}
```



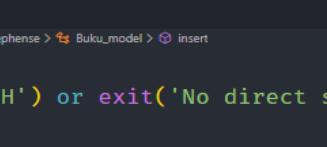
STUDI-KASUS

Buku.php

```
application > controllers > Buku.php > PHP Intelephense > Buku > tambah
```

```
24     public function tambah()
25     {
26         $this->form_validation->set_rules('judul', 'judul', 'required|trim');
27         $this->form_validation->set_rules('deskripsi', 'deskripsi', 'required|trim');
28         $this->form_validation->set_rules('jumlah', 'jumlah', 'required');
29         $this->form_validation->set_error_delimiters('<div class="invalid-feedback">
30             text-capitalize">', '</div>');
31
32         if ($this->form_validation->run() == FALSE) {
33             $data['title'] = 'Tambah buku';
34             $this->load->view('templates/header', $data);
35             $this->load->view('buku/tambah', $data);
36             $this->load->view('templates/footer', $data);
37         } else {
38             $this->Buku_model->insert();
39             $this->session->set_flashdata('berhasil', 'Data Berhasil Ditambahkan');
40             redirect('buku');
41         }
42     }
```

2. Model buku



STUDI-KASUS

Buku.php

Buku_model.php

```
application > models > Buku_model.php > PHP Intelephense > Buku_model > insert
```

```
1  <?php
2  defined('BASEPATH') or exit('No direct script access allowed');
3
4  class Buku_model extends CI_Model
5  {
6
7      public function get()
8      {
9          return $this->db->get('buku')->result();
10     }
11
12     public function insert()
13     {
14         $url = 'buku/tambah';
15         $data = [
16             'judul' => $this->input->post('judul'),
17             'deskripsi' => $this->input->post('deskripsi'),
18             'jumlah' => $this->input->post('jumlah'),
19             'gambar' => $this->upload($url)
20         ];
21         $this->db->insert('buku', $data);
22     }
23 }
```

STUDI-KASUS

```

        public function upload($url)
        {
            $config['upload_path']          = './uploads/';
            $config['allowed_types']        = 'jpeg|jpg|png';
            $config['max_size']             = 500;
            $config['encrypt_name']         = TRUE;

            $this->load->library('upload', $config);

            if (!$this->upload->do_upload('gambar')) {
                $error = array('error' => $this->upload->display_errors());
                $this->session->set_flashdata('error', $error['error']);
                redirect($url);
            } else {
                $data = $this->upload->data();
                return $data['file_name'];
            }
        }
    }
}

```

3. View buku index

- ⇒ Untuk gambar dan deskripsi tidak kita tampilkan dalam table
- ⇒ Nantinya kita akan membuat halaman detail

STUDI-KASUS

```

        <div class="container py-3">

            <a class="btn btn-primary" href="= base_url('buku/tambah'); ?" role="button">Tambah</a>

            <?php if ($this->session->flashdata('berhasil')) : ?>
                <div class="alert alert-primary my-3" role="alert">
                    <?= $this->session->flashdata('berhasil'); ?>
                </div>
            <?php endif ?>

            <div class="table-responsive my-3">
                <table class="table table-striped table-bordered text-center" id="my-table">
                    <thead>
                        <tr>
                            <th scope="col">#</th>
                            <th scope="col">Judul</th>
                            <th scope="col">Jumlah</th>
                            <th scope="col">Aksi</th>
                        </tr>
                    </thead>

```

STUDI-KASUS

```

EXPLORER Buku.php Buku_model.php index.php X
application > views > buku > index.php > div.container py-3

21   <tbody>
22       <?php $i = 1; ?>
23       <?php foreach ($buku as $u) : ?>
24           <tr>
25               <th scope="row"><?= $i; ?></th>
26               <td><?= $u->judul; ?></td>
27               <td><?= $u->jumlah; ?></td>
28               <td>
29                   <a class="btn btn-outline-warning" href="= base_url('buku/ubah/' . $u-&gt;id_buku); ?&gt;" role="button"&gt;
30                       &lt;i class="fa fa-edit"&gt;&lt;/i&gt;
31                   &lt;/a&gt;
32                   &lt;a class="btn btn-outline-danger" href="<?= base_url('buku/hapus/' . $u-&gt;id_buku); ?&gt;" onclick="return confirm('Anda Yakin?')" role="button"&gt;
33                       &lt;i class="fa fa-trash"&gt;&lt;/i&gt;
34                   &lt;/a&gt;
35               &lt;/td&gt;
36           &lt;/tr&gt;
37           &lt;?php $i++ ?&gt;
38       &lt;?php endforeach; ?&gt;
39   &lt;/tbody&gt;
40 &lt;/table&gt;
41 &lt;/div&gt;
42
43 &lt;/div&gt;
</pre

```

4. View buku tambah

STUDI-KASUS

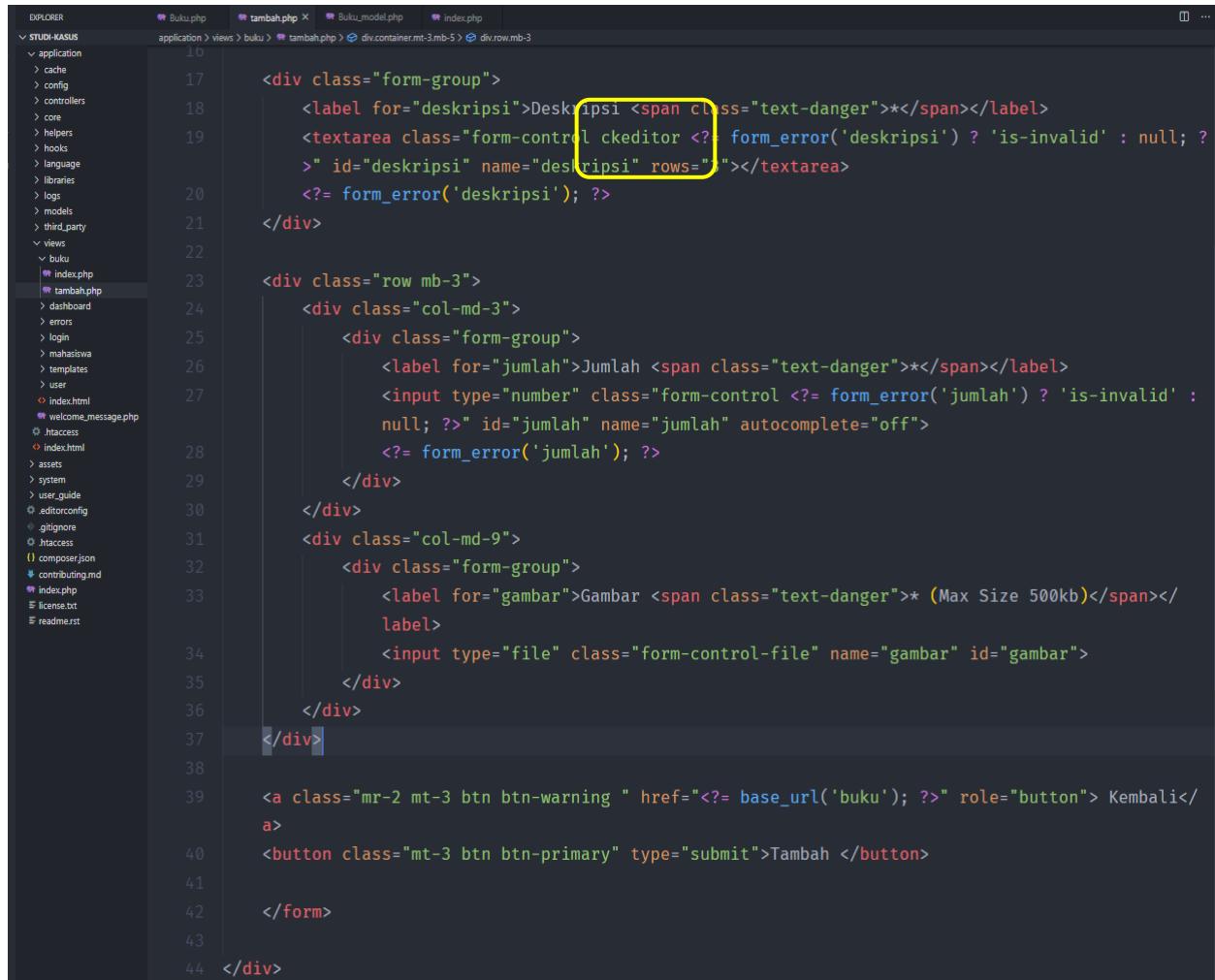
```

EXPLORER Buku.php Buku_model.php tambah.php X index.php ...
application > views > buku > tambah.php > div.container mt-3 mb-5

1  <div class="container mt-3 mb-5">
2
3      <?php if ($this->session->flashdata('error')) : ?>
4          <div class="alert alert-danger my-3" role="alert">
5              <?= $this->session->flashdata('error'); ?>
6          </div>
7      <?php endif ?>
8
9      <?= form_open_multipart(); ?>
10
11     <div class="form-group">
12         <label for="judul">Judul <span class="text-danger">*</span></label>
13         <input type="text" class="form-control" <?= form_error('judul') ? 'is-invalid' : null; ?>
14         id="judul" name="judul" autocomplete="off">
15         <?= form_error('judul'); ?>
16     </div>

```

- ⇒ Untuk menggunakan library ckeditor tambahkan classnya pada text area
- ⇒ Ckeditor hanya bisa digunakan pada text area



```

EXPLORER
Buku.php tambah.php Buku_model.php index.php
STUDI-KASUS application
  > cache
  > config
  > controllers
  > core
  > helpers
  > hooks
  > language
  > libraries
  > logs
  > models
  > third_party
  > views
    > buku
      index.php
      tambah.php
        dashboard
        errors
        login
        mahasiswa
        templates
        user
        index.html
        .htaccess
        index.html
        assets
        system
        user_guide
        editorconfig
        .gitignore
        .htaccess
        composer.json
        contributing.md
        index.php
        license.txt
        readme.rst

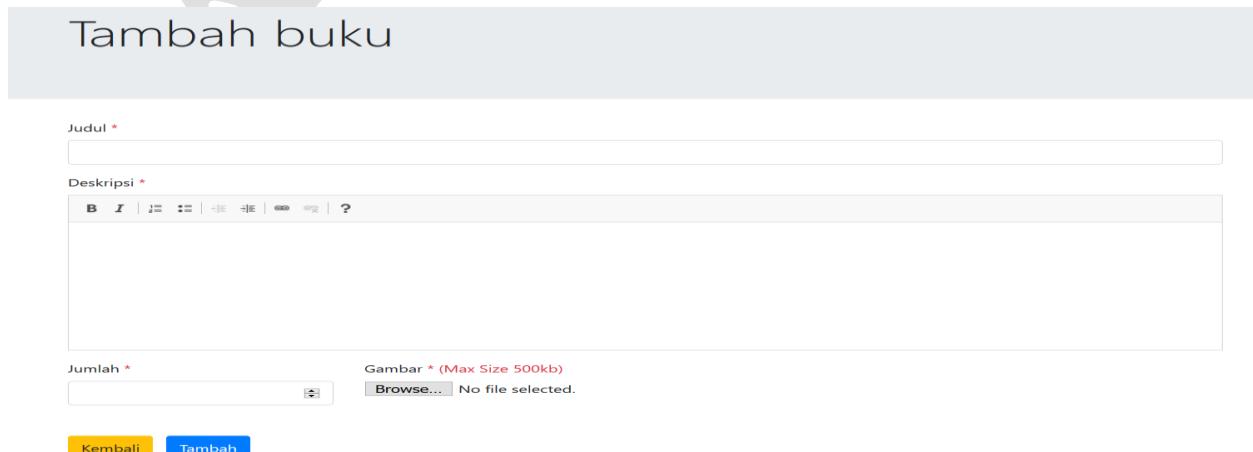
10
17 <div class="form-group">
18   <label for="deskripsi">Deskripsi <span class="text-danger">*</span></label>
19   <textarea class="form-control ckeditor" <?= form_error('deskripsi') ? 'is-invalid' : null; ?>
20     id="deskripsi" name="deskripsi" rows="3"></textarea>
21   <?= form_error('deskripsi'); ?>
22 </div>

23 <div class="row mb-3">
24   <div class="col-md-3">
25     <div class="form-group">
26       <label for="jumlah">Jumlah <span class="text-danger">*</span></label>
27       <input type="number" class="form-control" <?= form_error('jumlah') ? 'is-invalid' : null; ?>
28         id="jumlah" name="jumlah" autocomplete="off">
29       <?= form_error('jumlah'); ?>
30     </div>
31   <div class="col-md-9">
32     <div class="form-group">
33       <label for="gambar">Gambar <span class="text-danger">*( Max Size 500kb)</span></label>
34       <input type="file" class="form-control-file" name="gambar" id="gambar">
35     </div>
36   </div>
37 </div>

38
39 <a class="mr-2 mt-3 btn btn-warning" href="= base_url('buku'); ?&gt;" role="button"&gt; Kembali &lt;/a&gt;
40 &lt;button class="mt-3 btn btn-primary" type="submit"&gt;Tambah &lt;/button&gt;
41
42 &lt;/form&gt;
43
44 &lt;/div&gt;
</pre

```

5. Browser



Tambah buku

Judul *

Deskripsi *



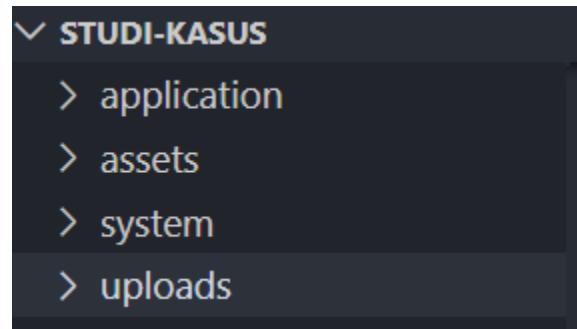
Jumlah *

Gambar * (Max Size 500kb)

 Browse... No file selected.

Kembali **Tambah**

6. Buat folder uploads pada root folder



7. Pengujian

- ⇒ Jika form kosong
- ⇒ Jika tidak ada foto diupload
- ⇒ Jika type file tidak diizinkan
- ⇒ Jika size file melebihi max size

8. Jika berhasil ➔ tambahkanlah beberapa data

Data Buku

Tambah

Data Berhasil Ditambahkan

Show 10 entries		Search:
#	Judul	Jumlah
1	One Piece Arc Wano	5

Showing 1 to 1 of 1 entries

Previous 1 Next

DETAIL

1. Search modal pada bootstrap

The screenshot shows a browser window with four tabs: 'localhost / 127.0.0.1 / perpus...', 'Data Buku', 'Modal - Bootstrap', and 'Font Awesome'. The main content area displays the 'Live demo' section for the Modal component. It includes a search bar, a sidebar with navigation links, and a central panel with a button labeled 'Launch demo modal'. Below the button is a code snippet:

```
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">
  Launch demo modal
</button>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel">
```

A 'Copy' button is located to the right of the code snippet. To the right of the main content area is a sidebar with links to other documentation sections.

2. Copy button → view / buku / index

The screenshot shows a code editor with a sidebar containing project files like 'STUDI-KASUS', 'application', 'views', 'buku', 'index.php', 'tambah.php', and 'dashboard'. The main pane shows the 'index.php' file with the following code:

```
<th scope="row"><?= $i; ?></th>
<td><?= $u->judul; ?></td>
<td><?= $u->jumlah; ?></td>
<td>
  <button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">
    Launch demo modal
  </button>
  <a class="btn btn-outline-warning" href="<?= base_url('buku/ubah/' .
```

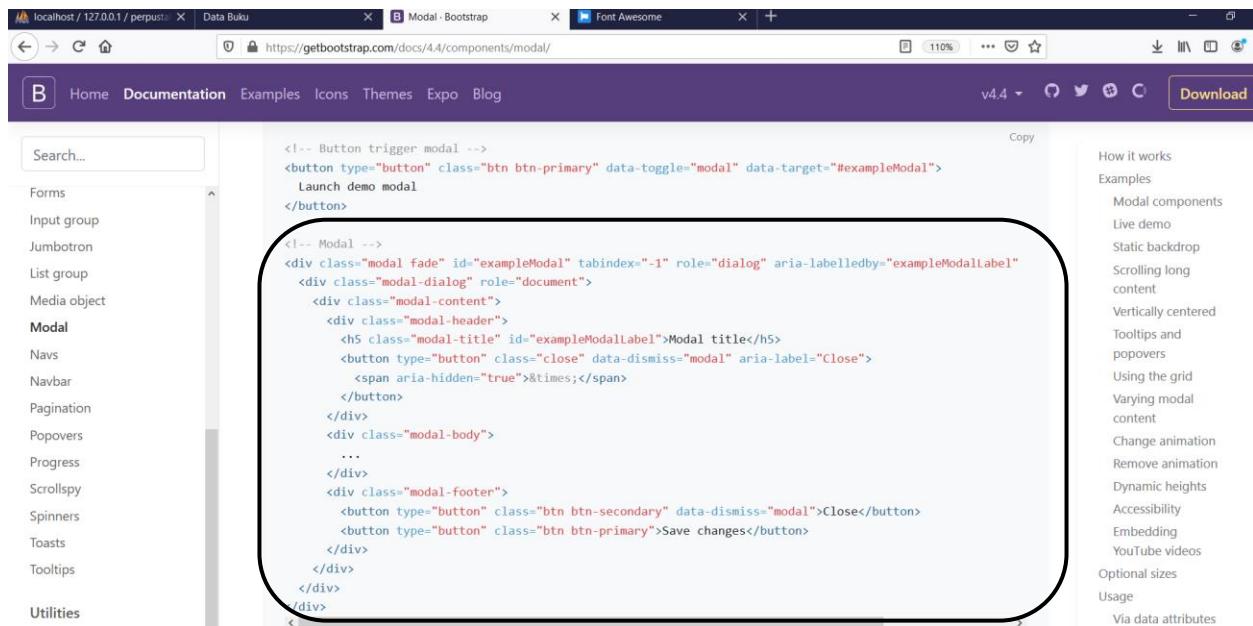
3. Ubah button biasa menjadi outline

4. Ganti tulisannya menggunakan icon → search pada font aweseome → eye

The screenshot shows the same code editor with the 'index.php' file open. The button has been modified to use the 'fa-eye' icon from Font Awesome:

```
<td><?= $u->jumlah; ?></td>
<td>
  <button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">
    <i class="fa fa-eye"></i>
  </button>
  <a class="btn btn-outline-warning" href="<?= base_url('buku/ubah/' .
```

5. Copy isi modal dari bootstrap ke index ke footer sebelum script

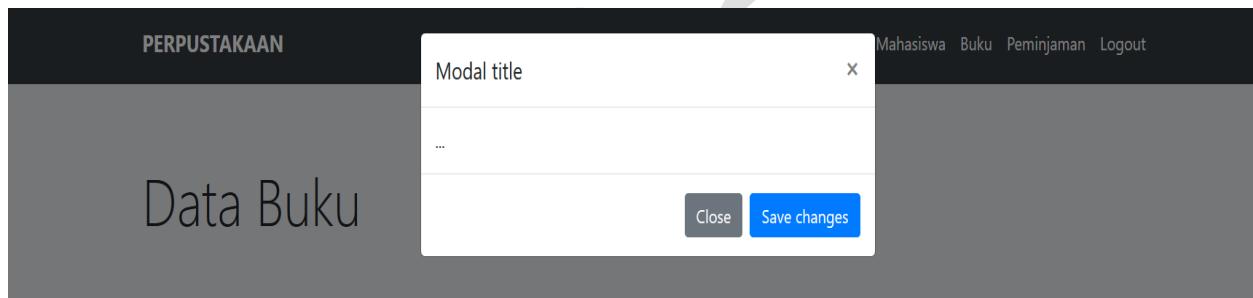


The screenshot shows a browser window with four tabs: 'localhost / 127.0.0.1 / perpus...', 'Data Buku', 'B Modal - Bootstrap', and 'Font Awesome'. The main content area displays the Bootstrap documentation for 'Modal' components. On the left, there's a sidebar with links like 'Forms', 'Input group', 'Jumbotron', etc. The central content area contains the HTML code for a modal, which is circled in black. To the right of the code, there's a sidebar with links such as 'How it works', 'Examples', 'Modal components', 'Live demo', etc.

```
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">
  Launch demo modal
</button>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        ...
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary" data-id="<?=> $u->id_buku; ?>">Save changes</button>
      </div>
    </div>
  </div>
</div>
```

6. Test pada browser ➔ klik icon detail



7. Kirim parameter id buku pada button modal

8. Tambahkan class baru yang nantinya digunakan pada jquery untuk mengambil id yang dikirim



The screenshot shows a code editor with a sidebar on the left listing files and line numbers. The main area displays PHP code. On line 28, there is a line of code: '<td><?= \$u->jumlah; ?></td>'. On line 29, there is a line starting with '<td>'. On line 30, there is a line of code: '<button type="button" class="btn btn-outline-primary get-detail-buku" data-toggle="modal" data-target="#exampleModal" data-id="<?= \$u->id_buku; ?>">'. On line 31, there is a line of code: '<i class="fa fa-eye"></i>'. The 'get-detail-buku' class is highlighted in green.

9. Ambil id buku dengan menggunakan jquery di footer dan load sebuah halaman



```
EXPLORER      Buku.php  Buku_model.php  index.php  footer.php X
STUDI-KASUS   application > views > templates > footer.php > ...
application    37  <script>
                 // ketika halamannya sudah ready (selesai diload semua file) jalankan fungsi berikut
                 $(document).ready(function() {
                     // jQuery seleksi elemen yang idnya my-table jalankan function datatable
                     $('#my-table').DataTable();
                     // jQuery seleksi elemen yang idnya my-table
                     // pada saat di klik class get-detail-buku jalankan fungsi berikut
                     $('#my-table').on('click', '.get-detail-buku', function() {
                         // jQuery ambil data id pada tombol ini dan simpan dalam variabel id
                         const id = $(this).data('id');
                         // jQuery seleksi elemen yang classnya detail-buku
                         // load url berikut
                         $(".detail-buku").load('<?= base_url('buku/detail/'); ?>' + id);
                     });
                 });
             </script>
```

10. Method detail controller user



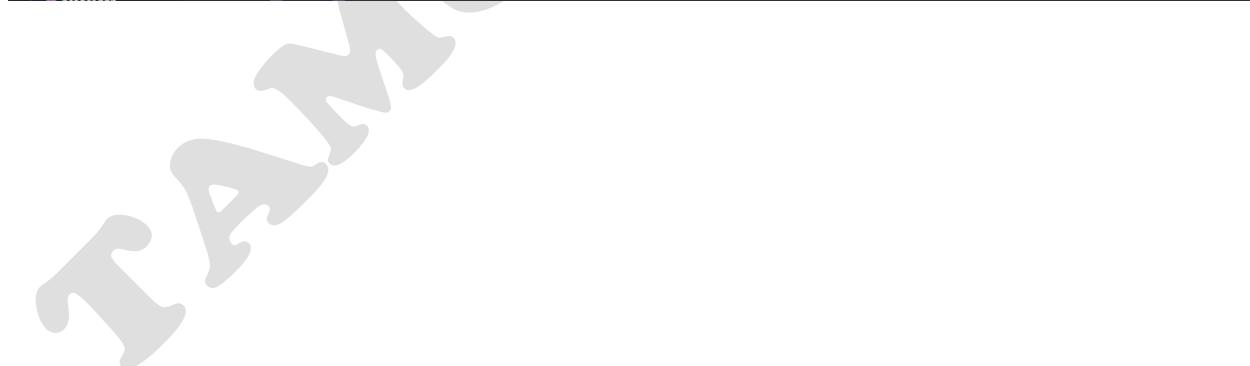
```
EXPLORER      Buku.php X  User.php  Buku_model.php  index.php  footer.php
STUDI-KASUS   application > controllers > Buku.php > ...
application    43  function detail()
                 {
                     // uri->segment(1) ==> controller
                     // uri->segment(2) ==> method
                     // uri->segment(3) ==> parameter
                     // dalam hal ini id yang dikirim dari footer
                     $id_buku = $this->uri->segment(3);
                     $data['detailbuku'] = $this->Buku_model->getRow($id_buku);
                     $this->load->view('buku/detail', $data);
                 }
             }
```

11. Method getRow model buku



```
> libraries  
> logs  
  < models  
    < Buku_model.php  
      42  
    < index.html  
    < Mahasiswa_model.php  
    < User_model.php  
  > third_party  
  < views  
    > buku  
    > dashboard  
    > errors  
      43  
      44  
      45  
      46  
      47  
      public function getRow($id_buku)  
      {  
        $where = [ 'id_buku' => $id_buku];  
        return $this->db->get_where('buku', $where)->row();  
      }
```

12. View buku detail



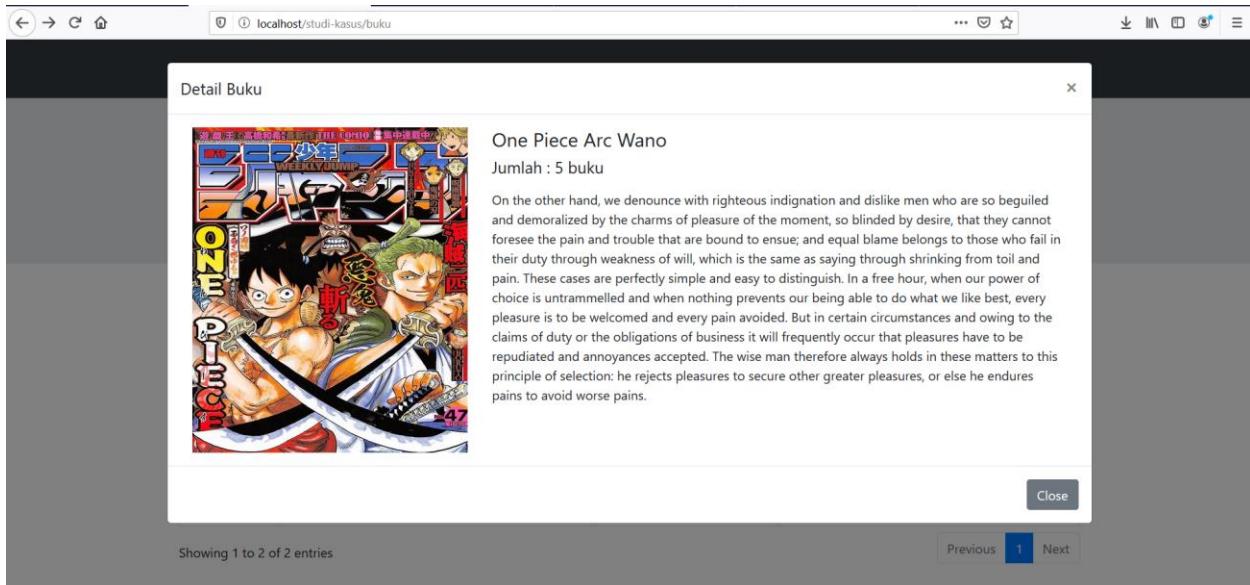
```
EXPLORER          Buku.php  Buku_model.php  detail.php X  index.php  footer.php  ...  
STUDI-KASUS       application > views > buku > detail.php > div.container  
  < application  
    > cache  
    > config  
    > controllers  
    > core  
    > helpers  
    > hooks  
    > language  
    > libraries  
    > logs  
    < models  
      1   <div class="container">  
      2     <div class="row">  
      3       <div class="col-md-4">  
      4         <img src=<?= base_url('uploads/' . $detailbuku->gambar); ?>  
      5           alt="" height="400" class="w-100">  
      6         </div>  
      7       <div class="col-md-8">  
      8         <h4><?= $detailbuku->judul; ?></h4>  
      9         <h5>Jumlah : <?= $detailbuku->jumlah; ?> buku</h5>  
     10        <p><?= $detailbuku->deskripsi; ?></p>  
     11      </div>  
     12    </div>
```

13. Tampilkan datanya pada modal footer

- ⇒ Ubah ukuran modal → tambahkan class modal-xl → ukuran paling besar
- ⇒ Ganti judul modal
- ⇒ Buat class baru detail-buku → tempat load halaman detail
- ⇒ Delete tombil save

```
9 <!-- modal detail buku -->
10 <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog"
aria-labelledby="exampleModalLabel" aria-hidden="true">
11     <div class="modal-dialog modal-xl" role="document">
12         <div class="modal-content">
13             <div class="modal-header">
14                 <h5 class="modal-title" id="exampleModalLabel">Detail Buku</h5>
15                 <button type="button" class="close" data-dismiss="modal"
aria-label="Close">
16                     <span aria-hidden="true">&times;</span>
17                 </button>
18             </div>
19             <div class="modal-body detail-buku">
20                 ...
21             </div>
22             <div class="modal-footer">
23                 <button type="button" class="btn btn-secondary"
data-dismiss="modal">Close</button>
24             </div>
25         </div>
26     </div>
27 </div>
28 <!-- end modal detail buku -->
```

14. Browser ➔ klik icon detail
15. Isi halaman detail ini akan berganti tanpa kita perlu merefresh halaman sesuai dengan id yang dikirim dari tombol detail



LANJUTKAN UNTUK
UBAH DAN HAPUS

CRUD PEMINJAMAN

DATABASE

- Buat table peminjaman dengan struktur:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_peminjaman	int(11)			No	None		AUTO_INCREMENT	Change Drop ▾ More
2	id_buku	int(11)			No	None			Change Drop ▾ More
3	id_mahasiswa	int(11)			No	None			Change Drop ▾ More
4	tanggal_peminjaman	date			No	None			Change Drop ▾ More
5	tanggal_pengembalian	date			No	None			Change Drop ▾ More
6	status	varchar(15)	utf8mb4_general_ci		No	None			Change Drop ▾ More

- Membuat trigger pada database, merupakan kode prosedural yang secara otomatis dijalankan untuk menanggapi perubahan tertentu pada table tertentu atau tampilan dalam database. Ketika table yang set trigger terjadi perubahan (insert/update/delete) akan mempengaruhi table yang di setting
- Trigger disini digunakan untuk mengurangi jumlah buku ketika terjadi peminjaman, pengembalian atau hapus data pada table peminjaman
- Table peminjamaan → pilih trigger

The screenshot shows the MySQL Workbench interface with the 'Triggers' tab selected. The table structure is identical to the one shown in the previous screenshot, with columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, Extra, and Action.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_peminjaman	int(11)			No	None		AUTO_INCREMENT	Change Drop ▾ More
2	id_buku	int(11)			No	None			Change Drop ▾ More
3	id_mahasiswa	int(11)			No	None			Change Drop ▾ More
4	tanggal_peminjaman	date			No	None			Change Drop ▾ More
5	tanggal_pengembalian	date			No	None			Change Drop ▾ More
6	status	varchar(10)	utf8mb4_general_ci		No	None			Change Drop ▾ More

5. Pilih add trigger



6. Set trigger insert → pilih go

⇒ Pada table peminjaman setelah insert data, update buku set jumlah = jumlah – 1

⇒ Dimana id_buku = id_buku pada data table peminjaman

7. Ketika ada peminjaman maka jumlah buku akan berkurang 1

The screenshot shows the 'Details' tab for creating a trigger. The trigger is named 'insert' for the table 'peminjaman'. It is defined as an AFTER INSERT trigger. The definition of the trigger is:

```
1 BEGIN
2     UPDATE buku SET jumlah = jumlah - 1
3     WHERE id_buku = NEW.id_buku;
4 END
```

The 'Definer' field is set to 'root@localhost'.

8. Set trigger update → pilih go
 - ⇒ Pada table peminjaman setelah update data
 - update buku set jumlah = jumlah + 1 (kembalikan jumlah buku lama yang dipinjam)
 - update buku set jumlah = jumlah - 1 (kurangi jumlah buku baru yang dipinjam)
 - ⇒ Dimana id buku = id buku pada data table peminjaman
9. Ketika ada pengembalian maka jumlah buku akan bertambah 1

Trigger name: update

Table: peminjaman

Time: AFTER

Event: UPDATE

```

1 BEGIN
2   UPDATE buku SET jumlah = jumlah + 1
3   WHERE id_buku = OLD.id_buku;
4
5   UPDATE buku SET jumlah = jumlah - 1
6   WHERE id_buku = NEW.id_buku;
7 END
  
```

Definer: root@localhost

10. Set trigger delete → pilih go
 - ⇒ Pada table peminjaman setelah delete data update buku set jumlah = jumlah + 1
 - ⇒ Dimana id buku = id buku pada data table peminjaman
11. Ketika ada hapus data pada table peminjaman maka jumlah buku akan bertambah 1

Trigger name: delete

Table: peminjaman

Time: AFTER

Event: DELETE

```

1 BEGIN
2   UPDATE buku SET jumlah = jumlah + 1
3   WHERE id_buku = OLD.id_buku;
4 END
  
```

Definer: root@localhost

CREATE

1. Controller peminjaman

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Peminjaman extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('Peminjaman_model');
        login();
    }

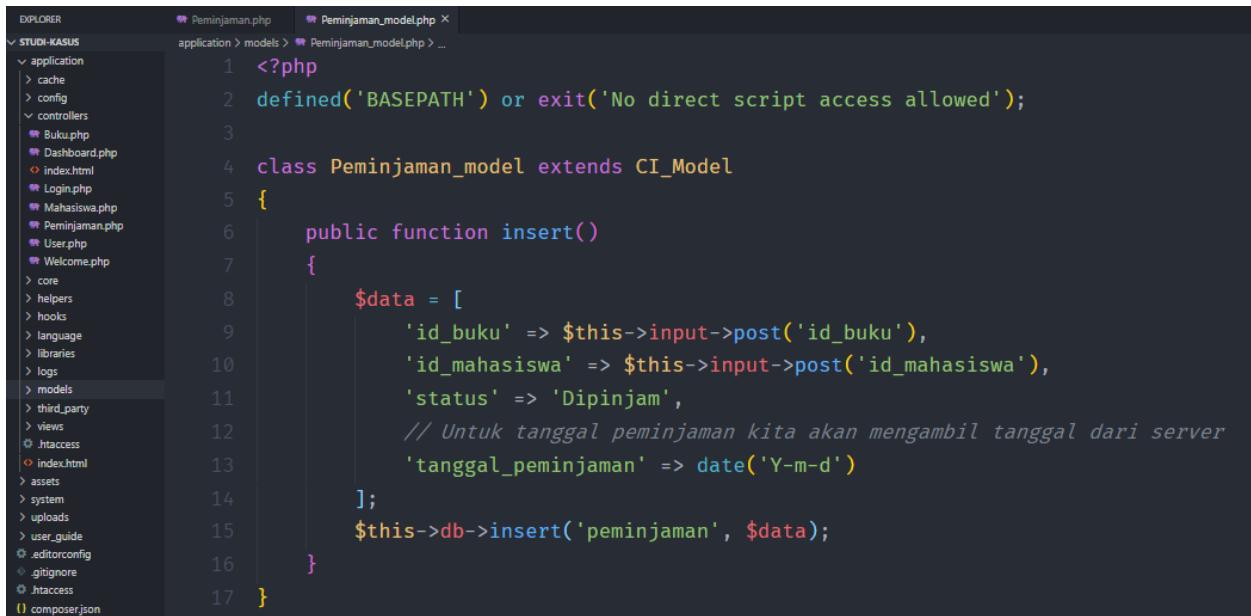
    public function index()
    {
        $data['title'] = 'Data Peminjaman';

        $this->load->view('templates/header', $data);
        $this->load->view('peminjaman/index', $data);
        $this->load->view('templates/footer', $data);
    }
}

public function tambah()
{
    $this->form_validation->set_rules('id_buku', 'id_buku', 'required');
    $this->form_validation->set_rules('id_mahasiswa', 'id_mahasiswa', 'required');
    $this->form_validation->set_error_delimiters('<div class="invalid-feedback text-capitalize">', '</div>');

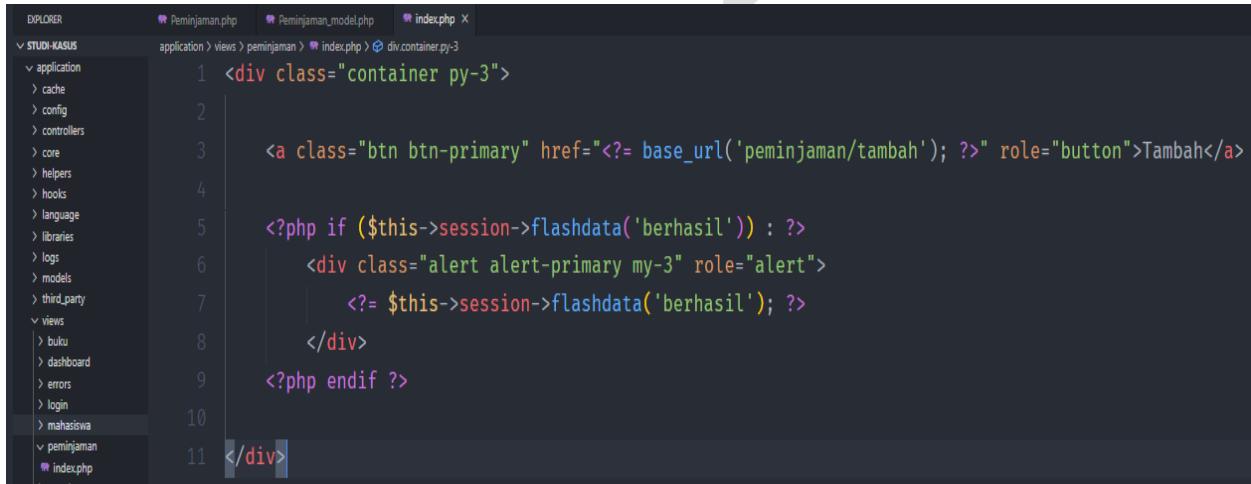
    if ($this->form_validation->run() == FALSE) {
        $data['title'] = 'Tambah Peminjaman';
        $this->load->view('templates/header', $data);
        $this->load->view('peminjaman/tambah', $data);
        $this->load->view('templates/footer', $data);
    } else {
        $this->Peminjaman_model->insert();
        $this->session->set_flashdata('berhasil', 'Data Berhasil Ditambahkan');
        redirect('peminjaman');
    }
}
```

2. Peminjaman model



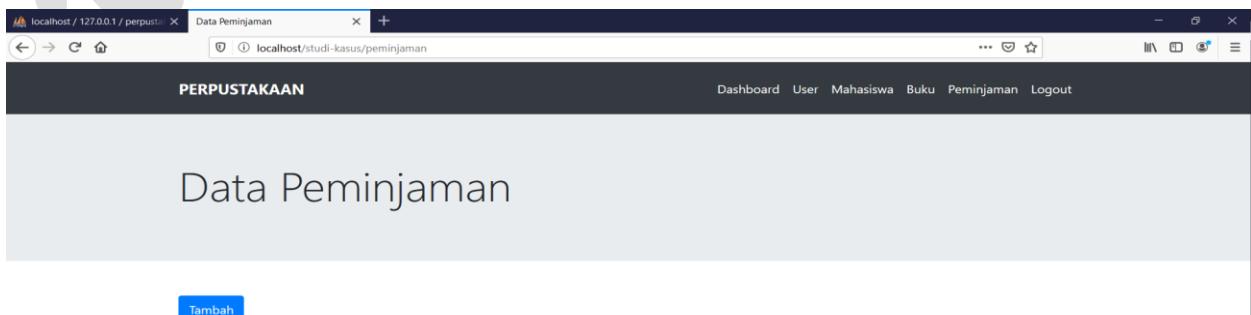
```
1 <?php
2 defined('BASEPATH') or exit('No direct script access allowed');
3
4 class Peminjaman_model extends CI_Model
5 {
6     public function insert()
7     {
8         $data = [
9             'id_buku' => $this->input->post('id_buku'),
10            'id_mahasiswa' => $this->input->post('id_mahasiswa'),
11            'status' => 'Dipinjam',
12            // Untuk tanggal peminjaman kita akan mengambil tanggal dari server
13            'tanggal_peminjaman' => date('Y-m-d')
14        ];
15        $this->db->insert('peminjaman', $data);
16    }
17 }
```

3. View peminjaman index



```
1 <div class="container py-3">
2
3     <a class="btn btn-primary" href="= base_url('peminjaman/tambah'); ?" role="button">Tambah</a>
4
5     <?php if ($this->session->flashdata('berhasil')) : ?>
6         <div class="alert alert-primary my-3" role="alert">
7             <?= $this->session->flashdata('berhasil'); ?>
8         </div>
9     <?php endif ?>
10
11 </div>
```

4. Browser



5. View peminjaman tambah
6. Untuk data mahasiswa dan data buku yang akan kita pilih disarankan jangan menggunakan select dropdown karena jika nantinya datanya banyak scrollnya akan panjangggggg
7. Disini kita akan menggunakan autocomplete dari jQuery
8. Library yang dibutuhkan yaitu jQuery dan jQuery UI

```

EXPLORER          tambah.php X
STUDY-KASUS
  application > views > peminjaman > tambah.php > div.container.mt-3 mb-5
    1  <div class="container mt-3 mb-5">
    2
    3      <?= form_open(); ?>
    4
    5      <!-- mahasiswa -->
    6      <!--
    7          yang akan kita kirim adalah id mahasiswa jadi yang akan kita periksa errornya adalah id mahasiswa
    8          karena name id mahasiswa type hidden, oleh karena itu errornya kita cetak pada input nim
    9          jadi ketika idnya kosong maka input nimnya akan error
   10     -->
   11     <div class="form-group">
   12         <label for="nim">Cari Mahasiswa <span class="text-danger">*</span></label>
   13         <input type="text" class="form-control <?= form_error('id_mahasiswa') ? 'is-invalid' : null; ?>" id="nim"
   14             placeholder="Masukan NIM Mahasiswa">
   15         <?= form_error('id_mahasiswa'); ?>
   16     </div>
   17
   18     <div class="form-group">
   19         <label for="nama">Nama Lengkap <span class="text-danger">*</span></label>
   20         <input type="text" class="form-control <?= form_error('nama') ? 'is-invalid' : null; ?>" id="nama"
   21             name="nama" autocomplete="off" readonly>
   22         <?= form_error('nama'); ?>
   23     </div>
   24
   25     <input type="hidden" name="id_mahasiswa" id="id_mahasiswa">
   26     <!-- end mahasiswa -->
   27
   28
   29     <a class="mr-2 mt-3 btn btn-warning" href="<?= base_url('peminjaman'); ?>" role="button"> Kembali </a>
   30     <button class="mt-3 btn btn-primary" type="submit">Tambah </button>
   31

```

9. Panggil function autocomplete pada footer



application > views > templates > footer.php

```
// jQuery seleksi elemen yang idnya nim
// jalankan fungsi autocomplete (mempunya 2 parameter)
// sumber data dan fungsi ketika datanya ada
$('#nim').autocomplete({
    source: "<?= base_url('peminjaman/getMahasiswa'); ?>",
    select: function(event, ui) {
        // jQuery seleksi elemen yang idnya nim isi valuenya dengan label
        // label jangan diganti dengan nama lain, karena default dari jqueryi autocomplete
        $('#nim').val(ui.item.label);
        $('#id_mahasiswa').val(ui.item.id_mahasiswa);
        $('#nama').val(ui.item.nama);
    }
});
```

10. Method getMahasiswa Controller peminjaman



application > controllers > Peminjaman.php > PHP Intelephense > Peminjaman > getMahasiswa

```
public function getMahasiswa()
{
    // $_GET['term'] bawaan dari jqueryi autocomplete akan menerima parameter yang dikirim dari keyword pada input
    // isset jika variabelnya ada
    if (isset($_GET['term'])) {
        // ambil data mahasiswa berdasarkan parameter yang dikirim
        $mahasiswa = $this->Peminjaman_model->getMahasiswa($_GET['term']);
        // jika data mahasiswa ada
        if ($mahasiswa) {
            // datanya akan kita pecah dan bangun ulang karena awalnya 'nim' = $m->nim
            // sendangkan jqueryi autocomplete akan menerima akan menggap keyword pada input sebagai label
            foreach ($mahasiswa as $m) {
                $data[] = [
                    'label' => $m->nim,
                    'id_mahasiswa' => $m->id_mahasiswa,
                    'nama' => $m->nama
                ];
            }
            /*
             json_encode adalah fungsi yang mengubah format data Array menjadi JSON
             JSON (JavaScript Object Notation)
             Adalah sebuah format data yang digunakan untuk pertukaran dan penyimpanan data
             Data JSON dapat digunakan di semua bahasa pemrograman
            */
            echo json_encode($data);
        }
    }
}
```

11. Model peminjaman

```
> helpers  
> hooks  
> language  
> libraries  
> logs  
< models  
  Buku_model.php  
  index.html  
  Mahasiswa_model.php  
  Peminjaman_model.php  
  User_model.php  
> third_party  
< views  
  buku  
  dashboard  
>  
18   public function getMahasiswa($nim)  
19   {  
20       // cari nim yang mirip $nim  
21       $this->db->like('nim', $nim);  
22       // data yang ditampilkan akan dilimit 5 data saja  
23       $this->db->limit(5);  
24       return $this->db->get('mahasiswa')->result();  
25   }
```

12. Browser

Tambah Peminjaman

Cari Mahasiswa *

Nama Lengkap *

Kembali Tambah

13. Ketika anda menginput pada pencarian maka jQueri autocomplete akan mencari data yang mirip dengan keyword pada input

14. Jika datanya ada tampilkan datanya, maksimal 5 data

Tambah Peminjaman

Cari Mahasiswa *

- 201200001
- 201200002
- 201200003
- 201200004
- 201200005

15. Pilih salah satu maka input nim, id mahasiswa dan nama otomatis disikan

Cari Mahasiswa *

Nama Lengkap *

Kembali Tambah

16. Pencarian untuk buku ➔ view tambah peminjaman



EXPLORER tambah.php X

application views peminjaman tambah.php div.container.mt-3.mb-5

STUHO-KASUS

application cache config controllers core helpers hooks language libraries logs models third_party views buku dashboard errors login mahasiswa peminjaman index.php tambah.php templates user index.html welcome.message.php .htaccess index.html assets system uploads user_guide .editorconfig .gitignore .htaccess composer.json contributing.md index.php license.txt README

```
<!-- buku -->
<div class="form-group">
    <label for="judul">Cari Judul Buku <span class="text-danger">*</span></label>
    <input type="text" class="form-control <?= form_error('id_buku') ? 'is-invalid' : null; ?>" id="judul" placeholder="Masukan Judul Buku"
        <?= form_error('id_buku'); ?>
    </div>

    <div class="row">
        <div class="col-md-4">
            <img src="" alt="" height="400" class="w-100" id="gambar">
        </div>
        <div class="col-md-8">
            <p id="deskripsi"></p>
        </div>
    </div>

    <input type="hidden" name="id_buku" id="id_buku">
<!-- end buku -->

<a class="mr-2 mt-3 btn btn-warning" href=<?= base_url('peminjaman'); ?>" role="button">
    Kembali</a>
<button class="mt-3 btn btn-primary" type="submit">Tambah </button>
</form>
</div>
```

17. Panggil function autocomplete pada footer untuk pencarian buku



```
// buku
> dashboard
> errors
> login
> mahasiswa
> peminjaman
< views
    < buku
    > dashboard
    > errors
    > login
    > mahasiswa
    > peminjaman
    < templates
        < footer.php
        < header.php
        > user
        < index.html
        < welcome_message.php
        < .htaccess
        < index.html
        > assets
        > system
        > uploads
        > user_guide
        < editorconfig
        < .gitignore
        < .htaccess
        < composer.json
        < contributing.md
    > .gitignore
    > .htaccess
    > composer.json
    > contributing.md

// getbuku
$('#judul').autocomplete({
    source: "<?= base_url('peminjaman/getBuku'); ?>",
    select: function(event, ui) {
        // jQuery seleksi elemen yang idnya judul isi valuenya dengan label
        $('#judul').val(ui.item.label);
        $('#id_buku').val(ui.item.id_buku);
        // jQuery seleksi elemen yang idnya deskripsi cetak html dengan deskripsi
        $('#deskripsi').html(ui.item.deskripsi);
        // jQuery seleksi elemen yang idnya gambar isi atribut sourcanya dengan gambar
        $('#gambar').attr("src", "<?= base_url('uploads/'); ?>" + ui.item.gambar);
    }
});
```

18. Method getBuku controller peminjaman



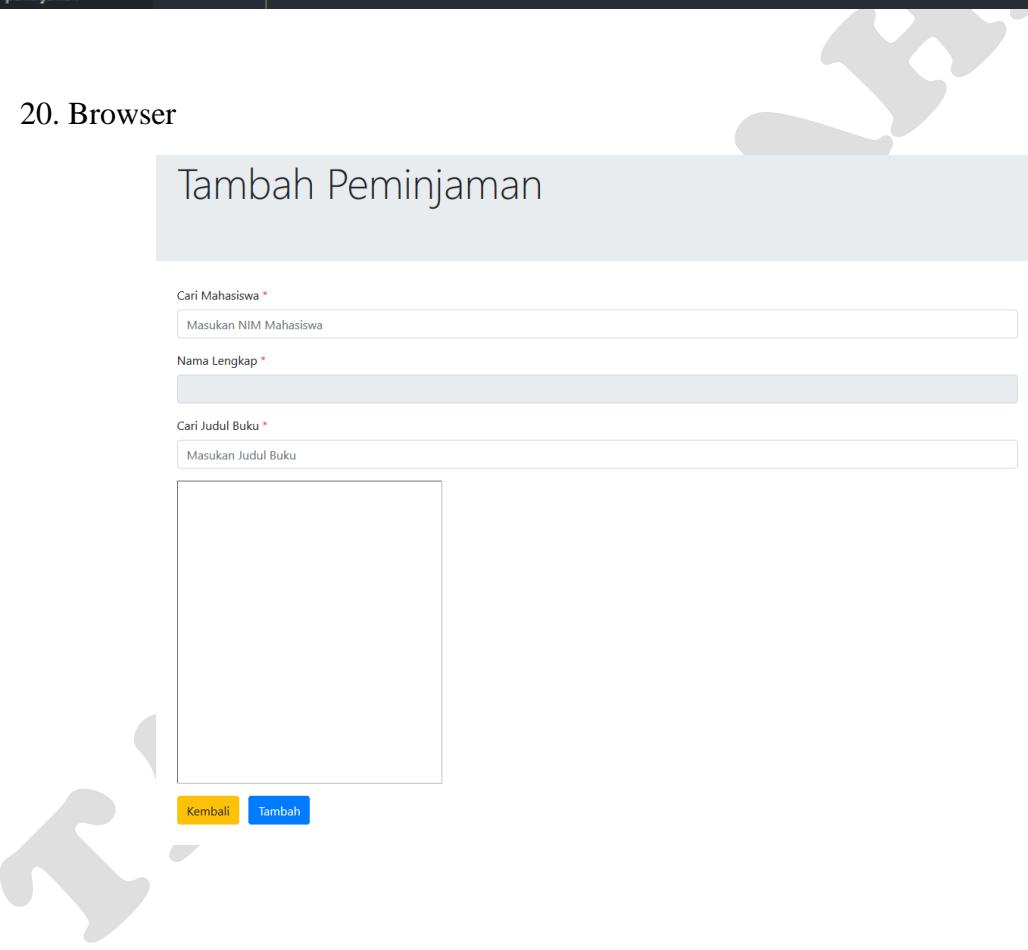
```
EXPLORER          tambah.php  Peminjaman.php  footer.php
STUDI-KASUS       application > controllers > Peminjaman.php > ...
application
    > cache
    > config
    < controllers
        < Buku.php
        < Dashboard.php
        < index.html
        < Login.php
        < Mahasiswa.php
        < Peminjaman.php
        < User.php
        < Welcome.php
    > core
    > helpers
    > hooks
    > language
    > libraries
    > logs
    > models
    > third_party
    < views
        > buku
        > dashboard
        > errors
        > login
        > mahasiswa
        > peminjaman
        < templates
            < footer.php
            < header.php
            > user
            < index.html
            < welcome_message.php
            < .htaccess
            < index.html
            > assets
            > system
            > uploads
            > user_guide
            < editorconfig
            < .gitignore
            < .htaccess
            < composer.json
            < contributing.md
    > .gitignore
    > .htaccess
    > composer.json
    > contributing.md

public function getBuku()
{
    if (isset($_GET['term'])) {
        $buku = $this->Peminjaman_model->getBuku($_GET['term']);
        if ($buku) {
            foreach ($buku as $b)
                $data[] = [
                    'label' => $b->judul,
                    'id_buku' => $b->id_buku,
                    'deskripsi' => $b->deskripsi,
                    'gambar' => $b->gambar
                ];
            echo json_encode($data);
        }
    }
}
```

19. Method getBuku model peminjaman

```
> libraries          26
> logs              27     public function getBuku($judul)
< models           28     {
<   Buku_model.php 29       $this->db->like('judul', $judul);
<   index.html      30       // mengambil data buku dimana jumlah lebih besar dari 0
<   Mahasiswa_model.php 31       $this->db->where('jumlah > 0');
<   Peminjaman_model.php 32       $this->db->limit(5);
<   User_model.php 33       return $this->db->get('buku')->result();
> third_party        34     }
```

20. Browser



Tambah Peminjaman

Cari Mahasiswa *

Nama Lengkap *

Cari Judul Buku *

[Kembali](#) [Tambah](#)

21. Pegujian

- ⇒ Cari buku yang jumlahnya 0 → seharusnya tidak muncul
- ⇒ Jika memilih salah satu judul
- ⇒ Sebelum meminjam cek jumlah buku yang akan dipinjam
- ⇒ Bandingkan jumlah bukunya setelah berhasil melakukan peminjaman
- ⇒ Seharusnya bukunya berkurang 1

Cari Mahasiswa *

Nama Lengkap *

Cari Judul Buku *

On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These cases are perfectly simple and easy to distinguish. In a free hour, when our power of choice is untrammelled and when nothing prevents our being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted. The wise man therefore always holds in these matters to this principle of selection: he rejects pleasures to secure other greater pleasures, or else he endures pains to avoid worse pains.

22. Peminjaman berhasil

Data Berhasil Ditambahkan

23. Database table peminjaman

+ Options	← ↑ →	▼	id_peminjaman	id_buku	id_mahasiswa	tanggal_peminjaman	tanggal_pengembalian	status
<input type="checkbox"/>   			1	1	1	2020-04-29	0000-00-00	Dipinjam

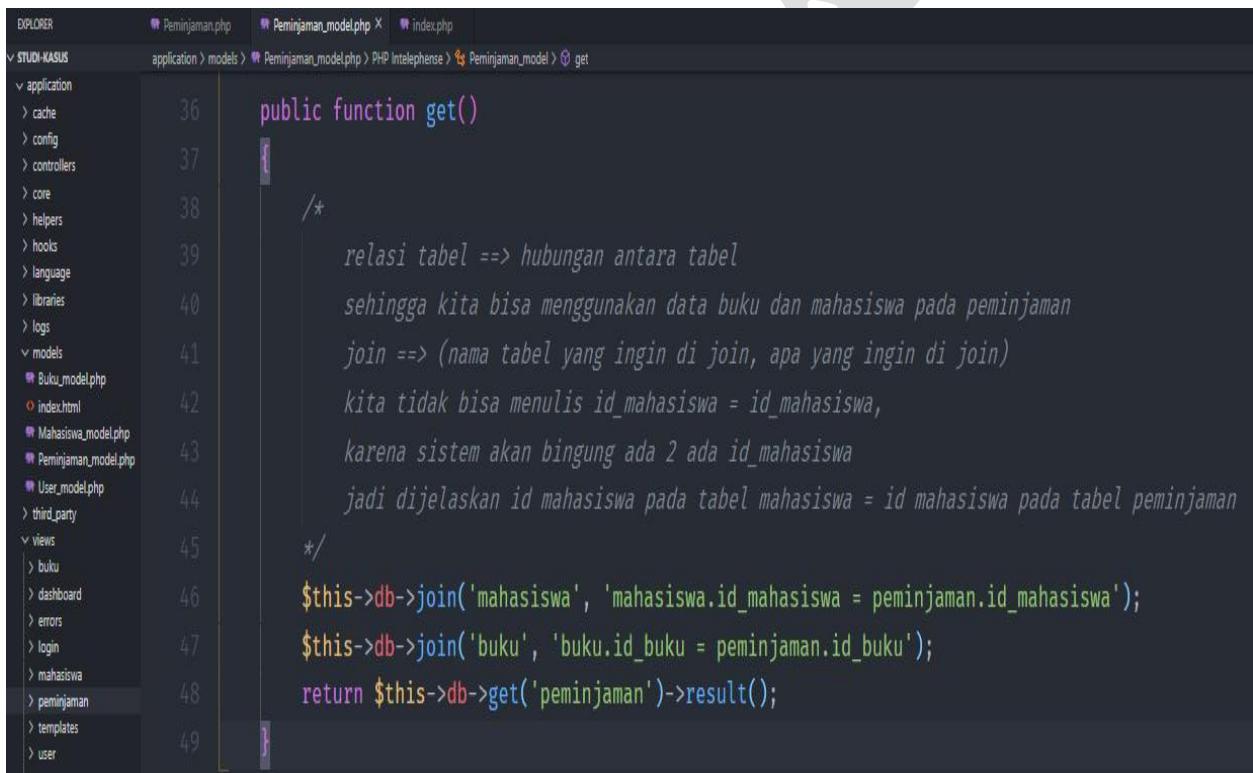
READ

1. Controller peminjaman



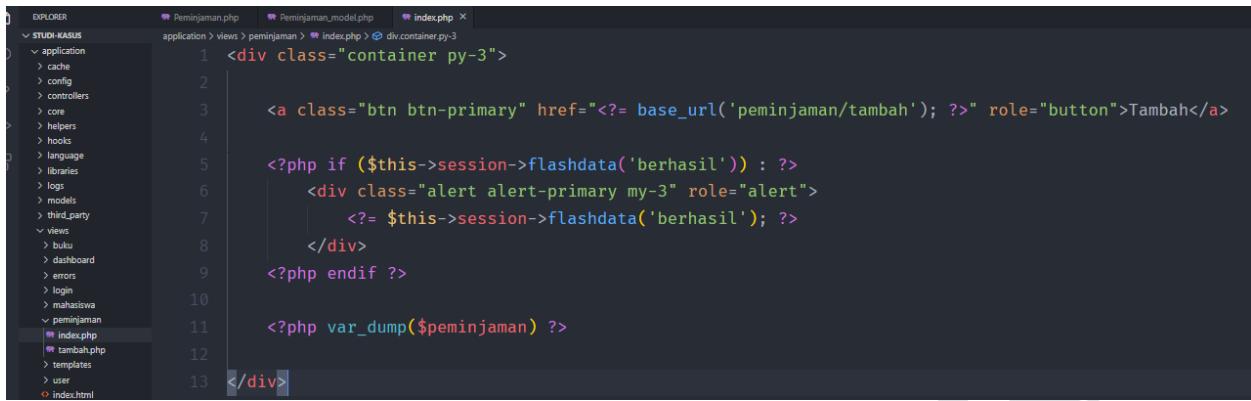
```
application
> cache
> config
< controllers
  Buku.php
  Dashboard.php
  index.html
  Login.php
  Mahasiswa.php
  Peminjaman.php
  User.php
  Welcome.php
> core
> helpers
> hooks
> language
> libraries
> logs
13   public function index()
14   {
15       $data['title'] = 'Data Peminjaman';
16       $data['peminjaman'] = $this->Peminjaman_model->get();
17
18       $this->load->view('templates/header', $data);
19       $this->load->view('peminjaman/index', $data);
20       $this->load->view('templates/footer', $data);
21   }
```

2. Model peminjaman



```
EXPLORER      Peminjaman.php  Peminjaman_model.php  index.php
STUDI-KASUS    application > models > Peminjaman_model.php > PHP Intelephense > Peminjaman_model > get
application
> cache
> config
> controllers
> core
> helpers
> hooks
> language
> libraries
> logs
< models
  Buku_model.php
  index.html
  Mahasiswa_model.php
  Peminjaman_model.php
  User_model.php
> third_party
< views
  buku
  dashboard
  errors
  login
  mahasiswa
  peminjaman
  templates
  user
36   public function get()
37   /*
38   * relasi tabel ==> hubungan antara tabel
39   * sehingga kita bisa menggunakan data buku dan mahasiswa pada peminjaman
40   * join ==> (nama tabel yang ingin di join, apa yang ingin di join)
41   * kita tidak bisa menulis id_mahasiswa = id_mahasiswa,
42   * karena sistem akan bingung ada 2 ada id_mahasiswa
43   * jadi dijelaskan id mahasiswa pada tabel mahasiswa = id mahasiswa pada tabel peminjaman
44   */
45
46   $this->db->join('mahasiswa', 'mahasiswa.id_mahasiswa = peminjaman.id_mahasiswa');
47   $this->db->join('buku', 'buku.id_buku = peminjaman.id_buku');
48   return $this->db->get('peminjaman')->result();
49 }
```

3. View peminjaman index



```
<div class="container py-3">
    <a class="btn btn-primary" href="= base_url('peminjaman/tambah'); ?" role="button">Tambah</a>
    <?php if ($this->session->flashdata('berhasil')) : ?>
        <div class="alert alert-primary my-3" role="alert">
            <?= $this->session->flashdata('berhasil'); ?>
        </div>
    <?php endif ?>
    <?php var_dump($peminjaman) ?>
</div>
```

4. Pada saat di var dump data yang dicetak adalah data peminjaman, data buku dan data mahasiswa

Data Peminjaman

Tambah array(1) { [0]=> object(stdClass)#22 (12) { ["id_peminjaman"]=> string(1) "1" ["id_buku"]=> string(1) "1" ["id_mahasiswa"]=> string(1) "1" ["tanggal_peminjaman"]=> string(10) "2020-04-29" ["tanggal_pengembalian"]=> string(10) "0000-00-00" ["status"]=> string(8) "Dipinjam" ["nama"]=> string(11) "Eren Yeager" ["nim"]=> string(9) "201200001" ["judul"]=> string(18) "One Piece Arc Wano" ["deskripsi"]=> string(1003) "On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These cases are perfectly simple and easy to distinguish. In a free hour, when our power of choice is untrammeled and when nothing prevents our being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted. The wise man therefore always holds in these matters to this principle of selection: he rejects pleasures to secure other greater pleasures, or else he endures pains to avoid worse pains.

"[jumlah"]=> string(1) "4" ["gambar"]=> string(36) "638d79fe333b9d297853be09bbff19b6.jpg" } }

5. Hapus var dump, datanya kita cetak pada table



```
<div class="table-responsive my-3">
    <table class="table table-striped table-bordered text-center" id="my-table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Tanggal</th>
                <th scope="col">Judul</th>
                <th scope="col">NIM</th>
                <th scope="col">Status</th>
                <th scope="col">Aksi</th>
            </tr>
        </thead>
```

```

 23 <tbody>
 24     <?php $i = 1; ?>
 25     <?php foreach ($peminjaman as $p) : ?>
 26         <tr>
 27             <th scope="row"><?= $i; ?></th>
 28             <td><?= date('d-m-Y', strtotime($p->tanggal_peminjaman)); ?></td>
 29             <td><?= $p->judul; ?></td>
 30             <td><?= $p->nim; ?></td>
 31             <td><?= $p->status; ?></td>
 32             <td>
 33                 <a class="btn btn-outline-warning" href="<?= base_url('peminjaman/ubah/' .
 34                     $p->id_peminjaman); ?>" role="button">
 35                     <i class="fa fa-edit"></i>
 36                 </a>
 37                 <a class="btn btn-outline-danger" href="<?= base_url('peminjaman/hapus/' .
 38                     $p->id_peminjaman); ?>" onclick="return confirm('Anda Yakin?')" role="button">
 39                     <i class="fa fa-trash"></i>
 40                 </a>
 41             </td>
 42         </tr>
 43     <?php $i++ ?>
 44     <?php endforeach; ?>
 45     </tbody>
 46 </table>
 47 </div>

```

6. Browser

Tambah						
Show <input type="text" value="10"/> entries Search: <input type="text"/>						
#	Tanggal	Judul	NIM	Status	Aksi	
1	29-04-2020	One Piece Arc Wano	201200001	Dipinjam	 	

Showing 1 to 1 of 1 entries Previous Next

UPDATE

1. Method ubah controller peminjaman

The screenshot shows the PHP Intelephense code editor interface. The left sidebar displays the project structure under 'STUDI-KASUS'. The main editor window shows the 'Peminjaman.php' file with the following code:

```
public function ubah($id_peminjaman)
{
    $this->form_validation->set_rules('id_buku', 'id_buku', 'required');
    $this->form_validation->set_rules('id_mahasiswa', 'id_mahasiswa', 'required');
    $this->form_validation->set_error_delimiters('<div class="invalid-feedback text-capitalize">', '</div>');

    if ($this->form_validation->run() == FALSE) {
        $data['title'] = 'Ubah Peminjaman';
        $data['peminjaman'] = $this->Peminjaman_model->getRow($id_peminjaman);

        $this->load->view('templates/header', $data);
        $this->load->view('peminjaman/ubah', $data);
        $this->load->view('templates/footer', $data);
    } else {
        $this->Peminjaman_model->update();
        $this->session->set_flashdata('berhasil', 'Data Berhasil Diubah');
        redirect('peminjaman');
    }
}
```

2. Method getRow model peminjaman

The screenshot shows the PHP Intelephense code editor interface. The left sidebar displays the project structure under 'STUDI-KASUS'. The main editor window shows the 'Peminjaman_model.php' file with the following code:

```
public function getRow($id_peminjaman)
{
    $this->db->join('mahasiswa', 'mahasiswa.id_mahasiswa = peminjaman.id_mahasiswa');
    $this->db->join('buku', 'buku.id_buku = peminjaman.id_buku');
    $where = ['id_peminjaman' => $id_peminjaman];
    return $this->db->get_where('peminjaman', $where)->row();
}
```

3. Method update model peminjaman

The screenshot shows the PHP Intelephense code editor interface. The left sidebar displays the project structure under 'STUDI-KASUS'. The main editor window shows the 'Peminjaman_model.php' file with the following code:

```
public function update()
{
    $data = [
        'id_buku' => $this->input->post('id_buku'),
        'id_mahasiswa' => $this->input->post('id_mahasiswa'),
        'status' => 'Dipinjam',
        'tanggal_peminjaman' => date('Y-m-d')
    ];
    $id_peminjaman = $this->input->post('id_peminjaman');
    $this->db->where('id_peminjaman', $id_peminjaman);
    $this->db->update('peminjaman', $data);
}
```

4. View peminjaman ubah

```
<?php

// Peminjaman.php
use App\Models\Peminjaman;
use App\Models\Mahasiswa;
use App\Models\Buku;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Session;

class PeminjamanController extends Controller
{
    public function index()
    {
        $peminjamans = Peminjaman::all();
        return view('peminjaman.index', compact('peminjamans'));
    }

    public function tambah()
    {
        $mahasiswas = Mahasiswa::all();
        $bukus = Buku::all();
        return view('peminjaman.tambah', compact('mahasiswas', 'bukus'));
    }

    public function ubah($id)
    {
        $peminjaman = Peminjaman::find($id);
        $mahasiswa = Mahasiswa::find($peminjaman->id_mahasiswa);
        $buku = Buku::find($peminjaman->id_buku);

        return view('peminjaman.ubah', compact('peminjaman', 'mahasiswa', 'buku'));
    }

    public function simpan(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'id_mahasiswa' => 'required|exists:mahasiswa,id',
            'id_buku' => 'required|exists:buku,id'
        ]);

        if ($validator->fails())
        {
            return Redirect::back()->withErrors($validator->errors());
        }

        $peminjaman = new Peminjaman();
        $peminjaman->id_mahasiswa = $request->id_mahasiswa;
        $peminjaman->id_buku = $request->id_buku;
        $peminjaman->tgl_peminjaman = $request->tgl_peminjaman;
        $peminjaman->tgl_kembali = $request->tgl_kembali;
        $peminjaman->status = $request->status;

        $peminjaman->save();

        Session::flash('success', 'Peminjaman berhasil ditambahkan');
        return Redirect::route('peminjaman.index');
    }

    public function hapus($id)
    {
        $peminjaman = Peminjaman::find($id);
        $peminjaman->delete();

        Session::flash('success', 'Peminjaman berhasil dihapus');
        return Redirect::route('peminjaman.index');
    }
}
```

```
<?php

// ubah.php
use App\Models\Peminjaman;
use App\Models\Mahasiswa;
use App\Models\Buku;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Session;

class PeminjamanController extends Controller
{
    public function index()
    {
        $peminjamans = Peminjaman::all();
        return view('peminjaman.index', compact('peminjamans'));
    }

    public function tambah()
    {
        $mahasiswas = Mahasiswa::all();
        $bukus = Buku::all();
        return view('peminjaman.tambah', compact('mahasiswas', 'bukus'));
    }

    public function ubah($id)
    {
        $peminjaman = Peminjaman::find($id);
        $mahasiswa = Mahasiswa::find($peminjaman->id_mahasiswa);
        $buku = Buku::find($peminjaman->id_buku);

        return view('peminjaman.ubah', compact('peminjaman', 'mahasiswa', 'buku'));
    }

    public function simpan(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'id_mahasiswa' => 'required|exists:mahasiswa,id',
            'id_buku' => 'required|exists:buku,id'
        ]);

        if ($validator->fails())
        {
            return Redirect::back()->withErrors($validator->errors());
        }

        $peminjaman = new Peminjaman();
        $peminjaman->id_mahasiswa = $request->id_mahasiswa;
        $peminjaman->id_buku = $request->id_buku;
        $peminjaman->tgl_peminjaman = $request->tgl_peminjaman;
        $peminjaman->tgl_kembali = $request->tgl_kembali;
        $peminjaman->status = $request->status;

        $peminjaman->save();

        Session::flash('success', 'Peminjaman berhasil ditambahkan');
        return Redirect::route('peminjaman.index');
    }

    public function hapus($id)
    {
        $peminjaman = Peminjaman::find($id);
        $peminjaman->delete();

        Session::flash('success', 'Peminjaman berhasil dihapus');
        return Redirect::route('peminjaman.index');
    }
}
```

5. Browser

Ubah Peminjaman

Cari Mahasiswa *

201200001

Nama Lengkap *

Eren Yeager

Cari Judul Buku *

One Piece Arc Wano



On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These cases are perfectly simple and easy to distinguish. In a free hour, when our power of choice is untrammelled and when nothing prevents our being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted. The wise man therefore always holds in these matters to this principle of selection: he rejects pleasures to secure other greater pleasures, or else he endures pains to avoid worse pains.

Kembali

Ubah

6. Pengujian

- ⇒ Perhatikan jumlah buku yang sementara mau diubah, ubah mahasiswa seharusnya jumlah bukunya tetap
- ⇒ Perhatikan jumlah buku yang sementara mau diubah, ubah buku buku yang awalnya dipinjam jumlahnya akan bertambah 1 dan buku yang di update pada peminjaman akan berkurang 1

Tambah

Data Berhasil Diubah

Show 10 entries

Search:

#	Tanggal	Judul	NIM	Status	Aksi

DETAIL

1. Tambahkan tombol detail dengan menggunakan modal pada table peminjaman
2. Data target dan id pada modal detail peminjaman jangan disamakan dengan detail buku

```
<td><?= $p->status; ?></td>
<td>
    <button type="button" class="btn btn-outline-primary get-detail-peminjaman" data-toggle="modal"
    data-target="#detail-peminjaman" data-id="<?= $p->id_peminjaman; ?>">
        <i class="fa fa-eye"></i>
    </button>
    <a class="btn btn-outline-warning" href="<?= base_url('peminjaman/ubah/' . $p->id_peminjaman); ?>">
        ... <!-- Lihat Detail -->
    </a>
</td>
```

3. Buat modal detail peminjaman pada footer



```
> third_party
  <-- modal detail buku -->
30  <div class="modal fade" id="detail-peminjaman" tabindex="-1" role="dialog"
31    aria-labelledby="exampleModalLabel" aria-hidden="true">
32    <div class="modal-dialog modal-xl" role="document">
33      <div class="modal-content">
34        <div class="modal-header">
35          <h5 class="modal-title" id="exampleModalLabel">Detail Peminjaman</h5>
36          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
37            <span aria-hidden="true">&times;</span>
38          </button>
39        </div>
40        <div class="modal-body detail-peminjaman">
41          ...
42        </div>
43        <div class="modal-footer">
44          <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
45        </div>
46      </div>
47    </div>
48  </div>
49  <!-- end modal detail buku -->
```

4. Buat script untuk memanggil file detail peminjaman pada footer

```
w tamuandipu
  ubah.php
  templates
    footer.php
    header.php
  user
  index.html
  welcome_message.php
  .htaccess
  index.html
```

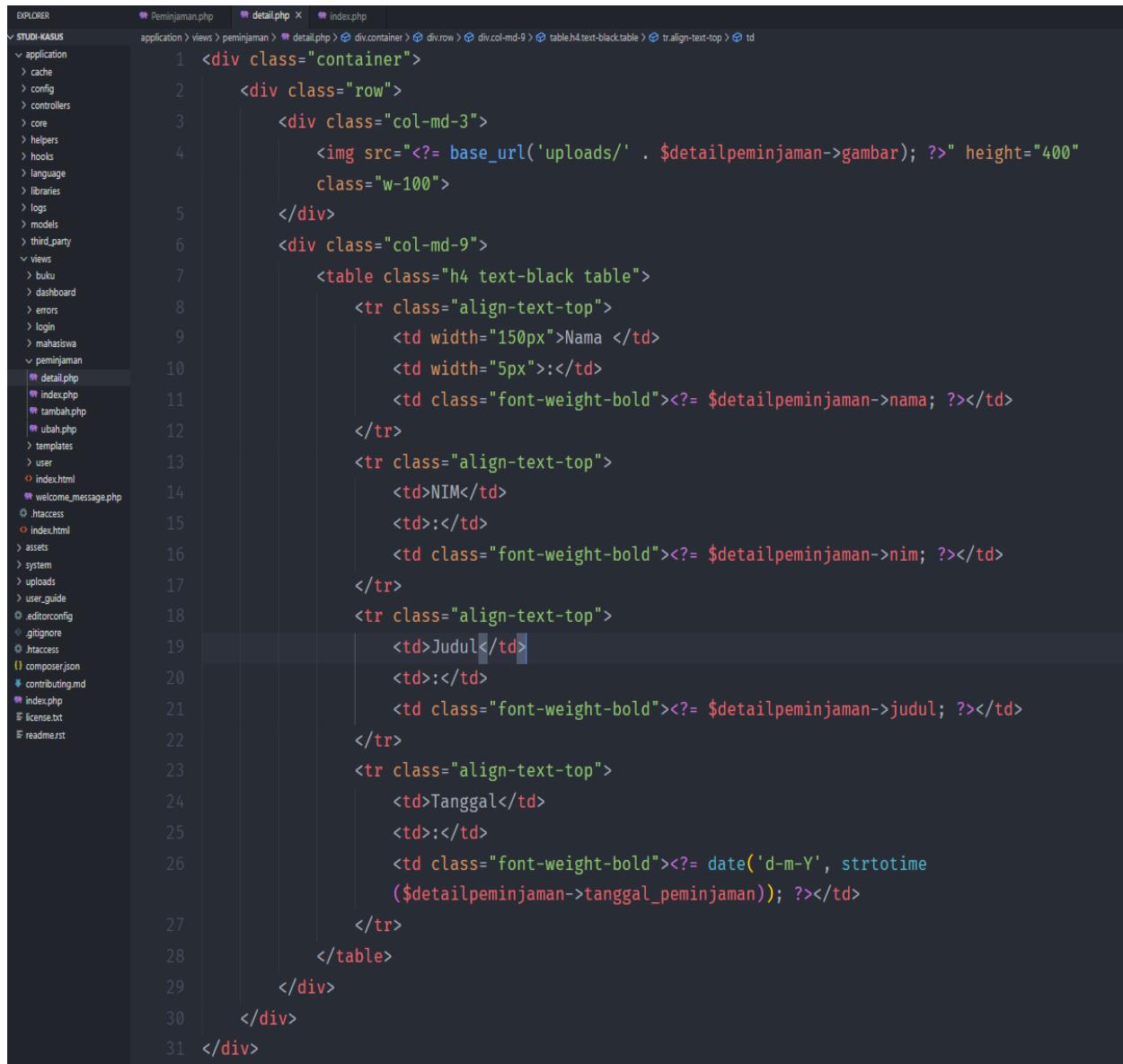
	103	
	104	\$('my-table').on('click', '.get-detail-peminjaman', function() {
	105	const id = \$(this).data('id');
	106	\$(".detail-peminjaman").load("<?= base_url('peminjaman/detail/')?>" + id);
	107	});

5. Method detail controller peminjaman

```
Buku.php
Dashboard.php
index.html
Login.php
Mahasiswa.php
Peminjaman.php
User.php
Welcome.php
```

	107	
	108	function detail()
	109	{
	110	\$id_peminjaman = \$this->uri->segment(3);
	111	\$data['detailpeminjaman'] = \$this->Peminjaman_model->getRow(\$id_peminjaman);
	112	\$this->load->view('peminjaman/detail', \$data);
	113	}

6. View detail peminjaman



The screenshot shows a code editor with the file `detail.php` open. The code displays a loan record with its details. The record includes the name (`Historia Reiss`), NIM (`201200005`), title (`One Piece Arc Zou`), and date (`29-04-2020`). An image of the `One Piece` comic book cover is also shown.

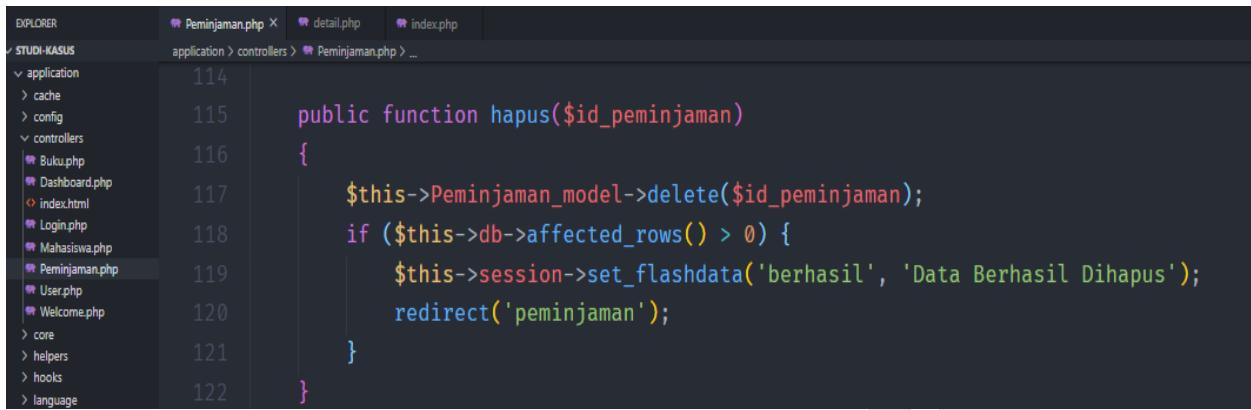
```
<div class="container">
<div class="row">
    <div class="col-md-3">
        
    </div>
    <div class="col-md-9">
        <table class="h4 text-black table">
            <tr class="align-text-top">
                <td width="150px">Nama </td>
                <td width="5px">:</td>
                <td class="font-weight-bold"><?= $detailpeminjaman->nama; ?></td>
            </tr>
            <tr class="align-text-top">
                <td>NIM</td>
                <td>:</td>
                <td class="font-weight-bold"><?= $detailpeminjaman->nim; ?></td>
            </tr>
            <tr class="align-text-top">
                <td>Judul</td>
                <td>:</td>
                <td class="font-weight-bold"><?= $detailpeminjaman->judul; ?></td>
            </tr>
            <tr class="align-text-top">
                <td>Tanggal</td>
                <td>:</td>
                <td class="font-weight-bold"><?= date('d-m-Y', strtotime($detailpeminjaman->tanggal_peminjaman)); ?></td>
            </tr>
        </table>
    </div>
</div>
</div>
```

7. Browser



DELETE

1. Method hapus controller peminjaman



The screenshot shows a code editor with the file `Peminjaman.php` open. The code defines a `hapus` method:

```
114
115     public function hapus($id_peminjaman)
116     {
117         $this->Peminjaman_model->delete($id_peminjaman);
118         if ($this->db->affected_rows() > 0) {
119             $this->session->set_flashdata('berhasil', 'Data Berhasil Dihapus');
120             redirect('peminjaman');
121         }
122     }
```

2. Method delete model peminjaman



The screenshot shows a code editor with the file `Peminjaman_model.php` open. The code defines a `delete` method:

```
72     public function delete($id_peminjaman)
73     {
74         $this->db->where('id_peminjaman', $id_peminjaman);
75         $this->db->delete('peminjaman');
76     }
```

3. Pengujian

⇒ Perhatikan jumlah buku yang dipinjam, jika data peminjamannya di hapus jumlah bukunya akan bertambah

PENGEMBALIAN

1. Untuk pengembalian kita hanya merubah statusnya dari dipinjam menjadi dikembalikan dan jumlah bukunya ditambah 1
2. Buat tombol pengembalian

```
<i class="fa fa-trash"></i>
</a>
<a class="btn btn-outline-secondary" href="= base_url('peminjaman/pengembalian/' . $p-&gt;id_peminjaman); ?" role="button" onclick="return confirm('Anda Yakin?')">
    <i class="fa fa-sign-in-alt"></i>
</a>
</td>
```

3. Untuk pengembalian kita tidak bisa menggunakan trigger pada table peminjaman, karena trigger pada saat update akan berkurang 1 kemudian bertambah 1 jadi jumlah buku tetap
4. Seharusnya jumlah buku akan bertambah 1 jika buku dikembalikan
5. Jadi kita akan membuatnya manual.
6. Method pengembalian controller peminjaman

```
> cache
> config
< controllers
  Buku.php
  Dashboard.php
  index.html
  Login.php
  Mahasiswa.php
  Peminjaman.php
  User.php
  Welcome.php
> core
> helpers
> hooks
> language
> libraries
> logs
< models
  Buku_model.php
  index.html
  Mahasiswa_model.php
  Peminjaman_model.php
  User_model.php
> third_party
< views
  buku
> dashboard
```

```
124
125
126
127
128
129
130
131
132
133
134
135
136
137
```

```
public function pengembalian($id_peminjaman)
{
    // mengembalikan buku
    $this->Peminjaman_model->return($id_peminjaman);
    // mengambil data pengembalian karena kita butuh jumlah buku dan id buku
    $peminjaman = $this->Peminjaman_model->getRow($id_peminjaman);
    // jumlah buku baru
    $jumlah = $peminjaman->jumlah + 1;
    $id_buku = $peminjaman->id_buku;
    // update jumlah buku baru
    $this->Peminjaman_model->updateBuku($jumlah, $id_buku);
    $this->session->set_flashdata('berhasil', 'Buku Berhasil Dekembalikan');
    redirect('peminjaman');
```

7. Model pengembalian

```
index.html          78  public function return($id_peminjaman)
Login.php          79  {
Mahasiswa.php      79
Peminjaman.php    80  $data = [
User.php          80  'status' => 'Dikembalikan',
Welcome.php       81  'tanggal_pengembalian' => date('Y-m-d')
                     ];
core               81
helpers            82
hooks              83
language           83
libraries          84
logs               84
models              >
Buku_model.php    85  $this->db->where('id_peminjaman', $id_peminjaman);
index.html         85
Mahasiswa_model.php 86  $this->db->update('peminjaman', $data);
Peminjaman_model.php 87
User_model.php     87
third_party        88  public function updateBuku($jumlah, $id_buku)
views              88
buku               89  {
dashboard          89
errors              >
login               90
mahasiswa          91  $data = ['jumlah' => $jumlah];
peminjaman         92  $this->db->where('id_buku', $id_buku);
detail.php         92  $this->db->update('buku', $data);
index.php          93  }
tambah.php         93
```

8. Browser

Data Peminjaman

Data Peminjaman						
Tambah						
Search:						
#	Tanggal	Judul	NIM	Status	Aksi	
1	29-04-2020	One Piece Arc Wano	201200001	Dipinjam	 	
2	29-04-2020	One Piece Arc Zou	201200002	Dipinjam	 	
3	29-04-2020	One Piece Arc Dressrosa	201200003	Dipinjam	 	

Showing 1 to 3 of 3 entries

Previous [1](#) Next

9. Pengujian

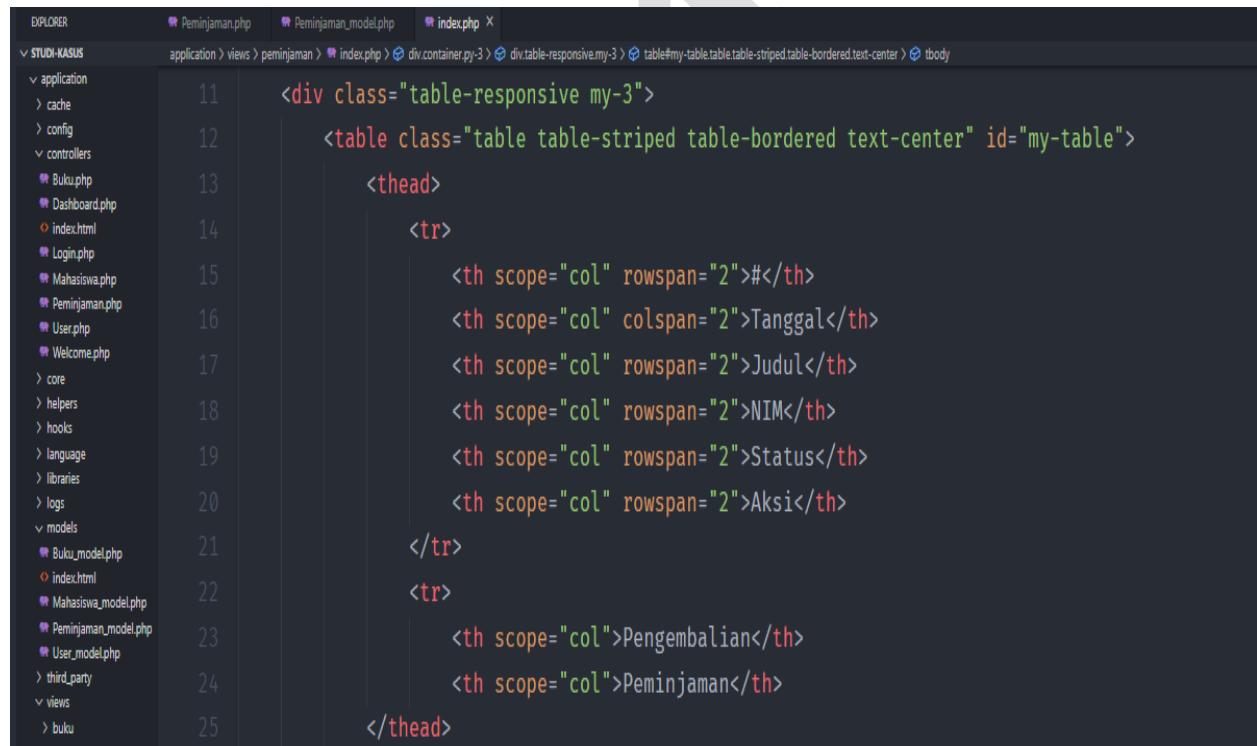
⇒ Perhatikan jumlah buku yang dipinjam, pada saat dikembalikan jumlah bukunya bertambah 1



The screenshot shows a table with a single row of data. The columns are labeled '#', 'Tanggal', 'Judul', 'NIM', 'Status', and 'Aksi'. The data row contains the values: 1, 29-04-2020, One Piece Arc Wano, 201200001, Dikembalikan, and four action buttons (eye, edit, delete, refresh). Above the table, a blue header bar says 'Buku Berhasil Dekembalikan'.

#	Tanggal	Judul	NIM	Status	Aksi
1	29-04-2020	One Piece Arc Wano	201200001	Dikembalikan	

10. Ketika bukunya telah dikembalikan seharusnya tombol update, ubah, hapus dan pengembalian sudah tidak ada
11. Kemudian tanggal pengembalian juga harusnya muncul
12. Ubah table peminjaman



The screenshot shows a code editor with a dark theme. On the left is the file structure (EXPLORER) showing files like Peminjaman.php, Peminjaman_model.php, and index.php. The main area shows the code for the table head:

```
<div class="table-responsive my-3">
    <table class="table table-striped table-bordered text-center" id="my-table">
        <thead>
            <tr>
                <th scope="col" rowspan="2">#</th>
                <th scope="col" rowspan="2">Tanggal</th>
                <th scope="col" rowspan="2">Judul</th>
                <th scope="col" rowspan="2">NIM</th>
                <th scope="col" rowspan="2">Status</th>
                <th scope="col" rowspan="2">Aksi</th>
            </tr>
            <tr>
                <th>Pengembalian</th>
                <th>Peminjaman</th>
            </tr>
        </thead>
```

```

<tbody>
    <?php $i = 1; ?>
    <?php foreach ($peminjaman as $p) : ?>
        <tr>
            <th scope="row"><?= $i; ?></th>
            <td><?= date('d-m-Y', strtotime($p->tanggal_peminjaman)); ?></td>
            <td><?= $p->status == 'Dikembalikan' ? date('d-m-Y', strtotime($p->tanggal_pengembalian)) : '' ; ?></td>
            <td><?= $p->judul; ?></td>
            <td><?= $p->nim; ?></td>
            <td><?= $p->status; ?></td>
            <td class="text-nowarp">
                <button type="button" class="btn btn-outline-primary get-detail-peminjaman" data-toggle="modal"
                    data-target="#detail-peminjaman" data-id="<?= $p->id_peminjaman; ?>">
                    <i class="fa fa-eye"></i>
                </button>
                <?php if ($p->status == 'Dipinjam') : ?>
                    <a class="btn btn-outline-warning" href="<?= base_url('peminjaman/ubah/' . $p->id_peminjaman); ?>"
                        role="button">
                        <i class="fa fa-edit"></i>
                    </a>
                    <a class="btn btn-outline-danger" href="<?= base_url('peminjaman/hapus/' . $p->id_peminjaman); ?>"
                        onclick="return confirm('Anda Yakin?')" role="button">
                        <i class="fa fa-trash"></i>
                    </a>
                    <a class="btn btn-outline-secondary" href="<?= base_url('peminjaman/pengembalian/' .
                        $p->id_peminjaman); ?>" role="button" onclick="return confirm('Anda Yakin?')">
                        <i class="fa fa-sign-in-alt"></i>
                    </a>
                <?php endif ?>
            </td>
        </tr>
        <?php $i++ ?>
    <?php endforeach; ?>
</tbody>
</table>
</div>

```

13. Browser

Tanggal							
#	Pengembalian	Peminjaman	Judul	NIM	Status	Aksi	
1	29-04-2020	29-04-2020	One Piece Arc Wano	201200001	Dikembalikan		
2	29-04-2020		One Piece Arc Zou	201200002	Dipinjam		
3	29-04-2020		One Piece Arc Dressrosa	201200003	Dipinjam		

Showing 1 to 3 of 3 entries

Previous 1 Next

RESET

1. Reset dibutuhkan jika terjadi kesalahan dalam pengembalian
2. Kita ingin mengembalikan statusnya dari dikembalikan menjadi dipinjam
3. Tombol ini hanya akan muncul ketika buku telah dikembalikan
4. Buat tombolnya pada tabel peminjaman

```
<?php if ($p->status == 'Dikembalikan') : ?>
    <a class="btn btn-outline-success" href="= base_url('peminjaman/reset/' . $p-&gt;id_peminjaman); ?" role="button" onclick="return confirm('Anda Yakin?')">
        <i class="fa fa-retweet"></i>
    </a>
</?php endif ?>
</td>
</tr>
<?php $i++ ?>
```

5. Browser

Tanggal							Aksi
#	Pengembalian	Peminjaman	Judul	NIM	Status		
1	29-04-2020	29-04-2020	One Piece Arc Wano	201200001	Dikembalikan	 	
2	29-04-2020		One Piece Arc Zou	201200002	Dipinjam	   	
3	29-04-2020		One Piece Arc Dressrosa	201200003	Dipinjam	   	

Showing 1 to 3 of 3 entries

Previous **1** Next

6. Method reset controller peminjaman

```
> cache
> config
> controllers
  ↗ Buku.php
  ↗ Dashboard.php
  ↗ index.html
  ↗ Login.php
  ↗ Mahasiswa.php
  ↗ Peminjaman.php
  ↗ User.php
  ↗ Welcome.php
> core
> helpers
> hooks
> language
> libraries
> logs
> models
  ↗ Buku_model.php
138
139    public function reset($id_peminjaman)
140    {
141        $this->Peminjaman_model->reset($id_peminjaman);
142        $peminjaman = $this->Peminjaman_model->getRow($id_peminjaman);
143        $jumlah = $peminjaman->jumlah - 1;
144        $id_buku = $peminjaman->id_buku;
145        $this->Peminjaman_model->updateBuku($jumlah, $id_buku);
146        $this->session->set_flashdata('berhasil', 'Data Berhasil Direset');
147        redirect('peminjaman');
148    }
```

7. Method reset model peminjaman

```
> language          94
> libraries        95
> logs             96
< models           97
  < Buku_model.php 98
  < index.html      99
  < Mahasiswa_model.php
  < Peminjaman_model.php
  < User_model.php
> third_party      100
< views            101
  < buku
  < dashboard
  < errors
  < login
  < mahasiswa
< peminjaman       102
  < detail.php      103

public function reset($id_peminjaman)
{
    $data = [
        'status' => 'Dipinjam',
        'tanggal_pengembalian' => '0000-00-00'
    ];
    $this->db->where('id_peminjaman', $id_peminjaman);
    $this->db->update('peminjaman', $data);
}
```

8. Pengujian

- ⇒ Periksa jumlah buku yang telah dikembalikan ketika direset jumlah bukunya akan berkurang 1 dan status peminjaman pada table peminjaman akan berubah

Data Berhasil Direset						
<input type="button" value="Show"/> <input type="button" value="10"/> entries <input type="text" value="Search:"/>						
#	Tanggal		Judul	NIM	Status	Aksi
	Pengembalian	Peminjaman				
1	29-04-2020		One Piece Arc Wano	201200001	Dipinjam	   
2	29-04-2020		One Piece Arc Zou	201200002	Dipinjam	   

CODE REPAIR

- Jika tinggi content tidak full terhadap layar maka akan terdapat ruang kosong di bawah footer

1. Header.php

```
> buku
> dashboard
> errors
> login
> mahasiswa
> peminjaman
└─ templates
    └─ footer.php
    └─ header.php
> user
< index.html
└─ welcome_message.php
.htaccess
index.html
assets
system
uploads
user_guide
.editorconfig
.gitignore
.htaccess
composer.json
contributing.md
index.php
license.txt
readme.rst

16 <style>
17     .content {
18         min-height: calc(100vh - 50px);
19     }
20
21     footer {
22         height: 50px;
23     }
24 </style>
25
26 </head>
27
28 <body>
29     <!-- content -->
30     <div class="content">
31
32         <!-- nav -->
```

2. Footer.php

```
header.php  footer.php ×
application > views > templates > footer.php > footer.bg-dark.p-3
1  </div>
2  <!-- end content -->
3
4  <!-- footer -->
5  <footer class="bg-dark p-3">
```

- Semua pengambilan data ditampilkan diurutkan berdasarkan id yang paling kecil ke besar atau data paling lama ke data paling baru. Seharusnya datanya berurut dari id paling besar ke id paling kecil, sehingga data yang ditampilkan pertama adalah data paling baru

1. Buku_model.php

```
> libraries
> logs
< models
  Buku_model.php
  index.html
  Mahasiswa_model.php
  Peminjaman_model.php
  User_model.php
> third_party
> views
    7   public function get()
    8 {
    9     $this->db->order_by('id_buku', 'DESC');
   10    return $this->db->get('buku')->result();
   11 }
```

2. Mahasiswa_model.php

```
> logs
< models
  Buku_model.php
  index.html
  Mahasiswa_model.php
  Peminjaman_model.php
  User_model.php
> third_party
> views
    6   public function get()
    7 {
    8     $this->db->order_by('id_mahasiswa', 'DESC');
    9    return $this->db->get('mahasiswa')->result();
   10 }
```

3. Peminjaman_model.php

```
> config
> controllers
> core
> helpers
> hooks
> language
> libraries
> logs
< models
  Buku_model.php
  index.html
  Mahasiswa_model.php
  Peminjaman_model.php
  User_model.php
> third_party
> views
  .htaccess
  index.html
> assets
> system
> uploads
> user_guide
  .editorconfig
  .gitignore
  .htaccess
  composer.json
  contributing.md
  index.php
    36  public function get()
    37  {
    38    /*
    39      relasi tabel ==> hubungan antara tabel
    40      sehingga kita bisa menggunakan data buku dan mahasiswa pada peminjaman
    41      join ==> (nama tabel yang ingin di join, apa yang ingin di join)
    42      kita tidak bisa menulis id_mahasiswa = id_mahasiswa,
    43      karena sistem akan bingung ada 2 ada id_mahasiswa
    44      jadi dijelaskan id_mahasiswa pada tabel mahasiswa = id_mahasiswa pada tabel peminjaman
    45    */
    46    $this->db->join('mahasiswa', 'mahasiswa.id_mahasiswa = peminjaman.id_mahasiswa');
    47    $this->db->join('buku', 'buku.id_buku = peminjaman.id_buku');
    48    $this->db->order_by('id_peminjaman', 'DESC');
    49    return $this->db->get('peminjaman')->result();
   50 }
```

4. User_model.php

```
> logs
< models
  Buku_model.php
  index.html
  Mahasiswa_model.php
  Peminjaman_model.php
  User_model.php
> third_party
> views
  .htaccess
    6   public function get()
    7 {
    8     $this->db->order_by('id_user', 'DESC');
    9    return $this->db->get('user')->result();
   10 }
```

- Logout seharusnya sebelum logout, terdapat konfirmasi
- Header.php

```
<a class="nav-item nav-link text-white" href="= base_url('login/logout'); ?" onclick="return confirm('Anda Yakin Ingin logout?') >Logout</a>
```

- Halaman login masih bisa diakses oleh user yang telah login, seharusnya ketika user telah login dan mengakses halaman login dia akan dialihkan ke dashboard
- Login.php



The screenshot shows a code editor with the file `Login.php` open. The code is as follows:

```

public function index()
{
    $this->form_validation->set_rules('username', 'username', 'required|trim');
    $this->form_validation->set_rules('password', 'password', 'required|trim');
    $this->form_validation->set_error_delimiters('<div class="invalid-feedback text-capitalize">', '</div>');

    if ($this->form_validation->run() == false) {
        $data['title'] = 'Login';

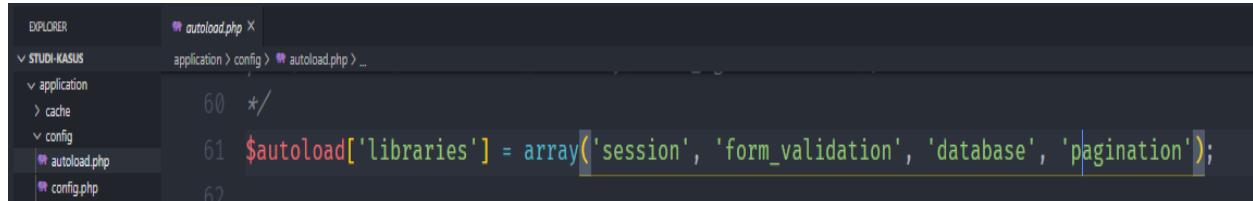
        if ($this->session->userdata('id_user')) {
            redirect('dashboard');
        }
    }
}

```

The code defines a public `index()` method. It sets validation rules for 'username' and 'password'. If validation fails, it sets the title to 'Login' and checks if a user is logged in. If a user is logged in, it redirects to the 'dashboard'.

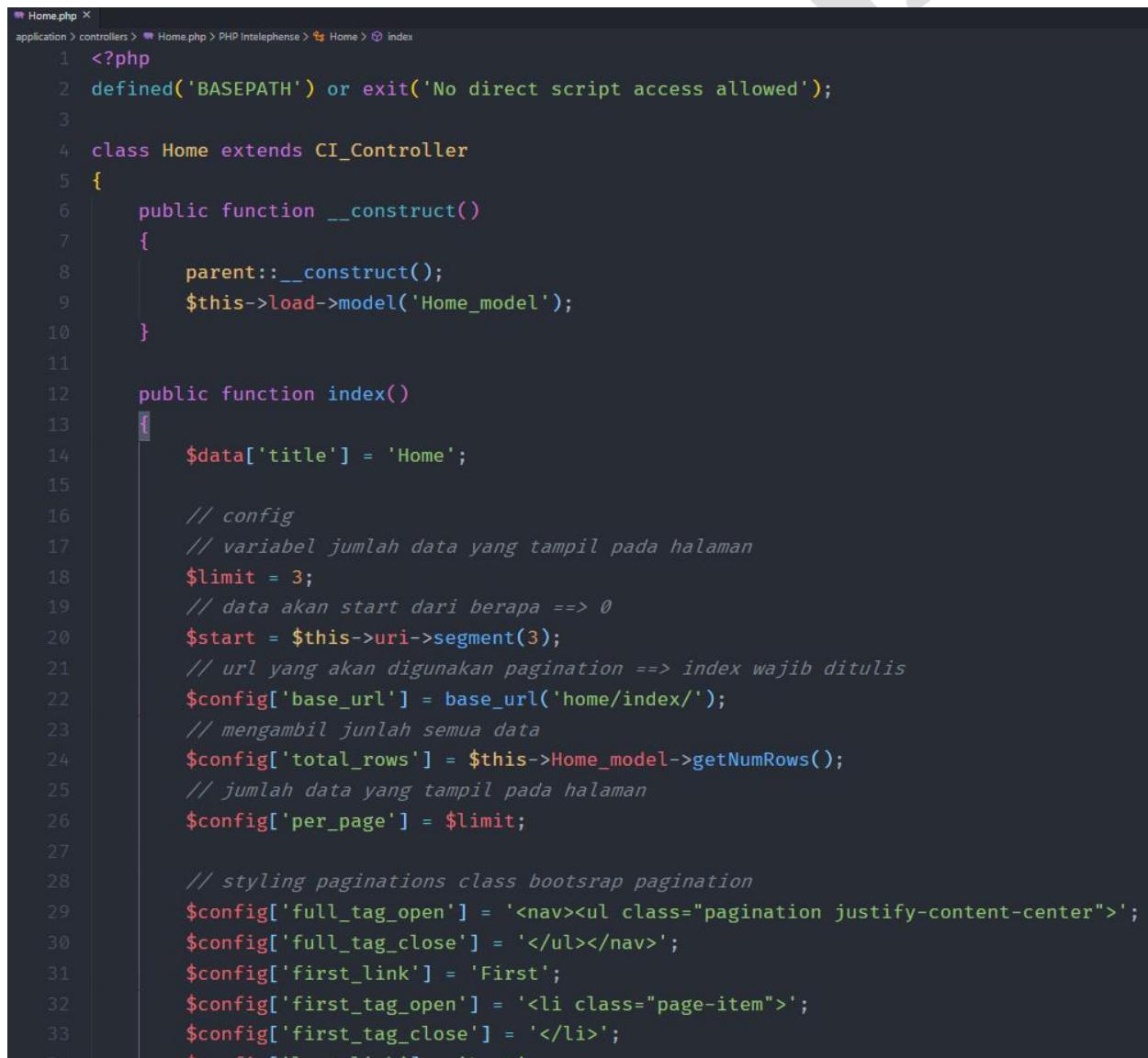
HOME

1. Halaman home adalah halaman yang akan di akses oleh user yang tidak melakukan login
2. Untuk dapat menggunakan pagination kita perlu meload library pagination



```
autoload.php
application > config > autoload.php
60 */
61 $autoload['libraries'] = array('session', 'form_validation', 'database', 'pagination');
62
```

3. Controller home



```
Home.php
application > controllers > Home.php > PHP Intelephense > Home > index
1 <?php
2 defined('BASEPATH') or exit('No direct script access allowed');
3
4 class Home extends CI_Controller
5 {
6     public function __construct()
7     {
8         parent::__construct();
9         $this->load->model('Home_model');
10    }
11
12    public function index()
13    {
14        $data['title'] = 'Home';
15
16        // config
17        // variabel jumlah data yang tampil pada halaman
18        $limit = 3;
19        // data akan start dari berapa ==> 0
20        $start = $this->uri->segment(3);
21        // url yang akan digunakan pagination ==> index wajib ditulis
22        $config['base_url'] = base_url('home/index/');
23        // mengambil jumlah semua data
24        $config['total_rows'] = $this->Home_model->getNumRows();
25        // jumlah data yang tampil pada halaman
26        $config['per_page'] = $limit;
27
28        // styling paginations class bootstrap pagination
29        $config['full_tag_open'] = '<nav><ul class="pagination justify-content-center">';
30        $config['full_tag_close'] = '</ul></nav>';
31        $config['first_link'] = 'First';
32        $config['first_tag_open'] = '<li class="page-item">';
33        $config['first_tag_close'] = '</li>';
```

```
34     $config['last_link'] = 'Last';
35     $config['last_tag_open'] = '<li class="page-item">';
36     $config['last_tag_close'] = '</li>';
37     $config['next_link'] = 'Next';
38     $config['next_tag_open'] = '<li class="page-item">';
39     $config['next_tag_close'] = '</li>';
40     $config['prev_link'] = 'Previous';
41     $config['prev_tag_open'] = '<li class="page-item">';
42     $config['prev_tag_close'] = '</li>';
43     $config['cur_tag_open'] = '<li class="page-item active"><a class="page-link" href="#">';
44     $config['cur_tag_close'] = '</a></li>';
45     $config['attributes'] = ['class' => 'page-link'];
46
47     // menyiapkan config
48     $this->pagination->initialize($config);
49     // membuat link pagination
50     $data['halaman'] = $this->pagination->create_links();
51     // data buku dimulai dari data ke berapa dan dilimit berapa data
52     $data['buku'] = $this->Home_model->getPagination($limit, $start);
53
54     // karena halaman user yang tidak login hanya 1 kita tida perlu membuat template
55     $this->load->view('home/index', $data);
56 }
57
58 function detail()
59 {
60     $id_buku = $this->uri->segment(3);
61     $data['detailbuku'] = $this->Home_model->getRow($id_buku);
62     $this->load->view('home/detail', $data);
63 }
64 }
```

4. Home_model.php

```
Home_model.php ×
application > models > Home_model.php > PHP Intelephense > Home_model > getPagination
1 <?php
2
3 class Home_model extends CI_Model
4 {
5     function getNumRows()
6     {
7         $this->db->order_by('id_buku', "DESC");
8         // num_rows() ==> akan mengembalikan berapa banyak jumlah baris data
9         return $this->db->get('buku')->num_rows();
10    }
11
12    function getPagination($limit, $start)
13    {
14        $this->db->order_by('id_buku', "DESC");
15        // ini akan mengembalikan data buku dengan jumlah data = $limit
16        return $this->db->get('buku', $limit, $start)->result();
17    }
18
19    public function.getRow($id_buku)
20    {
21        $where = ['id_buku' => $id_buku];
22        return $this->db->get_where('buku', $where)->row();
23    }
24 }
```

5. View home index

```
index.php ×
application > views > home > index.php > head
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title><?= $title; ?></title>
8     <link rel="stylesheet" href="<?= base_url('assets/bootstrap-4.4.1-dist/css/bootstrap.css'); ?>">
9
```

```
10 <style>
11     .content {
12         min-height: calc(100vh - 50px);
13     }
14
15     footer {
16         height: 50px;
17     }
18 </style>
19 </head>
20
21 <body>
22     <!-- content -->
23     <div class="content">
24         <div class="container">
25
26             <h1 class="my-5 display-3 text-center">SI PERPUSTAKAAN</h1>
27
28             <div class="row justify-content-center">
29
30                 <?php foreach ($buku as $b) : ?>
31                 <div class="col-md-4 col-sm-6 mb-3">
32
33                     <div class="card">
34                         
35                         <div class="card-body">
36                             <p class="card-title"><?= $b->judul; ?></p>
37                             <div class="card-text">Stock :
38                                 <?php if ($b->jumlah == 0) : ?>
39                                     <p class="badge badge-danger">Out Of Stock</p>
40                                 <?php else : ?>
41                                     <p class="badge badge-primary"><?= $b->jumlah; ?></p>
42                                 <?php endif ?>
43                                 <a href="#" class="btn btn-outline-success float-right get-detail-buku"
44                                     data-toggle="modal" data-target="#exampleModal" data-id="<?= $b->id_buku; ?>">Detail</a>
45                         </div>
46                     </div>
47                 </div>
48             <?php endforeach; ?>
49
50         </div>
51         <?= $halaman; ?>
52     </div>
53     </div>
54 </div>
55 <!-- end content -->
```

```
58    <!-- footer -->
59    <footer class="bg-dark p-3">
60        <div class="text-center text-light">
61            <h6>Copyright &copy; Tamus Tahir 2020</h6>
62        </div>
63    </footer>
64    <!-- end footer -->
65
66    <!-- modal detail buku -->
67    <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
aria-hidden="true">
68        <div class="modal-dialog modal-xl" role="document">
69            <div class="modal-content">
70                <div class="modal-header">
71                    <h5 class="modal-title" id="exampleModalLabel">Detail Buku</h5>
72                    <button type="button" class="close" data-dismiss="modal" aria-label="Close">
73                        <span aria-hidden="true">&times;</span>
74                    </button>
75                </div>
76                <div class="modal-body detail-buku">
77                    ...
78                </div>
79                <div class="modal-footer">
80                    <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
81                </div>
82            </div>
83        </div>
84    <!-- end modal detail buku -->
85
86    <script src=<?=& base_url('assets/jquery-3.4.1/jquery-3.4.1.min.js'); ?>"></script>
87    <script src=<?=& base_url('assets/bootstrap-4.4.1-dist/js/bootstrap.js'); ?>"></script>
88
89
90    <script>
91        $(document).ready(function() {
92            $('.get-detail-buku').on('click', function() {
93                const id = $(this).data('id');
94                $(".detail-buku").load("<?=& base_url('home/detail/'); ?>" + id);
95            });
96        });
97    </script>
98 </body>
99
100 </html>
```

6. View home detail

```
detail.php x
application > views > home > detail.php > div.container
1 <div class="container">
2   <div class="row">
3     <div class="col-md-4">
4       
5     </div>
6     <div class="col-md-8">
7       <h4><?= $detailbuku->judul; ?></h4>
8       <h5>Jumlah : <?= $detailbuku->jumlah; ?> buku</h5>
9       <p><?= $detailbuku->deskripsi; ?></p>
10      </div>
11    </div>
12 </div>
```

7. Karena kita mempunyai halaman yang bisa di akses oleh user yang tidak login yaitu halaman home sebaiknya controller default dipindahkan ke home

```
migration.php
mimes.php
profiler.php
routes.php
smileys.php
user_agents.php
controllers
51  */
52 $route['default_controller'] = 'home';
53 $route['404_override'] = '';
54 $route['translate_uri_dashes'] = FALSE;
```

8. Controller default sudah pindah ke home, kita juga harus merubah nav pada admin, yaitu pada link dashboard

```
<!-- nav -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark py-3">
  <div class="container">
    <a class="navbar-brand font-weight-bold" href="= base_url('dashboard'); ?&gt;"PERPUSTAKAAN</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup"
           aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
      <div class="navbar-nav ml-auto">
        <a class="nav-item nav-link text-white" href="= base_url('dashboard'); ?&gt;"Dashboard</a>
```

9. Browser

