

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD dan Login pada Codeigniter

Oleh:

M. Adi Syahputra

NIM. 2210817210017

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
JUNI 2024**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN WEB II
MODUL 6

Laporan Praktikum Pemrograman Web II Modul 6 : CRUD dan Login pada Codeigniter ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Web II. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : M. Adi Syahputra
NIM : 2210817210017

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

Ir. Eka Setya Wijaya, S.T., M.Kom.
NIP. 198205082008011010

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code	7
B. Output Program.....	33
C. Pembahasan	36
D. Tautan Git	49

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	33
Gambar 2. Screenshot Hasil Jawaban Soal 1	34
Gambar 3. Screenshot Hasil Jawaban Soal 1	34
Gambar 4. Screenshot Hasil Jawaban Soal 1	35
Gambar 5. Screenshot Hasil Jawaban Soal 1	35

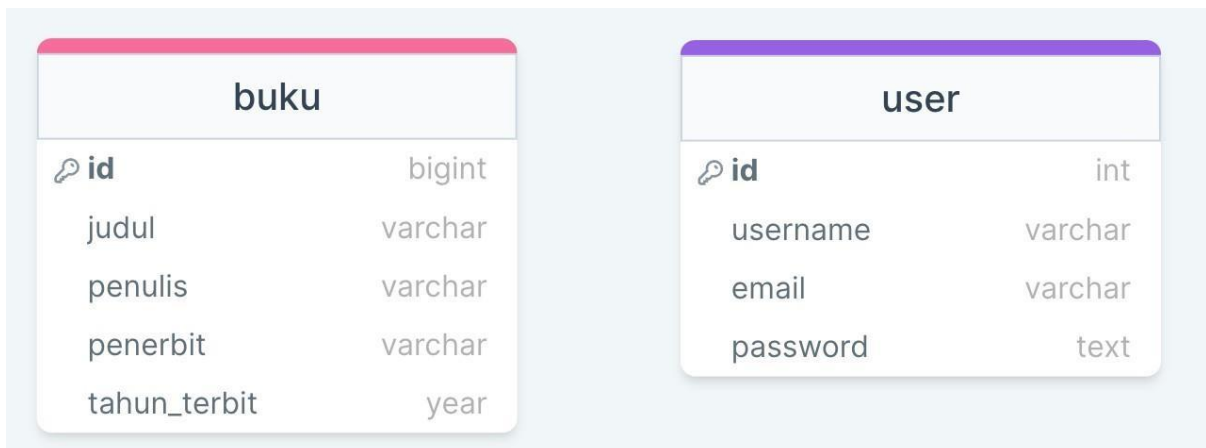
DAFTAR TABEL

Table 1. Source Code Soal 1	7
Table 2. Source Code Soal 1	11
Table 3. Source Code Soal 1	11
Table 4. Source Code Soal	12
Table 5. Source Code Soal 1	14
Table 6. Source Code Soal 1	15
Table 7. Source Code Soal 1	17
Table 8. Source Code Soal 1	20
Table 9. Source Code Soal 1	22
Table 10. Source Code Soal 1	25
Table 11. Source Code Soal 1	27
Table 12. Source Code Soal 1	29
Table 13. Source Code Soal 1	29
Table 14. Source Code Soal 1	33

SOAL 1

Buatlah sebuah website yang memiliki fitur login dan logout. Setelah login pengguna website dapat melakukan CRUD yang terhubung langsung dengan database. Beberapa syarat yang harus ada pada website :

- Jika seorang pengguna belum login dan mengakses halaman yang mengharuskan login, maka tampilkan sebuah pesan peringatan di halaman login, peringatan berisi “Login terlebih dahulu!”.
- Berikan validasi pada bagian ‘Create’ atau ‘tambah data’ di form, sebagai berikut:
 - Judul : Harus diisi dan berupa string
 - penulis : Harus diisi dan berupa string
 - penerbit : Harus diisi dan berupa string
 - tahun_terbit : Harus diisi dan berupa angka, angka harus lebih besar dari 1800 dan lebih kecil dari 2024
- (opsional) buatlah custom pesan validasi menggunakan bahasa indonesia
- Buat file migration pada codeigniter dengan Desain konseptual database sebagai



Simpan dengan nama projek di Github: PRAK601

A. Source Code

Routes.php

1	<?php
2	
3	use CodeIgniter\Router\RouteCollection;
4	
5	/**
6	* @var RouteCollection \$routes
7	*/
8	\$routes->setAutoRoute(true);
9	\$routes->get('/', 'Buku::login');
10	\$routes->get('home', 'Buku::index');
11	\$routes->get('/register', 'Register::index');
12	\$routes->post('/register/process', 'Register::process');
13	\$routes->get('/login', 'Login::index');
14	\$routes->post('/login/process', 'Login::process');
15	\$routes->get('/logout', 'Login::logout');
16	\$routes->get('/buku/create', 'Buku::create');
17	\$routes->post('buku/store', 'Buku::store');
18	\$routes->post('buku/update/', 'Buku::update', ['filter'
19	=> 'auth']);
20	\$routes->get('buku/edit/', 'Buku::edit', ['filter' =>
	'auth']);
	\$routes->get('buku/delete/', 'Buku::delete', ['filter'
	=> 'auth']);

Table 1. Source Code Soal 1

Controllers Buku.php

1	<?php namespace App\Controllers;
2	
3	use App\Models\BukuModel;
4	use CodeIgniter\Controller;
5	use App\Models\PostModel;
6	
7	class Buku extends Controller
8	{
9	/**
10	*
11	* @var Model
12	*/
13	protected \$model;
14	

```
15 public function __construct()
16 {
17     $this->model = new BukuModel();
18     $this->helpers = ['form', 'url'];
19
20 }
21
22 public function login()
23 {
24     $data = [
25         'posts' => $this->model->paginate(10),
26         'pager' => $this->model->pager,
27         'title' => 'POST LIST'
28     ];
29
30     return view('vw_login', $data);
31 }
32
33 public function index()
34 {
35     if (!session()->get('logged_in')) {
36         return redirect()->to(base_url('/login'));
37     }
38     $data = [
39         'posts' => $this->model->paginate(10),
40         'pager' => $this->model->pager,
41         'title' => 'POST LIST'
42     ];
43
44     return view('vw_buku', $data);
45 }
46
47 public function create()
48 {
49     // Check if user is authenticated
50     if (!session()->get('logged_in')) {
51         return redirect()->to(base_url('/login'));
52     }
53     $data = ['title' => 'Create new post'];
54
55     return view('vw_add', $data);
56 }
57
58 public function store()
59 {
60     $data = $this->request->getPost(['judul',
61     'penulis', 'penerbit', 'tahun_terbit']);
```



```

62
63         if (! $this->validateData($data, $this->model-
64 >validationRules)) {
65             return $this->create();
66         }
67
68         $post = $this->validator->getValidated();
69
70         $save = $this->model->save([
71             'judul' => $post['judul'],
72             'penulis' => $post['penulis'],
73             'penerbit' => $post['penerbit'],
74             'tahun_terbit' => $post['tahun_terbit'],
75         ]);
76
77         session()->setFlashdata('success', 'Post has
78 been added successfully.');
```

```

79         return redirect()->to(base_url('home'));
80     }
81
82     public function edit($id)
83     {
84         // Check if user is authenticated
85         if (!session()->get('logged_in')) {
86             return redirect()->to(base_url('/login'));
87         }
88         $post = $this->model->find($id);
89         if (empty($post)) {
90             session()->setFlashdata('error', 'Post not
91 found');
```

```

92             return redirect()->back();
93         }
94         $data = [
95             'title' => 'Edit Post',
96             'post' => $post
97         ];
98
99         return view('vw_edit', $data);
100     }
101
102     public function update($id)
103     {
104         // Check if user is authenticated
105         if (!session()->get('logged_in')) {
106             return redirect()->to(base_url('/login'));
107         }
108         $post = $this->model->find($id);

```

```

109
110         if (empty($post)) {
111             session()->setFlashdata('error', 'Post      not
112 found');
113             return redirect()->back();
114         }
115
116         $data      =      $this->request->getPost(['judul',
117 'penulis', 'penerbit', 'tahun_terbit']);
118
119         if (! $this->validateData($data, $this->model-
120 >validationRules)) {
121             return $this->create();
122         }
123
124         $updatedPost = $this->validator->getValidated();
125
126         $update      =      $this->model->update($post['id'],
127 $updatedPost);
128
129         if ($update) {
130             session()->setFlashdata('success', 'Post has
131 been updated successfully');
132             return redirect()->to(base_url('home'));
133         } else {
134             session()->setFlashdata('error', 'Some
135 problems occured, please try again. ');
136             return redirect()->back();
137         }
138     }
139
140     public function delete($id)
141     {
142         // Check if user is authenticated
143         if (!session()->get('logged_in')) {
144             return redirect()->to(base_url('home'));
145         }
146
147         if (empty($id)) {
148             return redirect()->to(base_url('home'));
149         }
150
151         $delete = $this->model->delete($id);
152
153         if ($delete) {
154             session()->setFlashdata('success', 'Post has
155 been removed successfully. ');

```

	<pre> return redirect()->to(base_url('home')); } else { session()->setFlashdata('error', 'Some problems occured, please try again.');</pre> <pre> return redirect()->to(base_url('home')); } } } </pre>
--	---

Table 2. Source Code Soal 1

Home.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	class Home extends BaseController
6	{
7	public function index(): string
8	{
9	return view('vw_buku');
10	}
11	}

Table 3. Source Code Soal 1

Login.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Models\UsersModel;
6	
7	class Login extends BaseController
8	{
9	public function index()
10	{
11	return view('vw_login');
12	}
13	
14	public function process()
15	{

16	\$users = new UsersModel();
17	\$username = \$this->request->getVar('username');
18	\$password = \$this->request->getVar('password');
19	\$dataUser = \$users->where('username',
20	\$username)->first();
21	
22	if (\$dataUser) {
23	// Check the password using object property
24	syntax
25	if (password_verify(\$password, \$dataUser->
26	password)) {
27	session()->set([
28	'username' => \$dataUser->username,
29	'logged_in' => true
30]);
31	return redirect()->to(base_url('home'));
32	} else {
33	session()->setFlashdata('error',
34	'Username & Password Salah');
35	return redirect()->back();
36	}
37	} else {
38	session()->setFlashdata('error', 'Username &
39	Password Salah');
40	return redirect()->back();
41	}
42	}
43	
44	public function logout()
	{
	session()->destroy();
	return redirect()->to('/login');
	}
	}

Table 4. Source Code Soal

Register.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Models\UsersModel;
6	
7	class Register extends BaseController

```

8 {
9     public function index()
10     {
11         return view('vw_register');
12     }
13
14     public function process()
15     {
16         if (!$this->validate([
17             'username' => [
18                 'rules' =>
19 'required|min_length[4]|max_length[20]|is_unique[users.
20 username]',
21                 'errors' => [
22                     'required' => '{field} Harus diisi',
23                     'min_length' => '{field} Minimal 4
24 Karakter',
25                     'max_length' => '{field} Maksimal 20
26 Karakter',
27                     'is_unique' => 'Username sudah
28 digunakan sebelumnya'
29                 ]
30             ],
31             'password' => [
32                 'rules' =>
33 'required|min_length[4]|max_length[50]',
34                 'errors' => [
35                     'required' => '{field} Harus diisi',
36                     'min_length' => '{field} Minimal 4
37 Karakter',
38                     'max_length' => '{field} Maksimal 50
39 Karakter',
40                 ]
41             ],
42             'password_conf' => [
43                 'rules' => 'matches[password]',
44                 'errors' => [
45                     'matches' => 'Konfirmasi Password
46 tidak sesuai dengan password',
47                 ]
48             ],
49             'email' => [
50                 'rules' =>
51 'required|min_length[4]|max_length[100]|valid_email',
52                 'errors' => [
53                     'required' => '{field} Harus diisi',
54

```

55	<code>'min_length' => '{field} Minimal 4</code>
56	<code>Karakter',</code>
57	<code>'max_length' => '{field} Maksimal</code>
58	<code>100 Karakter',</code>
59	<code>'valid_email' => 'Email tidak valid'</code>
60	<code>]</code>
61	<code>],</code>
62	<code>))) {</code>
63	<code>session()->setFlashdata('error', \$this-</code>
64	<code>>validator->listErrors());</code>
65	<code>return redirect()->back()->withInput();</code>
66	<code>}</code>
	<code>\$users = new UsersModel();</code>
	<code>\$hashedPassword = password_hash(\$this->request-</code>
	<code>>getVar('password'), PASSWORD_DEFAULT);</code>
	<code>\$users->insert([</code>
	<code> 'username' => \$this->request-</code>
	<code>>getVar('username'),</code>
	<code> 'password' => \$hashedPassword,</code>
	<code> 'email' => \$this->request->getVar('email')</code>
	<code>]);</code>
	<code>return redirect()->to('/login');</code>
	<code>}</code>
	<code>}</code>

Table 5. Source Code Soal 1

AuthFilter.php

1	<code><?php</code>
2	
3	<code>namespace App\Filters;</code>
4	
5	<code>use CodeIgniter\HTTP\RequestInterface;</code>
6	<code>use CodeIgniter\HTTP\ResponseInterface;</code>
7	<code>use CodeIgniter\Filters\FilterInterface;</code>
8	
9	<code>class AuthFilter implements FilterInterface</code>
10	<code>{</code>
11	<code> public function before(RequestInterface \$request,</code>
12	<code> \$args = null)</code>
13	<code> {</code>

14	if (!session()->has('user_id')) {
15	return redirect()->to('/login');
16	}
17	}
18	
19	public function after(RequestInterface \$request,
20	ResponseInterface \$response, \$arguments = null)
21	{
	}
	}

Table 6. Source Code Soal 1

vw_add.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	
5	<div class="container mt-5">
6	<div class="row">
7	<div class="col-md-12">
8	<div class="card">
9	<div class="card-header">
10	<?= \$title ?>
11	</div>
12	<div class="card-body">
13	
14	<?php if (session()-
15	>getFlashdata('error')) : ?>
16	<div class="alert alert-danger">
17	<?php echo session()-
18	>getFlashdata('error'); ?>
19	</div>
20	<?php endif; ?>
21	
22	<?= validation_list_errors() ?>
23	
24	<?= form_open('buku/store'); ?>
25	<div class="form-group">
26	<label>Judul</label>
27	<input type="text"
28	class="form-control" name="judul" placeholder="Masukkan
29	Judul">
30	</div>
31	<div class="form-group">

```

32         <label>Penulis</label>
33         <input                type="text"
34 class="form-control"          name="penulis"
35 placeholder="Masukkan Penulis">
36     </div>
37     <div class="form-group">
38         <label>Penerbit</label>
39         <input                type="text"
40 class="form-control"          name="penerbit"
41 placeholder="Masukkan Penerbit">
42     </div>
43     <div class="form-group">
44         <label>Tahun Terbit</label>
45         <input                type="number"
46 class="form-control"          name="tahun_terbit"
47 placeholder="Masukkan Tahun Terbit">
48     </div>
49     <div class="form-group">
50         <button                class="btn    btn-
51 primary">Save</button>
52         <a href="<?= base_url('/') ?>"
53 class="btn btn-link">Back</a>
54     </div>
55     <?= form_close(); ?>
56 </div>
57
58 </div>
59 </div>
60 </div>
61 </div>
62
63 <?= $this->endSection() ?>
64
65 <?= $this->section('extra-js') ?>
66 <!-- include summernote css/js -->
67 <link
68 href="https://cdn.jsdelivr.net/npm/summernote@0.8.15/di
69 st/summernote-bs4.min.css" rel="stylesheet">
70 <script
71 src="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dis
72 t/summernote-bs4.min.js"></script>
73 <script>
74     $(document).ready(function() {
75         $('#post_content').summernote({
76             tabsize: 2,
77             height: 500
78         });
79     });

```


	<pre> }) </script> <?= \$this->endSection() ?> </pre>
--	--

Table 7. Source Code Soal 1

vw_buku.php

1	<pre><?= \$this->extend('layouts/main') ?></pre>
2	
3	<pre><?= \$this->section('content') ?></pre>
4	<pre><style></pre>
5	<pre> body {</pre>
6	<pre> background-image:</pre>
7	<pre>url('https://images.pexels.com/photos/2041540/pexels-</pre>
8	<pre>photo-</pre>
9	<pre>2041540.jpeg?auto=compress&cs=tinysrgb&h=750&w=1260');</pre>
10	<pre> background-repeat: no-repeat;</pre>
11	<pre> background-attachment: fixed;</pre>
12	<pre> background-position: center;</pre>
13	<pre> background-size: cover;</pre>
14	<pre> }</pre>
15	<pre></style></pre>
16	
17	<pre><!-- Navbar --></pre>
18	<pre><nav class="navbar navbar-expand-lg navbar-light bg-</pre>
19	<pre>light"></pre>
20	<pre> Data Buku</pre>
21	<pre> <div class="collapse navbar-collapse"</pre>
22	<pre>id="navbarNav"></pre>
23	<pre> <ul class="navbar-nav ml-auto"></pre>
24	<pre> <?php if (session()->get('logged_in')) : ?></pre>
25	<pre> <li class="nav-item"></pre>
26	<pre> <a class="nav-link"</pre>
27	<pre>href="#">Welcome, <?php echo session()-</pre>
28	<pre>>get('username'); ?></pre>
29	<pre> </pre>
30	<pre> <li class="nav-item"></pre>
31	<pre> <a class="nav-link" href="<?php</pre>
32	<pre>echo base_url('login/logout'); ?>">Logout</pre>
33	<pre> </pre>
34	<pre> <?php else : ?></pre>
35	<pre> <li class="nav-item"></pre>
36	<pre> <a class="nav-link" href="<?php</pre>
37	<pre>echo base_url('login'); ?>">Login</pre>
38	<pre> </pre>

```

39         <li class="nav-item">
40             <a class="nav-link" href="<?php
41 echo base_url('register'); ?>">Register</a>
42         </li>
43     <?php endif; ?>
44 </ul>
45 </div>
46 </nav>
47
48 <div class="container mt-5">
49     <div class="row">
50         <div class="col-md-12">
51             <div class="card">
52                 <div class="card-header">
53                     <a href="<?php echo
54 base_url('buku/create'); ?>" class="btn btn-primary
55 btn-sm float-right">Insert Data</a>
56                 </div>
57                 <div class="card-body">
58
59                     <?php if (session() -
60 >getFlashdata('success')) { ?>
61                         <div class="alert alert-
62 success">
63                             <?php echo session() -
64 >getFlashdata('success'); ?>
65                         </div>
66                     <?php } ?>
67
68                     <?php if (session() -
69 >getFlashdata('error')) { ?>
70                         <div class="alert alert-
71 danger">
72                             <?php echo session() -
73 >getFlashdata('error'); ?>
74                         </div>
75                     <?php } ?>
76
77                     <table class="table table-
78 bordered">
79                         <thead class="text-center">
80                             <tr>
81                                 <th
82 scope="col">Judul</th>
83                                 <th
84 scope="col">Penulis</th>
85

```

```

86         <th
87         scope="col">Penerbit</th>
88         <th      scope="col">Tahun
89         Terbit</th>
90         <th
91         scope="col">Action</th>
92         </tr>
93         </thead>
94         <tbody class="text-center">
95             <?php if (!empty($posts) &&
96             is_array($posts)) { ?>
97                 <?php foreach ($posts
98                 as $row) { ?>
99                     <tr>
100                         <td><?php echo
101                         $row['judul']; ?></td>
102                         <<td><?php echo
103                         $row['penulis']; ?></td>
104                         <td><?php echo
105                         $row['penerbit']; ?></td>
106                         <td><?php echo
107                         $row['tahun_terbit']; ?></td>
108                         <td>
109
110                             <form
111                             onsubmit="return confirm('Apakah Anda Yakin ?');"
112
113                             action="<?php      echo      base_url('buku/delete/'      .
114                             $row['id']); ?>" method="POST">
115
116                                 <input
117                                 type="hidden" name="{csrf_token}" value="{csrf_hash}">
118                                 <input
119                                 type="hidden" name="_method" value="DELETE">
120
121                                 <a
122                                 href="<?php echo base_url('buku/edit/' . $row['id']);
123                                 ?>" class="btn btn-primary btn-sm">Edit</a>
124
125                                 <button type="submit" class="btn btn-danger btn-
126                                 sm">Delete</button>
127
128                                 </form>
129                             </td>
130                     </tr>
131                 <?php } ?>

```

	<pre> <?php } else { ?> <tr> <td colspan="4" class="text-center">No post found.</td> </tr> <?php } ?> </tbody> </table> <?= \$pager->links(); ?> </div> </div> </div> </div> </div> <?= \$this->endSection() ?> <?= \$this->section('extra-js') ?> <script> \$(document).ready(function() { \$('.pagination li').addClass('page-item'); \$('.pagination li a').addClass('page-link'); }) </script> <?= \$this->endSection() ?> </pre>
--	---

Table 8. Source Code Soal 1

vm_edit.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	
5	<div class="container mt-5">
6	<div class="row">
7	<div class="col-md-12">
8	<div class="card">
9	<div class="card-header">
10	<?= \$title ?>
11	</div>
12	<div class="card-body">
13	

```

14         <?php         if         (session()-
15 >getFlashdata('error')) : ?>
16             <div class="alert alert-danger">
17                 <?php         echo         session()-
18 >getFlashdata('error'); ?>
19             </div>
20         <?php endif; ?>
21
22         <?= validation_list_errors() ?>
23
24         <?=         form_open('buku/update/'.
25 $post['id']); ?>
26         <input type="hidden" name="_method"
27 value="PUT">
28
29         <div class="form-group">
30             <label>Judul</label>
31             <input
32 class="form-control" name="judul" placeholder="Masukkan
33 Judul">
34         </div>
35         <div class="form-group">
36             <label>Penulis</label>
37             <input
38 class="form-control"
39 placeholder="Masukkan Penulis">
40         </div>
41         <div class="form-group">
42             <label>Penerbit</label>
43             <input
44 class="form-control"
45 placeholder="Masukkan Penerbit">
46         </div>
47         <div class="form-group">
48             <label>Tahun Terbit</label>
49             <input
50 class="form-control"
51 placeholder="Masukkan Tahun Terbit">
52         </div>
53         <div class="form-group">
54             <button class="btn btn-
55 primary">Update</button>
56             <a href="<?= base_url('/') ?>"
57 class="btn btn-link">Back</a>
58         </div>
59         <?= form_close(); ?>
60

```

61	</div>
62	
63	</div>
64	</div>
65	</div>
66	</div>
67	
68	<?= \$this->endSection() ?>
	<?= \$this->section('extra-js') ?>
	<!-- include summernote css/js -->
	<link
	href="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dist/summernote-bs4.min.css" rel="stylesheet">
	<script
	src="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dist/summernote-bs4.min.js"></script>
	<script>
	\$(document).ready(function() {
	\$('#post_content').summernote({
	tabsize: 2,
	height: 500
	});
	})
	</script>
	<?= \$this->endSection() ?>

Table 9. Source Code Soal 1

vm_login.php

1	<!doctype html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="utf-8">
6	<meta name="viewport" content="width=device-width,
7	initial-scale=1">
8	<meta name="description" content="">
9	<meta name="author" content="Mark Otto, Jacob
10	Thornton, and Bootstrap contributors">
11	<meta name="generator" content="Hugo 0.79.0">
12	<title>Login Page</title>
13	
14	<style>
15	body {

```
16         background-image:
17 url('https://images.pexels.com/photos/2041540/pexels-
18 photo-
19 2041540.jpeg?auto=compress&cs=tinysrgb&h=750&w=1260');
20         background-repeat: no-repeat;
21         background-attachment: fixed;
22         background-position: center;
23         background-size: cover;
24         display: flex;
25         justify-content: center;
26         align-items: center;
27         height: 100vh;
28         margin: 0;
29         font-family: Arial, sans-serif;
30     }
31
32     .form-signin {
33         width: 100%;
34         max-width: 330px;
35         padding: 15px;
36         background-color: rgba(255, 255, 255, 0.8);
37         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
38         border-radius: 8px;
39         text-align: center;
40     }
41
42     .form-signin h1 {
43         font-size: 1.5em;
44         margin-bottom: 1em;
45     }
46
47     .form-signin input {
48         width: calc(100% - 22px);
49         padding: 10px;
50         margin-bottom: 10px;
51         border: 1px solid #ccc;
52         border-radius: 4px;
53         margin-left: auto;
54         margin-right: auto;
55         display: block;
56     }
57
58     .form-signin button {
59         width: calc(100% - 22px);
60         padding: 10px;
61         border: 2px solid transparent;
62         border-radius: 4px;
```

```

63         background-color: transparent;
64         color: #007bff;
65         font-size: 1em;
66         cursor: pointer;
67         transition: color 0.3s;
68         margin-left: auto;
69         margin-right: auto;
70         display: block;
71     }
72
73     .form-signin button:hover {
74         color: #0056b3;
75     }
76
77     .alert {
78         padding: 10px;
79         background-color: #f9edbe;
80         color: #856404;
81         border: 1px solid #ffeeba;
82         border-radius: 4px;
83         margin-bottom: 15px;
84     }
85
86     .text-muted {
87         color: #6c757d;
88     }
89 </style>
90 </head>
91
92 <body>
93
94     <main class="form-signin">
95     <div class="alert" id="initial-alert">
96         Login terlebih dahulu!
97         <?php if (!empty(session()-
98 >getFlashdata('error'))): ?>
99             <div class="alert alert-warning alert-
100 dismissible fade show" role="alert">
101                 <?php echo session()-
102 >getFlashdata('error'); ?>
103             </div>
104             <?php endif; ?>
105             <form method="post" action="<?= base_url();
106 ?>/login/process">
107                 <?= csrf_field(); ?>
108                 <h1 class="h3 mb-3 fw-normal">Login</h1>
109

```


110	<code><input type="text" name="username"</code>
111	<code>id="username" placeholder="Username" required</code>
112	<code>autofocus></code>
113	<code><input type="password" name="password"</code>
114	<code>id="password" placeholder="Password" required></code>
115	<code><button type="submit">Login</button></code>
	<code><p class="mt-5 mb-3 text-muted">&copy;</code>
	<code>Warung Belajar</p></code>
	<code></form></code>
	<code></main></code>
	<code><script></code>
	<code> window.onload = function() {</code>
	<code> document.getElementById('initial-</code>
	<code>alert').style.display = 'block';</code>
	<code> }</code>
	<code></script></code>
	<code></body></code>
	<code></html></code>

Table 10. Source Code Soal 1

vm_register.php

1	<code><!doctype html></code>
2	<code><html lang="en" class="h-100"></code>
3	
4	<code><head></code>
5	<code> <meta charset="utf-8"></code>
6	<code> <meta name="viewport" content="width=device-width,</code>
7	<code>initial-scale=1"></code>
8	<code> <meta name="description" content=""></code>
9	<code> <meta name="author" content="Mark Otto, Jacob</code>
10	<code>Thornton, and Bootstrap contributors"></code>
11	<code> <meta name="generator" content="Hugo 0.79.0"></code>
12	<code> <title>Sticky Footer Template · Bootstrap</code>
13	<code>v5.0</title></code>
14	
15	<code> <link rel="canonical"</code>
16	<code>href="https://getbootstrap.com/docs/5.0/examples/sticky</code>
17	<code>-footer/"></code>
18	
19	<code> <!-- Bootstrap core CSS --></code>
20	<code> <link href="<?= base_url('css/bootstrap.min.css')</code>
21	<code>?>" rel="stylesheet"></code>

```

22
23     <style>
24         .bd-placeholder-img {
25             font-size: 1.125rem;
26             text-anchor: middle;
27             -webkit-user-select: none;
28             -moz-user-select: none;
29             user-select: none;
30         }
31
32         @media (min-width: 768px) {
33             .bd-placeholder-img-lg {
34                 font-size: 3.5rem;
35             }
36         }
37     </style>
38
39     <!-- Custom styles for this template -->
40     <link href="<? = base_url('css/sticky-footer.css')
41     ?>" rel="stylesheet">
42 </head>
43
44 <body class="d-flex flex-column h-100">
45
46     <!-- Begin page content -->
47     <main class="flex-shrink-0">
48         <div class="container">
49             <h1 class="mt-5">Register Form</h1>
50             Silahkan Daftarkan Identitas Anda
51             <hr />
52             <?php if (!empty(session() -
53 >getFlashdata('error')) : ?>
54                 <div class="alert alert-warning alert-
55 dismissible fade show" role="alert">
56                     <h4>Periksa Entrian Form</h4>
57                     </hr />
58                     <?php echo session() -
59 >getFlashdata('error'); ?>
60                 </div>
61             <?php endif; ?>
62             <form method="post" action="<? = base_url();
63 ?>/register/process">
64                 <? = csrf_field(); ?>
65                 <div class="mb-3">
66                     <label for="username" class="form-
67 label">Username</label>
68

```

69	<code><input type="text" class="form-</code>
70	<code>control" id="username" name="username"></code>
71	<code></div></code>
72	<code><div class="mb-3"></code>
73	<code><label for="password" class="form-</code>
74	<code>label">Password</label></code>
75	<code><input type="password" class="form-</code>
76	<code>control" id="password" name="password"></code>
77	<code></div></code>
78	<code><div class="mb-3"></code>
79	<code><label for="password_conf"</code>
80	<code>class="form-label">Confirm Password</label></code>
81	<code><input type="password" class="form-</code>
82	<code>control" id="password_conf" name="password_conf"></code>
83	<code></div></code>
84	<code><div class="mb-3"></code>
85	<code><label for="email" class="form-</code>
86	<code>label">Email</label></code>
87	<code><input type="text" class="form-</code>
88	<code>control" id="email" name="email"></code>
89	<code></div></code>
90	<code><div class="mb-3"></code>
	<code><button type="submit" class="btn</code>
	<code>btn-primary">Register</button></code>
	<code></div></code>
	<code></form></code>
	<code><hr /></code>
	<code></div></code>
	<code></main></code>
	<code><footer class="footer mt-auto py-3 bg-light"></code>
	<code><div class="container"></code>
	<code>Place sticky footer</code>
	<code>content here.</code>
	<code></div></code>
	<code></footer></code>
	<code><!-- Bootstrap core JS --></code>
	<code><script src="<?=</code>
	<code>base_url('js/bootstrap.bundle.min.js') ?>"></script></code>
	<code></body></code>
	<code></html></code>

Table 11. Source Code Soal 1

BukuModel.php

```
1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  class BukuModel extends Model
8  {
9      protected $table = 'buku';
10     protected $allowedFields = ['judul', 'penulis' ,
11     'penerbit', 'tahun_terbit'];
12     protected $createdField = 'created_at';
13     protected $updatedField = 'updated_at';
14     protected $deletedField = 'deleted_at';
15
16     protected $validationRules = ([
17         'judul' => [
18             'rules' => 'required|string',
19             'errors' => [
20                 'required' => 'Judul harus diisi.',
21                 'string' => 'Judul harus berupa
22 string.'
23             ]
24         ],
25         'penulis' => [
26             'rules' => 'required|string',
27             'errors' => [
28                 'required' => 'Penulis harus diisi.',
29                 'string' => 'Penulis harus berupa
30 string.'
31             ]
32         ],
33         'penerbit' => [
34             'rules' => 'required|string',
35             'errors' => [
36                 'required' => 'Penerbit harus diisi.',
37                 'string' => 'Penerbit harus berupa
38 string.'
39             ]
40         ],
41         'tahun_terbit' => [
42             'rules' => 'required|integer|greater_than[1800]|less_than[2024]',
43             'errors' => [
44                 'required' => 'Tahun terbit harus
45 diisi.',
```

47	'integer' => 'Tahun terbit harus
48	berupa angka.',
49	'greater_than' => 'Tahun terbit harus
50	lebih besar dari 1800.',
	'less_than' => 'Tahun terbit harus
	lebih kecil dari 2024.'
]
],
]);
	protected \$skipValidation = false;
	}

Table 12. Source Code Soal 1

UsersModel.php

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class UsersModel extends Model
8	{
9	protected \$table = "users";
10	protected \$returnType = "object";
11	protected \$useTimestamps = true;
12	protected \$allowedFields = ['username', 'password',
13	'email'];
	}

Table 13. Source Code Soal 1

Data_buku.sql

1	-- phpMyAdmin SQL Dump
2	-- version 5.2.1
3	-- https://www.phpmyadmin.net/
4	--
5	-- Host: 127.0.0.1
6	-- Waktu pembuatan: 13 Jun 2024 pada 10.49
7	-- Versi server: 10.4.32-MariaDB
8	-- Versi PHP: 8.2.12

```

9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET
16 @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
17 /*!40101 SET
18 @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
19 /*!40101 SET
20 @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
21 /*!40101 SET NAMES utf8mb4 */;
22
23 --
24 -- Database: `data_buku`
25 --
26
27 -----
28 ----
29
30 --
31 -- Struktur dari tabel `buku`
32 --
33
34 CREATE TABLE `buku` (
35   `id` bigint(20) UNSIGNED NOT NULL,
36   `judul` varchar(255) NOT NULL,
37   `penulis` varchar(255) NOT NULL,
38   `penerbit` varchar(255) NOT NULL,
39   `tahun_terbit` year(4) DEFAULT NULL
40 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
41 COLLATE=utf8mb4_general_ci;
42
43 --
44 -- Dumping data untuk tabel `buku`
45 --
46
47 INSERT INTO `buku` (`id`, `judul`, `penulis`,
48 `penerbit`, `tahun_terbit`) VALUES
49 (2, 'Bumi Cinta', 'Habiburrahman El Shirazy',
50 'Republika', '2020');
51
52 -----
53 ----
54
55 --

```

```

56 -- Struktur dari tabel `migrations`
57 --
58
59 CREATE TABLE `migrations` (
60   `id` bigint(20) UNSIGNED NOT NULL,
61   `version` varchar(255) NOT NULL,
62   `class` varchar(255) NOT NULL,
63   `group` varchar(255) NOT NULL,
64   `namespace` varchar(255) NOT NULL,
65   `time` int(11) NOT NULL,
66   `batch` int(11) UNSIGNED NOT NULL
67 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
68 COLLATE=utf8mb4_general_ci;
69
70 --
71 -- Dumping data untuk tabel `migrations`
72 --
73
74 INSERT INTO `migrations` (`id`, `version`, `class`,
75   `group`, `namespace`, `time`, `batch`) VALUES
76   (1, '2024-06-03-103310',
77   'App\\Database\\Migrations\\Users', 'default', 'App',
78   1718259339, 1),
79   (2, '2024-06-03-122804',
80   'App\\Database\\Migrations\\Buku', 'default', 'App',
81   1718259339, 1);
82
83 -----
84 ----
85
86 --
87 -- Struktur dari tabel `users`
88 --
89
90 CREATE TABLE `users` (
91   `id` int(10) UNSIGNED NOT NULL,
92   `username` varchar(255) NOT NULL,
93   `email` varchar(255) NOT NULL,
94   `password` text NOT NULL,
95   `created_at` datetime DEFAULT NULL,
96   `updated_at` datetime DEFAULT NULL
97 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
98 COLLATE=utf8mb4_general_ci;
99
100 --
101 -- Dumping data untuk tabel `users`
102 --

```

```

103
104 INSERT INTO `users` (`id`, `username`, `email`,
105 `password`, `created_at`, `updated_at`) VALUES
106 (1, 'AdiSyahputra', '2210817210017@mhs.ulm.ac.id',
107 '$2y$10$SRmP5ETTxBGHLAqUNdiAHn.L2WDWuyKw5uyjJL25ohxP2Dc
108 3dAY9Lu', '2024-06-13 06:33:37', '2024-06-13
109 06:33:37');
110
111 --
112 -- Indexes for dumped tables
113 --
114
115 --
116 -- Indeks untuk tabel `buku`
117 --
118 ALTER TABLE `buku`
119 ADD PRIMARY KEY (`id`);
120
121 --
122 -- Indeks untuk tabel `migrations`
123 --
124 ALTER TABLE `migrations`
125 ADD PRIMARY KEY (`id`);
126
127 --
128 -- Indeks untuk tabel `users`
129 --
130 ALTER TABLE `users`
131 ADD PRIMARY KEY (`id`),
132 ADD UNIQUE KEY `email` (`email`);
133
134 --
135 -- AUTO_INCREMENT untuk tabel yang dibuang
136 --
137
138 --
139 -- AUTO_INCREMENT untuk tabel `buku`
--
ALTER TABLE `buku`
MODIFY `id` bigint(20) UNSIGNED NOT NULL
AUTO_INCREMENT, AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT untuk tabel `migrations`
--
ALTER TABLE `migrations`

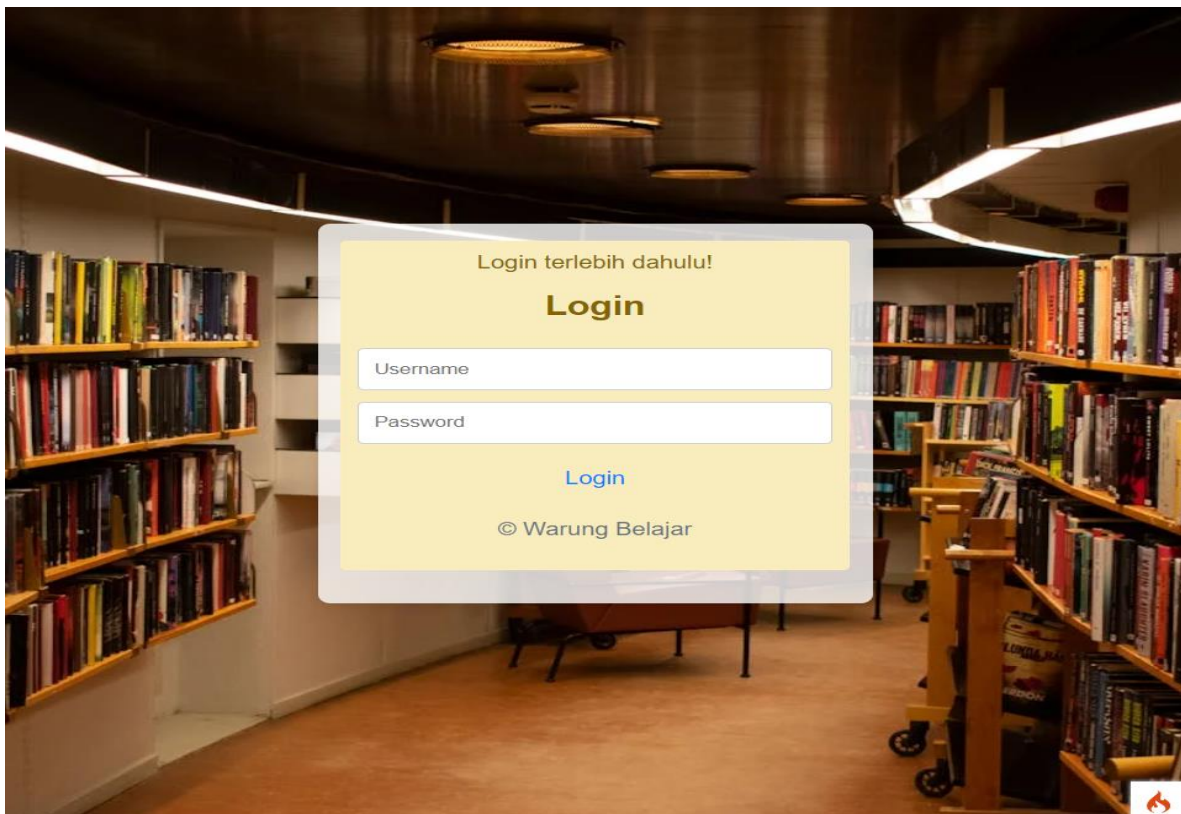
```


<pre> MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3; -- -- AUTO_INCREMENT untuk tabel `users` -- ALTER TABLE `users` MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2; COMMIT; /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */; /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */; /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */; </pre>

Table 14. Source Code Soal 1

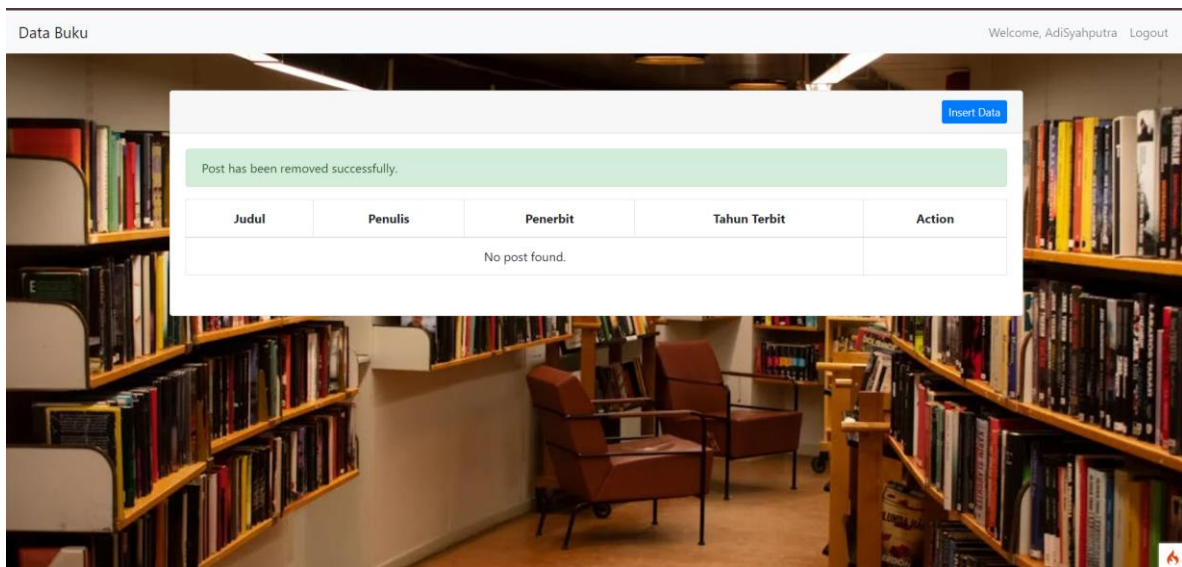
B. Output Program

vw_register.php:



Gambar 1. Screenshot Hasil Jawaban Soal 1

vm_login.php:



Gambar 2. Screenshot Hasil Jawaban Soal 1

vm_buku.php:

Create new post

Judul
Bumi Cinta

Penulis
Habiburrahman El Shirazy

Penerbit
Republika

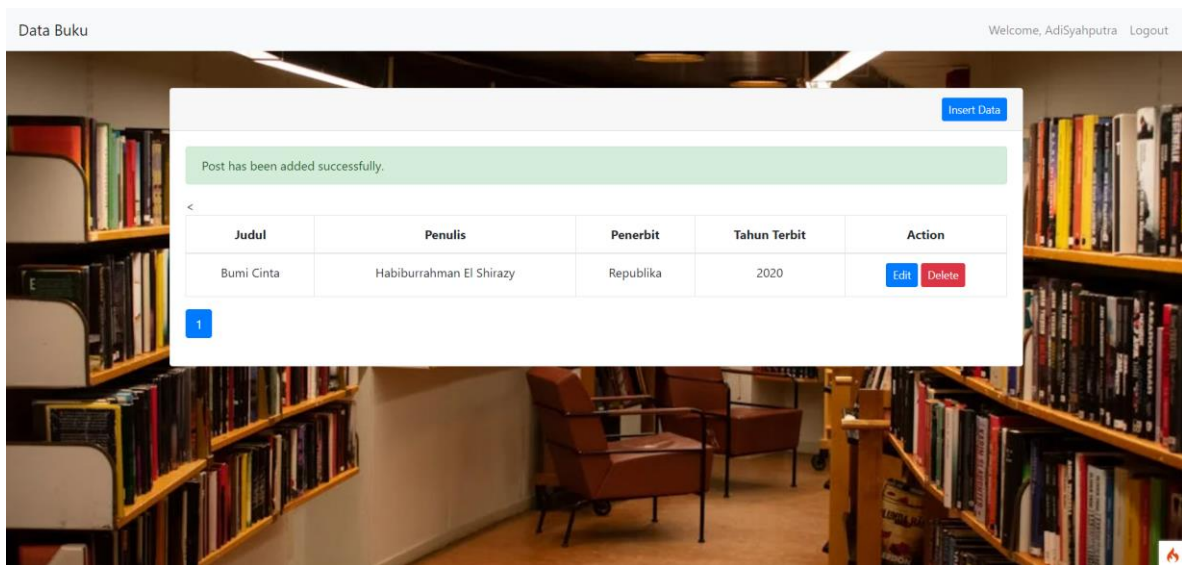
Tahun Terbit
2020

[Save](#) [Back](#)



Gambar 3. Screenshot Hasil Jawaban Soal 1

vm_add.php:



Gambar 4. Screenshot Hasil Jawaban Soal 1

vm_edit.php:

Edit Post

Judul
Bumi Cinta

Penulis
Habiburrahman El Shirazy

Penerbit
Republika

Tahun Terbit
2010

[Update](#) [Back](#)



Gambar 5. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

Routers.php

Menggunakan kelas `RouteCollection` dari namespace `CodeIgniter\Router` untuk mengelola rute.

Memberi petunjuk kepada IDE bahwa variabel `$routes` adalah instansi dari `RouteCollection`.

Mengaktifkan fitur `auto-routing` yang memungkinkan `CodeIgniter` secara otomatis mencari controller dan metode yang sesuai dengan URI yang diminta.

Menetapkan rute untuk URI root (`/`) yang mengarahkan ke metode `login` di controller `Buku`.

Menetapkan rute untuk URI `/home` yang mengarahkan ke metode `index` di controller `Buku`.

Rute GET untuk URI `/register` yang mengarahkan ke metode `index` di controller `Register`.

Rute POST untuk URI `/register/process` yang mengarahkan ke metode `process` di controller `Register`.

Rute GET untuk URI `/login` yang mengarahkan ke metode `index` di controller `Login`.

Rute POST untuk URI `/login/process` yang mengarahkan ke metode `process` di controller `Login`.

Rute GET untuk URI `/logout` yang mengarahkan ke metode `logout` di controller `Login`. Rute GET untuk URI `/buku/create` yang mengarahkan ke metode `create` di controller `Buku`.

Rute POST untuk URI `buku/store` yang mengarahkan ke metode `store` di controller `Buku`.

Rute POST untuk URI `buku/update/` yang mengarahkan ke metode `update` di controller `Buku`, dengan filter `auth` yang memastikan pengguna sudah terautentikasi.

Rute GET untuk URI `buku/edit/` yang mengarahkan ke metode `edit` di controller `Buku`, dengan filter `auth`.

Rute GET untuk URI `buku/delete/` yang mengarahkan ke metode `delete` di controller `Buku`, dengan filter `auth`.

Controllers Buku.php

Namespace dan Penggunaan: Controller ini berada dalam namespace App\Controllers dan menggunakan model BukuModel serta Controller dari CodeIgniter.

Konstruktor: Model BukuModel diinisialisasi, dan helper form serta url dimuat untuk membantu dalam pembuatan form dan pengelolaan URL.

Fungsi: Menampilkan halaman login.

Data yang Dipass: Menyediakan daftar post yang dipaginasi dan judul halaman kepada view vw_login.

Fungsi: Menampilkan halaman utama setelah login.

Otentikasi: Memeriksa apakah pengguna telah login. Jika belum, pengguna akan diarahkan ke halaman login.

Data yang Dipass: Menyediakan daftar post yang dipaginasi dan judul halaman kepada view vw_buku.

Fungsi: Menampilkan form untuk membuat post baru.

Otentikasi: Memeriksa apakah pengguna telah login. Jika belum, pengguna akan diarahkan ke halaman login.

Data yang Dipass: Menyediakan judul halaman kepada view vw_add.

Fungsi: Menyimpan data post baru ke database.

Validasi: Memvalidasi data yang diterima dari form. Jika validasi gagal, akan kembali ke form create.

Respon: Menyimpan data post dan menampilkan pesan sukses atau mengarahkan kembali jika terjadi kesalahan.

Fungsi: Menampilkan form untuk mengedit post yang sudah ada.

Otentikasi: Memeriksa apakah pengguna telah login. Jika belum, pengguna akan diarahkan ke halaman login.

Data yang Dipass: Menyediakan data post yang akan diedit dan judul halaman kepada view vw_edit.

Fungsi: Memperbarui data post yang ada di database.

Otentikasi: Memeriksa apakah pengguna telah login. Jika belum, pengguna akan diarahkan ke halaman login.

Validasi: Memvalidasi data yang diterima dari form. Jika validasi gagal, akan kembali ke form create.

Respon: Memperbarui data post dan menampilkan pesan sukses atau mengarahkan kembali jika terjadi kesalahan.

Fungsi: Menghapus post dari database.

Otentikasi: Memeriksa apakah pengguna telah login. Jika belum, pengguna akan diarahkan ke halaman utama.

Respon: Menghapus data post dan menampilkan pesan sukses atau mengarahkan kembali jika terjadi kesalahan.

Controller Buku mengelola operasi terkait buku seperti login, melihat daftar buku, membuat, mengedit, memperbarui, dan menghapus buku. Setiap metode memeriksa apakah pengguna telah login sebelum melakukan operasi, dan menyediakan data serta pesan yang diperlukan kepada view yang sesuai.

Home.php

Namespace: Controller ini berada dalam namespace `App\Controllers`, yang merupakan namespace standar untuk semua controller di aplikasi CodeIgniter.

Kelas: Home merupakan subclass dari `BaseController`, yang biasanya digunakan sebagai base class untuk semua controller di aplikasi. `BaseController` biasanya mengandung logika dasar yang umum digunakan oleh banyak controller, seperti pemuatan helper atau library.

Fungsi: Metode ini bertanggung jawab untuk menangani permintaan ke rute yang mengarah ke controller Home, biasanya rute dasar (/).

Pengembalian: Mengembalikan tampilan `vw_buku`.

`view('vw_buku')` adalah fungsi bawaan CodeIgniter yang digunakan untuk memuat view `vw_buku`, yang biasanya adalah file HTML atau PHP yang merender halaman web.

Controller Home sangat sederhana dan hanya memiliki satu metode, yaitu `index`, yang menangani permintaan HTTP GET ke rute dasar aplikasi dan mengembalikan tampilan `vw_buku`. Ini biasanya digunakan sebagai titik awal atau halaman utama aplikasi web.

Login.php

Namespace: Controller ini berada dalam namespace `App\Controllers`.

Penggunaan Model: `UsersModel` digunakan untuk mengakses data pengguna dari database.

Kelas: Login merupakan subclass dari `BaseController`, yang memberikan akses ke fungsionalitas dasar yang disediakan oleh `BaseController`.

Fungsi: Metode ini bertanggung jawab untuk menampilkan halaman login.

Pengembalian: Mengembalikan tampilan `vw_login`, yang merupakan halaman login.

Fungsi: Metode ini menangani proses autentikasi pengguna.

Penggunaan Model: Membuat instance dari `UsersModel` untuk mengakses data pengguna.

Mengambil Input: Mengambil username dan password dari permintaan (request) yang dikirimkan oleh pengguna.

Mencari data pengguna berdasarkan username.

Jika pengguna ditemukan, memverifikasi password menggunakan `password_verify`.

Jika verifikasi berhasil, mengatur data sesi (`session`) untuk menandai pengguna sebagai login.

Jika verifikasi gagal atau pengguna tidak ditemukan, mengatur pesan kesalahan (`Flashdata`) dan mengarahkan kembali ke halaman login.

Fungsi: Metode ini menangani proses logout pengguna.

Menghancurkan Sesi: Menghancurkan data sesi untuk mengeluarkan pengguna.

Pengembalian: Mengarahkan kembali pengguna ke halaman login setelah logout.

Register.php

Namespace: Controller ini berada dalam namespace `App\Controllers`.

Penggunaan Model: `UsersModel` digunakan untuk mengakses dan menyimpan data pengguna dalam database.

Kelas: Register merupakan subclass dari `BaseController`, memberikan akses ke fungsionalitas dasar yang disediakan oleh `BaseController`.

Fungsi: Metode ini bertanggung jawab untuk menampilkan halaman pendaftaran.

Pengembalian: Mengembalikan tampilan `vw_register`, yang merupakan halaman pendaftaran.

Fungsi: Metode ini menangani proses pendaftaran pengguna baru.

Melakukan validasi terhadap input yang diterima dari form pendaftaran. Validasi ini mencakup:

`username`: Harus diisi, panjang minimum 4 karakter, panjang maksimum 20 karakter, dan harus unik di tabel `users`.

`password`: Harus diisi, panjang minimum 4 karakter, panjang maksimum 50 karakter.

`password_conf`: Harus sesuai dengan `password`.

`email`: Harus diisi, panjang minimum 4 karakter, panjang maksimum 100 karakter, dan harus berupa email yang valid.

Jika validasi gagal, pesan kesalahan disimpan dalam `Flashdata` dan pengguna diarahkan kembali ke halaman pendaftaran dengan input sebelumnya.

Jika validasi berhasil, `password` pengguna di-hash menggunakan `password_hash` untuk keamanan.

Data pengguna baru (`username`, hashed `password`, `email`) disimpan dalam tabel `users` menggunakan model `UsersModel`.

Setelah data pengguna berhasil disimpan, pengguna diarahkan ke halaman login.

AuthFilter.php

Namespace: Filter ini berada dalam namespace `App\Filters`, yang merupakan tempat standar untuk menyimpan filter di aplikasi `CodeIgniter`.

Implementasi: Kelas `AuthFilter` mengimplementasikan `FilterInterface`, yang berarti kelas ini harus menyediakan implementasi untuk metode `before` dan `after`.

Fungsi Metode Before: Metode ini dieksekusi sebelum request diproses oleh controller.

Memeriksa apakah session memiliki `user_id` untuk menentukan apakah pengguna sudah login atau belum.

Jika tidak ada `user_id` dalam session (pengguna belum login), pengguna akan diarahkan ke halaman login (`/login`) menggunakan `redirect()`.

Fungsi Metode After: Metode ini dieksekusi setelah request diproses oleh controller, dan sebelum response dikirimkan ke browser.

Implementasi Kosong: Dalam filter ini, metode `after` tidak melakukan apa-apa. Ini bisa diisi sesuai kebutuhan aplikasi, seperti logging atau manipulasi response.

vm_register.php

xtend Layout: Menggunakan `<?= $this->extend('layouts/main') ?>` untuk memperluas layout utama. Ini berarti template ini akan menggunakan layout yang telah didefinisikan dalam layouts/main.

Section Content: `<?= $this->section('content') ?>` dan `<?= $this->endSection() ?>` digunakan untuk mendefinisikan konten utama yang akan dimasukkan ke dalam layout utama.

Container dan Grid System: Menggunakan Bootstrap untuk mengatur layout dengan container, row, dan col (kolom).

Card Component: Menggunakan card dari Bootstrap untuk membungkus konten.

Card Header: Menampilkan judul yang dinamis dari variabel \$title.

Card Body: Menyimpan form dan pesan kesalahan.

Menampilkan pesan kesalahan jika ada. Menggunakan `session()->getFlashdata('error')` untuk menampilkan pesan flash error yang disimpan dalam session.

Menampilkan daftar kesalahan validasi menggunakan `validation_list_errors()`, yang akan menampilkan kesalahan validasi yang terjadi saat form dikirim.

Form: Menggunakan helper `form_open('buku/store')` untuk membuka form dan menentukan action URL.

Form Fields: Memiliki beberapa input field untuk judul, penulis, penerbit, dan tahun_terbit, masing-masing dengan label dan placeholder.

Submit Button: Tombol untuk menyimpan data.

Back Link: Link untuk kembali ke halaman utama.

Section Extra JavaScript: Menggunakan `<?= $this->section('extra-js') ?>` dan `<?= $this->endSection() ?>` untuk mendefinisikan bagian khusus untuk JavaScript tambahan.

Menyertakan CSS dan JavaScript Summernote dari CDN.

Menginisialisasi Summernote pada elemen dengan ID `post_content` saat dokumen siap.

vm_login.php

Extend Layout: Template ini memperluas layout utama yang didefinisikan di layouts/main. Ini memungkinkan untuk menggunakan elemen layout yang sama di berbagai halaman.

Section Content: Bagian utama dari konten halaman didefinisikan dalam section content.

CSS Inline: Mengatur latar belakang halaman dengan gambar menggunakan CSS inline. Gambar diambil dari URL eksternal, diatur agar tidak mengulangi, tetap pada posisinya, dan menyesuaikan dengan ukuran halaman.

Navigasi: Menggunakan komponen navbar dari Bootstrap.

Brand: Teks "Data Buku" yang menunjukkan nama aplikasi atau bagian aplikasi.

Jika pengguna sudah login, navbar menampilkan pesan selamat datang dan opsi logout.

Jika pengguna belum login, navbar menampilkan opsi login dan register.

Container dan Grid System: Menggunakan Bootstrap untuk mengatur layout dengan container, row, dan col (kolom).

Card Component: Menggunakan card dari Bootstrap untuk membungkus konten utama.

Card Header: Menampilkan tombol untuk menambahkan data baru yang mengarahkan ke halaman pembuatan buku baru.

Menampilkan pesan sukses jika ada operasi yang berhasil dilakukan, seperti penambahan atau penghapusan data.

Menampilkan pesan kesalahan jika terjadi masalah saat melakukan operasi.

Tabel Data Buku: Menggunakan tabel Bootstrap untuk menampilkan daftar buku.

Header Tabel: Menampilkan judul kolom untuk judul, penulis, penerbit, tahun terbit, dan aksi.

Body Tabel: Menampilkan data buku yang diambil dari variabel posts. Jika data buku ada, data ditampilkan dalam baris tabel. Jika tidak ada, ditampilkan pesan "No post found."

Aksi: Tautan untuk mengedit buku yang mengarahkan ke halaman edit.

Tombol untuk menghapus buku, dengan konfirmasi sebelum penghapusan dilakukan. Penghapusan menggunakan form dengan metode POST yang disimulasikan sebagai DELETE.

Pagination: Menampilkan tautan pagination jika data buku terpaginasi, dengan menambahkan kelas Bootstrap untuk styling.

Section Extra JavaScript: Bagian ini digunakan untuk menambahkan JavaScript khusus.

Script jQuery: Menambahkan kelas Bootstrap ke elemen pagination saat dokumen siap, untuk memastikan styling yang konsisten.

vm_buku.php

Extend Layout: menggunakan layout utama yang didefinisikan di layouts/main.

Section Content: Bagian utama dari konten halaman didefinisikan dalam section content.

Container dan Grid System: Menggunakan Bootstrap untuk mengatur layout dengan container, row, dan col (kolom).

Card Component: Menggunakan card dari Bootstrap untuk membungkus konten utama.

Card Header: Menampilkan judul yang dinamis dari variabel \$title.

Menampilkan pesan kesalahan jika ada. Menggunakan `session()->getFlashdata('error')` untuk menampilkan pesan flash error yang disimpan dalam session.

Menampilkan daftar kesalahan validasi menggunakan `validation_list_errors()`, yang akan menampilkan kesalahan validasi yang terjadi saat form dikirim.

Menggunakan helper `form_open('buku/update/'. $post['id'])` untuk membuka form dengan method PUT.

Judul: Input untuk judul buku.

Penulis: Input untuk penulis buku.

Penerbit: Input untuk penerbit buku.

Tahun Terbit: Input untuk tahun terbit buku.

Submit Button: Tombol untuk mengirim data update.

Back Link: Link untuk kembali ke halaman utama.

Section Extra JavaScript: Bagian ini digunakan untuk menambahkan JavaScript khusus.

Menyertakan CSS dan JavaScript Summernote dari CDN.

Menginisialisasi Summernote pada elemen dengan ID `post_content` saat dokumen siap.

vm_add.php

DOCTYPE: Deklarasi `<!doctype html>` menunjukkan bahwa dokumen ini adalah HTML5.

HTML Tag: Tag `<html>` dengan atribut `lang="en"` menunjukkan bahwa bahasa dokumen adalah bahasa Inggris.

Head: Bagian `<head>` berisi metadata, termasuk charset, viewport untuk responsivitas, deskripsi, penulis, generator, dan judul halaman.

Style: CSS inline digunakan untuk styling halaman dan elemen-elemen di dalamnya.

Background Image: Mengatur gambar latar belakang untuk halaman dengan URL eksternal, memastikan gambar tidak mengulangi, posisinya tetap, dan menutupi seluruh viewport.

Flexbox Layout: Menggunakan flexbox untuk menampilkan konten secara terpusat di tengah halaman, baik secara horizontal maupun vertikal.

Form Container: `.form-signin` adalah kontainer utama untuk form login, diatur dengan lebar maksimum, padding, latar belakang semi-transparan, bayangan kotak, border-radius, dan teks yang rata tengah.

Header Form: `<h1>` untuk judul form dengan ukuran font yang disesuaikan dan margin bawah.

Input untuk username dan password, diatur dengan lebar penuh, padding, margin, border, border-radius, dan ditampilkan sebagai blok.

Tombol untuk mengirim form, dengan styling border transparan, latar belakang transparan, teks berwarna biru, dan efek hover untuk mengubah warna teks menjadi lebih gelap.

Div dengan class `.alert` untuk menampilkan pesan kesalahan dari session flash data.

Jika ada pesan error di session flash data, pesan tersebut akan ditampilkan di dalam alert box.

Onload Event: Menggunakan JavaScript untuk menampilkan alert box saat halaman dimuat. `window.onload` memastikan bahwa pesan "Login terlebih dahulu!" ditampilkan ketika halaman selesai dimuat.

Form Tag: Formulir menggunakan metode POST dan mengarahkan data ke URL `/login/process`.

CSRF Field: Menggunakan `csrf_field()` untuk menambahkan token CSRF guna mencegah serangan CSRF.

username: Input untuk nama pengguna dengan placeholder, atribut `required`, dan `autofocus`.

password: Input untuk kata sandi dengan placeholder dan atribut `required`.

Submit Button: Tombol untuk mengirim form.

Text Muted: Paragraf dengan kelas `text-muted` untuk teks footer, diatur dengan warna teks yang lebih redup untuk menunjukkan tahun dan hak cipta.

vm_edit.php

DOCTYPE: Deklarasi `<!doctype html>` menunjukkan bahwa dokumen ini adalah HTML5.

HTML Tag: Tag `<html>` dengan atribut `lang="en"` dan kelas `h-100` menunjukkan bahwa bahasa dokumen adalah bahasa Inggris dan tinggi halaman diatur penuh.

Head: Bagian `<head>` berisi metadata, termasuk charset, viewport untuk responsivitas, deskripsi, penulis, generator, dan judul halaman. Menyertakan link ke CSS Bootstrap dan custom styles.

Bootstrap CSS: Link ke file CSS Bootstrap yang diambil dari base URL aplikasi.

Custom Styles: Menyertakan beberapa gaya kustom dalam tag `<style>`, termasuk gaya untuk placeholder dan media query untuk teks besar pada layar yang lebih lebar.

Custom Stylesheet: Link ke file CSS kustom `sticky-footer.css` yang diambil dari base URL aplikasi.

Body Kelas Flexbox: Kelas `d-flex flex-column h-100` digunakan untuk membuat layout fleksibel dengan tinggi penuh.

Container: Menampung konten utama halaman, termasuk form registrasi.

Header: Judul halaman dan deskripsi singkat untuk form registrasi.

Flash Messages: Menampilkan pesan kesalahan dari session flash data jika ada.

CSRF Field: Menggunakan `csrf_field()` untuk menambahkan token CSRF guna mencegah serangan CSRF.

username: Input untuk nama pengguna.

password: Input untuk kata sandi.

password_conf: Input untuk konfirmasi kata sandi.

email: Input untuk email pengguna.

Submit Button: Tombol untuk mengirim form.

Sticky Footer: Menggunakan kelas `footer mt-auto py-3 bg-light` untuk membuat footer yang selalu berada di bawah halaman dengan padding dan background berwarna terang.

Footer Content: Menampilkan teks footer.

Bootstrap JS: Link ke file JavaScript Bootstrap yang diambil dari base URL aplikasi, memastikan semua komponen Bootstrap berfungsi dengan baik.

BukuModel.php

Namespace: Model ini berada dalam namespace `App\Models`, yang merupakan tempat standar untuk menyimpan model di aplikasi CodeIgniter.

Kelas: `BukuModel` merupakan subclass dari `CodeIgniter\Model`, yang menyediakan fungsionalitas dasar untuk berinteraksi dengan database.

Table: Properti `$table` mendefinisikan nama tabel database yang diakses oleh model ini, yaitu `buku`.

Allowed Fields: Properti `$allowedFields` adalah array yang mendefinisikan kolom-kolom yang diizinkan untuk diisi melalui metode `insert` dan `update`. Dalam model ini, kolom yang diizinkan adalah `judul`, `penulis`, `penerbit`, dan `tahun_terbit`.

Created Field: Properti `$createdField` mengatur kolom yang akan diisi dengan timestamp saat record dibuat, yaitu `created_at`.

Updated Field: Properti `$updatedField` mengatur kolom yang akan diisi dengan timestamp saat record diperbarui, yaitu `updated_at`.

Deleted Field: Properti `$deletedField` mengatur kolom yang akan diisi dengan timestamp saat record dihapus secara soft delete, yaitu `deleted_at`.

Validation Rules: Properti `$validationRules` adalah array yang mendefinisikan aturan validasi untuk setiap kolom yang diizinkan.

Judul: Harus diisi (required) dan berupa string (string).

Penulis: Harus diisi (required) dan berupa string (string).

Penerbit: Harus diisi (required) dan berupa string (string).

Tahun Terbit: Harus diisi (required), berupa angka (integer), lebih besar dari 1800 (`greater_than[1800]`), dan lebih kecil dari 2024 (`less_than[2024]`).

Validation Messages: Setiap aturan validasi memiliki pesan kesalahan yang spesifik untuk setiap skenario kesalahan. Pesan ini akan ditampilkan jika validasi gagal.

Skip Validation: Properti `$skipValidation` diatur ke `false` yang berarti validasi tidak akan dilewati. Setiap kali data disimpan atau diperbarui, aturan validasi akan diterapkan.

Model `BukuModel` ini digunakan untuk berinteraksi dengan tabel `buku` di database. Model ini memiliki beberapa properti penting:

`table` untuk menentukan tabel yang digunakan.

`allowedFields` untuk menentukan kolom mana yang bisa diisi.

`createdField`, `updatedField`, dan `deletedField` untuk otomatis mengelola kolom timestamp.

`validationRules` untuk mendefinisikan aturan validasi data yang dimasukkan ke dalam tabel `buku`.

Dengan menggunakan model ini, operasi CRUD (Create, Read, Update, Delete) pada tabel buku menjadi lebih mudah dan aman karena adanya validasi yang memastikan data yang dimasukkan sesuai dengan aturan yang telah ditentukan.

UsersModel.php

Namespace: Model ini berada dalam namespace `App\Models`, yang merupakan tempat standar untuk menyimpan model di aplikasi CodeIgniter.

Kelas: `UsersModel` merupakan subclass dari `CodeIgniter\Model`, yang menyediakan fungsionalitas dasar untuk berinteraksi dengan database.

Table: Properti `$table` mendefinisikan nama tabel database yang diakses oleh model ini, yaitu `users`.

Return Type: Properti `$returnType` diatur menjadi `object`, yang berarti data yang diambil dari database akan dikembalikan sebagai objek, bukan array.

Use Timestamps: Properti `$useTimestamps` diatur ke `true`, yang berarti CodeIgniter akan secara otomatis mengelola kolom `created_at` dan `updated_at` di tabel `users`. Ini mengharuskan tabel `users` memiliki kolom `created_at` dan `updated_at`.

Allowed Fields: Properti `$allowedFields` adalah array yang mendefinisikan kolom-kolom yang diizinkan untuk diisi melalui metode `insert` dan `update`. Dalam model ini, kolom yang diizinkan adalah `username`, `password`, dan `email`.

Model `UsersModel` ini digunakan untuk berinteraksi dengan tabel `users` di database.

table: Menentukan tabel yang digunakan (`users`).

returnType: Menentukan format data yang dikembalikan (`object`).

useTimestamps: Mengaktifkan pengelolaan otomatis kolom timestamp (`created_at` dan `updated_at`).

allowedFields: Menentukan kolom mana yang bisa diisi (`username`, `password`, dan `email`).

Dengan menggunakan model ini, operasi CRUD (Create, Read, Update, Delete) pada tabel `users` menjadi lebih mudah dan terstruktur. Penggunaan timestamps otomatis juga membantu dalam melacak waktu pembuatan dan pembaruan data. Model ini sederhana namun mencakup fungsionalitas dasar yang diperlukan untuk mengelola data pengguna dalam aplikasi.

Data_buku.sql

Header: Informasi tentang versi phpMyAdmin, host, waktu pembuatan, versi server MariaDB, dan versi PHP.

Pengaturan Transaksi: Menetapkan mode SQL, memulai transaksi, dan mengatur zona waktu.

Pengaturan Karakter: Menyimpan dan menetapkan pengaturan karakter lama untuk memastikan kompatibilitas.

Struktur dan Data Tabel Database data_buku: Semua tabel yang dideskripsikan berada dalam database data_buku.

Tabel buku Struktur Tabel:

id: Kolom kunci utama, tipe bigint, unsigned, tidak boleh null.

judul: Kolom untuk judul buku, tipe varchar, panjang maksimal 255, tidak boleh null.

penulis: Kolom untuk penulis buku, tipe varchar, panjang maksimal 255, tidak boleh null.

penerbit: Kolom untuk penerbit buku, tipe varchar, panjang maksimal 255, tidak boleh null.

tahun_terbit: Kolom untuk tahun terbit, tipe year, panjang 4, bisa null.

Data Tabel: Memasukkan satu record buku dengan id 2, judul "Bumi Cinta", penulis "Habiburrahman El Shirazy", penerbit "Republika", dan tahun terbit 2020.

Tabel migrations Struktur Tabel:

id: Kolom kunci utama, tipe bigint, unsigned, tidak boleh null.

version, class, group, namespace: Kolom-kolom untuk informasi versi migrasi, kelas, grup, dan namespace, tipe varchar, panjang maksimal 255, tidak boleh null.

time: Kolom untuk waktu migrasi, tipe int, tidak boleh null.

batch: Kolom untuk batch migrasi, tipe int, unsigned, tidak boleh null.

Data Tabel: Memasukkan dua record migrasi dengan informasi versi, kelas, grup, namespace, waktu, dan batch.

Tabel users Struktur Tabel:

id: Kolom kunci utama, tipe int, unsigned, tidak boleh null.

username, email: Kolom untuk username dan email, tipe varchar, panjang maksimal 255, tidak boleh null.

password: Kolom untuk password, tipe text, tidak boleh null.

`created_at`, `updated_at`: Kolom untuk mencatat waktu pembuatan dan pembaruan, tipe `datetime`, bisa null.

Data Tabel: Memasukkan satu record pengguna dengan id 1, username "AdiSyahputra", email "2210817210017@mhs.ulm.ac.id", password hashed, dan waktu pembuatan serta pembaruan yang sama.

Indeks buku: Menambahkan indeks utama pada kolom id.

Indeks migrations: Menambahkan indeks utama pada kolom id.

Indeks users: Menambahkan indeks utama pada kolom id dan indeks unik pada kolom email.

Menetapkan properti `AUTO_INCREMENT` pada kolom id dari tabel buku, migrations, dan users untuk memastikan nilai id bertambah secara otomatis untuk setiap record baru.

Mengakhiri transaksi dengan perintah `COMMIT` untuk memastikan semua perubahan yang dilakukan dalam transaksi diterapkan.

SQL dump ini berisi struktur dan data dari tiga tabel dalam database `data_buku`: buku, migrations, dan users. Tabel-tabel ini memiliki kolom yang ditetapkan dengan tipe data yang sesuai dan constraints seperti kunci utama dan indeks unik. Data sampel dimasukkan ke dalam setiap tabel untuk menunjukkan cara data diorganisir. Properti `AUTO_INCREMENT` ditetapkan untuk kolom id di setiap tabel untuk memudahkan penambahan record baru. Transaksi digunakan untuk memastikan konsistensi data selama proses impor.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/MAdiSyahputra/PemrogramanWeb_2/blob/cea07e1481cf61510e182637b75fc3cdb616b831/Modul%206_PRAK601.zip