

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD dan Login pada Codeigniter

Oleh:

Muhammad Kholid Syaifullah

NIM. 2210817210025

**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 : Muhammad Kholid Syaifullah

NIM : 2210817210025

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.....	48
C. Pembahasan.....	49
Tautan Git	53

DAFTAR GAMBAR

Gambar 1. Soal 1	6
Gambar 2. Screenshot Tampilan Login.....	48
Gambar 3. Screenshot Tampilan Home.....	48
Gambar 4. Screenshot Tampilan Tambah Data.....	49
Gambar 5. Screenshot Tampilan Edit Data	49

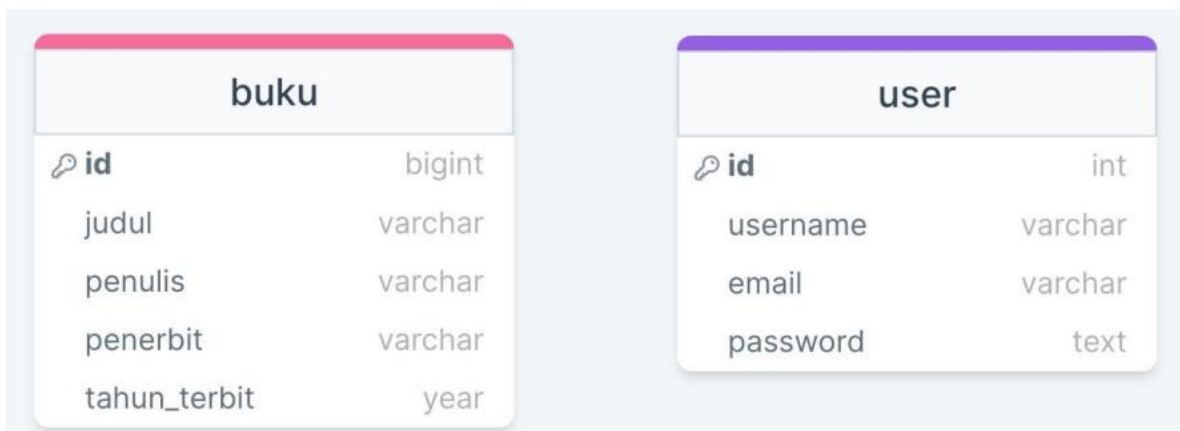
DAFTAR TABEL

Tabel 1. Source Code Routes.php	8
Tabel 2. Source Code AuthController.php	10
Tabel 3. Source Code BaseController.php	11
Tabel 4. Source Code CRUDController.php	13
Tabel 5. Source Code AuthFilter.php	15
Tabel 6. Source Code login.php	21
Tabel 7. Source Code index.php.....	26
Tabel 8. Source Code create.php	33
Tabel 9. Source Code edit.php.....	40
Tabel 10. Source Code BukuModel.php	42
Tabel 11. Source Code UserModel.php.....	43
Tabel 12. Skema Database prak601.sql.....	48

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 berikut:



Gambar 1. Soal 1

Simpan dalam folder github PRAK 601

A. Source Code

Config/Routes.php

```
1 <?php
2
3 namespace Config;
4
5 $routes = Services::routes();
6
7 $routes->setDefaultNamespace('App\Controllers');
8 $routes->setDefaultController('Home');
9 $routes->setDefaultMethod('index');
10 $routes->setTranslateURIDashes(false);
11 $routes->set404Override();
12
13 $routes->group('', ["filter" => "auth"], static
    function($routes){
14     $routes->get('/', 'CRUDController::index');
15     $routes->get('/tambahdata', 'CRUDController::tambah');
16     $routes->post('/tambahdata',
        'CRUDController::simpan');
17     $routes->get('/editdata/(:num)',
        'CRUDController::edit/$1');
18     $routes->post('/editdata/(:num)',
        'CRUDController::update/$1');
19     $routes->delete('/hapusdata/(:num)',
        'CRUDController::hapus/$1');
20     $routes->get('/logout', 'AuthController::logout');
21 });
22
23 $routes->get('/login', 'AuthController::index');
24 $routes->post('/login', 'AuthController::login');
```

25	
26	<code>if (is_file(APPPATH . 'Config/' . ENVIRONMENT .</code>
27	<code> '/Routes.php')) {</code>
28	<code> require APPPATH . 'Config/' . ENVIRONMENT .</code>
	<code> '/Routes.php';</code>
	<code>}</code>

Tabel 1. Source Code Routes.php

Controllers/AuthController.php

1	<code><?php</code>
2	
3	<code>namespace App\Controllers;</code>
4	
5	<code>use App\Controllers\BaseController;</code>
6	<code>use App\Models\UserModel;</code>
7	
8	<code>class AuthController extends BaseController</code>
9	<code>{</code>
10	<code> public function index()</code>
11	<code> {</code>
12	<code> return view('login');</code>
13	<code> }</code>
14	
15	<code> public function login()</code>
16	<code> {</code>
17	<code> \$model = new UserModel();</code>
18	<code> \$username = request()->getPost('username');</code>
19	<code> \$email = request()->getPost('email');</code>
20	<code> \$password = request()->getPost('password');</code>
21	


```
22         $dataUser = $model->where(["username" =>
$username]) -> first();
23
24         if ($dataUser) {
25             if(password_verify($password,
$username['password'])) {
26                 session() -> set([
27                     "username" => $username,
28                     "email" => $email,
29                     "isLoggedIn" => true
30                 ]);
31
32                 return redirect()->to(base_url('/'));
33             } else{
34                 session() -> setFlashdata(["pesan" =>
"Login gagal, username, email atau password salah"]);
35                 return redirect()-
>to(base_url('/login'));
36             }
37
38         } else {
39             session() -> setFlashdata(["pesan" =>
"Login gagal, username, email atau password salah"]);
40             return redirect()->to(base_url('/login'));
41         }
42     }
43
44     public function logout()
45     {
46         session()->destroy();
47         return redirect()->to(base_url('login'));
```

48	}
49	}

Tabel 2. Source Code AuthController.php

Controllers/BaseController.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	use CodeIgniter\Controller;
6	use CodeIgniter\HTTP\CLIRequest;
7	use CodeIgniter\HTTP\IncomingRequest;
8	use CodeIgniter\HTTP\RequestInterface;
9	use CodeIgniter\HTTP\ResponseInterface;
10	use Psr\Log\LoggerInterface;
11	
12	abstract class BaseController extends Controller
13	{
14	/**
15	* @var CLIRequest IncomingRequest
16	*/
17	protected \$request;
18	
19	protected \$helpers = [];
20	
21	public function initController(RequestInterface
	\$request, ResponseInterface \$response, LoggerInterface
	\$logger)
22	{
23	parent::initController(\$request, \$response,
	\$logger);

24	}
25	}

Tabel 3. Source Code BaseController.php

Crontrollers/CRUDController.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Controllers\BaseController;
6	use App\Models\BukuModel;
7	
8	class CRUDController extends BaseController
9	{
10	public function index()
11	{
12	\$model = new BukuModel();
13	\$semuaData = \$model->findAll();
14	return view("index", [
15	"data" => \$semuaData
16]);
17	
18	}
19	
20	public function tambah()
21	{
22	return view("create");
23	}
24	
25	public function simpan()
26	{

```
27     $model = new BukuModel();
28     $judul = request()->getPost("judul");
29     $penulis = request()->getPost("penulis");
30     $penerbit = request()->getPost("penerbit");
31     $tahun_terbit = request()-
>getPost("tahun_terbit");
32
33     $model -> insert([
34         "judul" => $judul,
35         "penulis" => $penulis,
36         "penerbit" => $penerbit,
37         "tahun_terbit" => $tahun_terbit
38     ]);
39     return redirect() ->to(base_url('/'));
40 }
41
42 public function edit($id)
43 {
44     $model = new BukuModel();
45     $data = $model-> find($id);
46
47     return view('edit', [
48         "data" => $data
49     ]);
50 }
51
52 public function update($id)
53 {
54     $model = new BukuModel();
55     $judul = request()->getPost("judul");
56     $penulis = request()->getPost("penulis");
```

57	\$penerbit = request()->getPost("penerbit");
58	\$tahun_terbit = request()->getPost("tahun_terbit");
59	
60	\$model -> update(\$id, [
61	"judul" => \$judul,
62	"penulis" => \$penulis,
63	"penerbit" => \$penerbit,
64	"tahun_terbit" => \$tahun_terbit
65]);
66	return redirect() ->to(base_url('/'));
67	
68	return view('edit', [
69	"data" => \$data
70]);
71	
72	return redirect() ->to(base_url('/'));
73	}
74	
75	public function hapus(\$id)
76	{
77	\$model = new BukuModel();
78	\$model -> delete(\$id);
79	return redirect() ->to(base_url('/'));
80	}
81	}

Tabel 4. Source Code CRUDController.php

Filters/AuthFiler.php

1	<?php
2	

```
3 namespace App\Filters;
4
5 use CodeIgniter\Filters\FilterInterface;
6 use CodeIgniter\HTTP\RequestInterface;
7 use CodeIgniter\HTTP\ResponseInterface;
8
9 class AuthFilter implements FilterInterface
10 {
11     /**
12      * @param RequestInterface $request
13      * @param array|null       $arguments
14      * @return mixed
15      */
16     public function before(RequestInterface $request,
17 $arguments = null)
18     {
19         if(!session()->get('isLoggedIn')) {
20             session() ->setFlashdata('error','Silahkan
Login Terlebih Dahulu');
21             return redirect()->to(base_url('login'));
22         }
23
24     /**
25      * @param RequestInterface $request
26      * @param ResponseInterface $response
27      * @param array|null       $arguments
28      * @return mixed
29      */
30     public function after(RequestInterface $request,
ResponseInterface $response, $arguments = null) {}
```

31	}
----	---

Tabel 5. Source Code AuthFilter.php

View/login.php

1	<!DOCTYPE html>
2	<html lang="en">
3	<head>
4	<meta charset="utf-8">
5	<meta name="viewport" content="width=device-
	width, initial-scale=1">
6	<title>Login</title>
7	<style>
8	body, html {
9	height: 100%;
10	margin: 0;
11	font-family: Arial, Helvetica, sans-
	serif;
12	display: flex;
13	justify-content: center;
14	align-items: center;
15	background-color: #222222;
16	}
17	.container {
18	width: 500px;
19	max-width: 400px;
20	background-color: white;
21	padding: 20px;
22	border-radius: 10px;
23	text-align: center;
24	}
25	.container h1 {

26	font-size: 35px;
27	margin-bottom: 20px;
28	color: #333;
29	}
30	.container form {
31	width: 100%;
32	}
33	.container form .form-group {
34	margin-bottom: 15px;
35	text-align: left;
36	}
37	.container form .form-group label {
38	display: block;
39	margin-bottom: 5px;
40	color: #666;
41	}
42	.container form .form-group input {
43	width: 100%;
44	padding: 10px;
45	border: 1px solid #ccc;
46	border-radius: 5px;
47	box-sizing: border-box;
48	}
49	.container form .form-group input:focus {
50	border-color: #007bff;
51	outline: none;
52	}
53	.container form .form-group .invalid-
	feedback {
54	color: red;
55	font-size: 12px;

56	display: none;
57	}
58	.container form .form-group .alert {
59	color: #856404;
60	background-color: #fff3cd;
61	border-color: #ffeeba;
62	padding: 10px;
63	border-radius: 5px;
64	margin-bottom: 15px;
65	text-align: left;
66	}
67	.container form .btn {
68	width: 100%;
69	padding: 10px;
70	border: none;
71	border-radius: 5px;
72	background-color: #007bff;
73	color: white;
74	font-size: 16px;
75	cursor: pointer;
76	box-sizing: border-box;
77	}
78	.container form .btn:hover {
79	background-color: #0056b3;
80	}
81	.container .footer {
82	margin-top: 20px;
83	color: #999;
84	}
85	</style>
86	<script>

87	function validateForm() {	
88	var isValid = true;	
89		
90	var username	=
	document.getElementById('username');	
91	var usernameFeedback	=
	username.nextElementSibling;	
92	if (username.value.trim() === '') {	
93	usernameFeedback.style.display	=
	'block';	
94	isValid = false;	
95	} else {	
96	usernameFeedback.style.display	=
	'none';	
97	}	
98		
99	var email	=
	document.getElementById('email');	
100	var emailFeedback	=
	email.nextElementSibling;	
101	var emailPattern	=
	/^[^\s@]+@[^\s@]+\.[^\s@]+\$/;	
102	if (!emailPattern.test(email.value)) {	
103	emailFeedback.style.display	=
	'block';	
104	isValid = false;	
105	} else {	
106	emailFeedback.style.display	=
	'none';	
107	}	
108		

109	var password =
	document.getElementById('password');
110	var passwordFeedback =
	password.nextElementSibling;
111	if (password.value.trim() === '') {
112	passwordFeedback.style.display =
	'block';
113	isValid = false;
114	} else {
115	passwordFeedback.style.display =
	'none';
116	}
117	
118	return isValid;
119	}
120	</script>
121	</head>
122	<body>
123	<div class="container">
124	<?php if(session() ->
	getFlashdata('pesan')): ?>
125	<div class="alert" role="alert">
126	<?= session()-
	>getFlashdata('pesan') ?>
127	</div>
128	<?php endif?>
129	
130	<?php if(session() ->
	getFlashdata('error')): ?>
131	<div class="alert" role="alert">
132	

133	<code><?= >getFlashdata('error') ?> </div> <?php endif?> <h1>Perpustakaan Online</h1> <h2>Login</h2> <form action="<?= method="POST" class="needs-validation" novalidate="" autocomplete="off" onsubmit="return validateForm()"> <div class="form-group"> <label for="username">User Name</label> <input id="username" type="text" name="username" required> <div class="invalid- feedback">Username is required</div> </div> <div class="form-group"> <label for="email">E-Mail Address</label> <input id="email" type="email" name="email" value="" required autofocus> <div class="invalid- feedback">Email is invalid</div> </div> <div class="form-group"> <label for="password">Password</label> <input id="password" type="password" name="password" required></code>
134	
135	
136	
137	
138	
139	
140	
141	
142	
143	
144	
145	
146	
147	
148	
149	
150	
151	
152	

153	<code><div class="invalid-feedback">Password is required</div></code>
154	<code></div></code>
155	<code><button type="submit" class="btn">Login</button></code>
156	<code></form></code>
157	<code></div></code>
158	<code></body></code>
159	<code></html></code>

Tabel 6. Source Code login.php

View/index.php

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code><meta charset="UTF-8"></code>
5	<code><meta http-equiv="X-UA-Compatible" content="IE=edge"></code>
6	<code><meta name="viewport" content="width=device-width, initial-scale=1.0"></code>
7	<code><title>Home</title></code>
8	<code><style></code>
9	<code>body, html {</code>
10	<code>height: 100%;</code>
11	<code>margin: 0;</code>
12	<code>font-family: Arial, Helvetica, sans-serif;</code>
13	<code>display: flex;</code>
14	<code>background-color: #222222;</code>
15	<code>}</code>
16	<code>.sidebar {</code>
17	<code>width: 250px;</code>

18	background-color: #007bff;
19	color: white;
20	display: flex;
21	flex-direction: column;
22	align-items: center;
23	padding-top: 20px;
24	box-shadow: 2px 0 5px rgba(0,0,0,0.1);
25	position: relative;
26	height: 100%;
27	}
28	
29	.logout-btn {
30	margin-bottom: 20px;
31	}
32	
33	.sidebar h2 {
34	margin: 0;
35	font-size: 24px;
36	}
37	.sidebar a {
38	color: white;
39	text-decoration: none;
40	padding: 10px 20px;
41	margin: 10px 0;
42	width: 70%;
43	text-align: center;
44	border-radius: 5px;
45	}
46	.sidebar a:hover {
47	background-color: #c82333;
48	}

```
49     .content {
50         flex: 1;
51         padding: 20px;
52     }
53     .content h1 {
54         font-size: 35px;
55         margin-bottom: 20px;
56         color: #fff;
57     }
58     .content h2 {
59         font-size: 24px;
60         margin-bottom: 20px;
61         color: #fff;
62     }
63     .table-container {
64         margin-top: 20px;
65     }
66     .table {
67         width: 100%;
68         border-collapse: collapse;
69     }
70     .table th, .table td {
71         background-color: #dddddd;
72         padding: 10px;
73         border: 2px solid #ccc;
74         text-align: left;
75     }
76     .table th {
77         background-color: #007bff;
78         color: white;
79     }
```

80	.btn {
81	padding: 10px 50px;
82	border: none;
83	border-radius: 5px;
84	color: white;
85	font-size: 16px;
86	cursor: pointer;
87	text-decoration: none;
88	display: inline-block;
89	}
90	.btn-success {
91	background-color: #28a745;
92	transition: background-color 0.3s;
93	}
94	.btn-success:hover {
95	background-color: #218838;
96	}
97	.btn-warning {
98	background-color: #ffc107;
99	}
100	.btn-danger {
101	background-color: #dc3545;
102	}
103	.btn-sm {
104	padding: 5px 10px;
105	font-size: 14px;
106	}
107	</style>
108	</head>
109	<body>
110	<div class="sidebar">

111	<code><h2>Menu</h2></code>
112	<code><a href="<?= base_url('/logout')?>" class="btn</code>
	<code>btn-danger logout-btn">Logout</code>
113	<code></div></code>
114	
115	<code><div class="content"></code>
116	<code><div></code>
117	<code><h1>Selamat Datang di database</code>
	<code>Perpustakaan</h1></code>
118	<code></div></code>
119	<code><div class="table-container"></code>
120	<code><h2>Daftar Buku</h2></code>
121	<code><table class="table"></code>
122	<code><thead></code>
123	<code><tr></code>
124	<code><th>Judul</th></code>
125	<code><th>Penulis</th></code>
126	<code><th>Penerbit</th></code>
127	<code><th>Tahun Terbit</th></code>
128	<code><th>Aksi</th></code>
129	<code></tr></code>
130	<code></thead></code>
131	<code><tbody></code>
132	<code><?php foreach(\$data as \$row): ?></code>
133	<code><tr></code>
134	<code><td><?= \$row['judul']</code>
	<code>?></td></code>
135	<code><td><?= \$row['penulis']</code>
	<code>?></td></code>
136	<code><td><?= \$row['penerbit']</code>
	<code>?></td></code>

137	<td><?=
138	\$row['tahun_terbit'] ?></td>
	<td>
139	<a href="<?=
140	base_url('/editdata/' . \$row['id']) ?>" class="btn btn-sm btn-warning">Edit
	<form action="<?=
141	base_url('/hapusdata/' . \$row['id']) ?>" style="display:inline" method="post">
	<input
142	type="hidden" name="_method" value="DELETE">
	<button class="btn
143	btn-sm btn-danger" type="submit" onclick="return confirm('Apakah anda yakin?')">Hapus</button>
	</form>
144	</td>
145	</tr>
146	<?php endforeach ?>
147	</tbody>
148	</table>
149	<a class="btn btn-success" href="<?=
150	base_url('/tambahdata') ?>">Tambah Data
	</div>
151	</div>
152	</body>
153	</html>
154	

Tabel 7. Source Code index.php

View/create.php

1	<!DOCTYPE html>
---	-----------------

2	<html lang="en">
3	<head>
4	<meta charset="UTF-8">
5	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
6	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
7	<title>Tambah Data</title>
8	<style>
9	body, html {
10	height: 100%;
11	margin: 0;
12	font-family: Arial, Helvetica, sans-serif;
13	display: flex;
14	justify-content: center;
15	align-items: center;
16	background-color: #222222;
17	}
18	.container {
19	width: 500px;
20	max-width: 400px;
21	background-color: white;
22	padding: 20px;
23	border-radius: 10px;
24	text-align: center;
25	}
26	.container h2 {
27	font-size: 35px;
28	margin-bottom: 20px;
29	color: #333;
30	}

31	.container form {
32	width: 100%;
33	}
34	.container form .form-group {
35	margin-bottom: 15px;
36	text-align: left;
37	}
38	.container form .form-group label {
39	display: block;
40	margin-bottom: 5px;
41	color: #666;
42	}
43	.container form .form-group input {
44	width: 100%;
45	padding: 10px;
46	border: 1px solid #ccc;
47	border-radius: 5px;
48	box-sizing: border-box;
49	}
50	.container form .form-group input:focus {
51	border-color: #007bff;
52	outline: none;
53	}
54	.container form .form-group .invalid-feedback
	{
55	color: red;
56	font-size: 12px;
57	display: none;
58	}
59	.container form .form-group .alert {
60	color: #856404;

61	background-color: #fff3cd;
62	border-color: #ffeeba;
63	padding: 10px;
64	border-radius: 5px;
65	margin-bottom: 15px;
66	text-align: left;
67	}
68	.container form .btn {
69	width: 48%;
70	padding: 10px;
71	border: none;
72	border-radius: 5px;
73	color: white;
74	font-size: 16px;
75	cursor: pointer;
76	box-sizing: border-box;
77	margin-top: 10px;
78	}
79	.container form .btn-primary {
80	background-color: #007bff;
81	}
82	.container form .btn-primary:hover {
83	background-color: #0056b3;
84	}
85	.container form .btn-secondary {
86	background-color: #dc3545;
87	}
88	.container form .btn-secondary:hover {
89	background-color: #c82333;
90	}
91	.container .footer {

92	margin-top: 20px;	
93	color: #999;	
94	}	
95	</style>	
96	<script>	
97	function validateForm() {	
98	var isValid = true;	
99		
100	var judul	=
	document.getElementById('judul');	
101	var judulFeedback	=
	judul.nextElementSibling;	
102	if (judul.value.trim() === '') {	
103	judulFeedback.style.display = 'block';	
104	isValid = false;	
105	} else {	
106	judulFeedback.style.display = 'none';	
107	}	
108		
109	var penulis	=
	document.getElementById('penulis');	
110	var penulisFeedback	=
	penulis.nextElementSibling;	
111	if (penulis.value.trim() === '') {	
112	penulisFeedback.style.display	=
	'block';	
113	isValid = false;	
114	} else {	
115	penulisFeedback.style.display	=
	'none';	
116	}	

117		
118	var penerbit	=
	document.getElementById('penerbit');	
119	var penerbitFeedback	=
	penerbit.nextElementSibling;	
120	if (penerbit.value.trim() === '') {	
121	penerbitFeedback.style.display	=
	'block';	
122	isValid = false;	
123	} else {	
124	penerbitFeedback.style.display	=
	'none';	
125	}	
126		
127	var tahun_terbit	=
	document.getElementById('tahun_terbit');	
128	var tahun_terbitFeedback	=
	tahun_terbit.nextElementSibling;	
129	if (tahun_terbit.value.trim() === '') {	
130	tahun_terbitFeedback.style.display	=
	'block';	
131	isValid = false;	
132	} else {	
133	tahun_terbitFeedback.style.display	=
	'none';	
134	}	
135	return isValid;	
136	}	
137	</script>	
138	</head>	
139		

```
140 <body>
141     <div class="container">
142         <?php if(session()->getFlashdata('pesan')): ?>
143             <div class="alert" role="alert">
144                 <?= session()->getFlashdata('pesan')
145             ?>
146             </div>
147         <?php endif?>
148         <?php if(session()->getFlashdata('error')): ?>
149             <div class="alert" role="alert">
150                 <?= session()->getFlashdata('error')
151             ?>
152             </div>
153         <?php endif?>
154         <h2>Tambah Data Buku</h2>
155         <form action="<?= base_url('/tambahdata') ?>"
156             method="post" onsubmit="return validateForm()">
157             <div class="form-group">
158                 <label for="judul">Judul</label>
159                 <input type="text" id="judul"
160                     name="judul">
161                 <div class="invalid-feedback">Judul is
162                     required</div>
163             </div>
164             <div class="form-group">
165                 <label for="penulis">Penulis</label>
166                 <input type="text" id="penulis"
167                     name="penulis">
168                 <div class="invalid-feedback">Penulis
169                     is required</div>
170             </div>
171         </form>
172     </div>
173 </body>
```


164	<code></div></code>
165	<code><div class="form-group"></code>
166	<code><label for="penerbit">Penerbit</label></code>
167	<code><input type="text" id="penerbit"</code>
	<code>name="penerbit"></code>
168	<code><div class="invalid-feedback">Penerbit</code>
	<code>is required</div></code>
169	<code></div></code>
170	<code><div class="form-group"></code>
171	<code><label for="tahun_terbit">Tahun</code>
	<code>Terbit</label></code>
172	<code><input type="text" id="tahun_terbit"</code>
	<code>name="tahun_terbit"></code>
173	<code><div class="invalid-feedback">Tahun</code>
	<code>Terbit is required</div></code>
174	<code></div></code>
175	<code><button type="submit" class="btn btn-</code>
	<code>primary">Tambah</button></code>
176	<code><a href="<?= base_url('/') ?>" class="btn</code>
	<code>btn-secondary">Batal</code>
177	<code></form></code>
178	<code></div></code>
179	<code></body></code>
180	<code></html></code>

Tabel 8. Source Code create.php

View/edit.php

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	
4	<code><head></code>

5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Edit Data</title>
9	<style>
10	body, html {
11	height: 100%;
12	margin: 0;
13	font-family: Arial, Helvetica, sans-serif;
14	display: flex;
15	justify-content: center;
16	align-items: center;
17	background-color: #222222;
18	}
19	.container {
20	width: 500px;
21	max-width: 400px;
22	background-color: white;
23	padding: 20px;
24	border-radius: 10px;
25	text-align: center;
26	}
27	.container h2 {
28	font-size: 35px;
29	margin-bottom: 20px;
30	color: #333;
31	}
32	.container form {
33	width: 100%;

```
34     }
35     .container form .form-group {
36         margin-bottom: 15px;
37         text-align: left;
38     }
39     .container form .form-group label {
40         display: block;
41         margin-bottom: 5px;
42         color: #666;
43     }
44     .container form .form-group input {
45         width: 100%;
46         padding: 10px;
47         border: 1px solid #ccc;
48         border-radius: 5px;
49         box-sizing: border-box;
50     }
51     .container form .form-group input:focus {
52         border-color: #007bff;
53         outline: none;
54     }
55     .container form .form-group .invalid-feedback
56 {
57         color: red;
58         font-size: 12px;
59         display: none;
60     }
61     .container form .form-group .alert {
62         color: #856404;
63         background-color: #fff3cd;
64         border-color: #ffeeba;
```

64	padding: 10px;
65	border-radius: 5px;
66	margin-bottom: 15px;
67	text-align: left;
68	}
69	.container form .btn {
70	width: 48%;
71	padding: 10px;
72	border: none;
73	border-radius: 5px;
74	color: white;
75	font-size: 16px;
76	cursor: pointer;
77	box-sizing: border-box;
78	margin-top: 10px;
79	}
80	.container form .btn-primary {
81	background-color: #007bff;
82	}
83	.container form .btn-primary:hover {
84	background-color: #0056b3;
85	}
86	.container form .btn-secondary {
87	background-color: #dc3545;
88	}
89	.container form .btn-secondary:hover {
90	background-color: #c82333;
91	}
92	.container .footer {
93	margin-top: 20px;
94	color: #999;

95	}	
96	</style>	
97	<script>	
98	function validateForm() {	
99	var isValid = true;	
100		
101	var judul =	
	document.getElementById('judul');	
102	var judulFeedback =	
	judul.nextElementSibling;	
103	if (judul.value.trim() === '') {	
104	judulFeedback.style.display = 'block';	
105	isValid = false;	
106	} else {	
107	judulFeedback.style.display = 'none';	
108	}	
109		
110	var penulis =	
	document.getElementById('penulis');	
111	var penulisFeedback =	
	penulis.nextElementSibling;	
112	if (penulis.value.trim() === '') {	
113	penulisFeedback.style.display	=
	'block';	
114	isValid = false;	
115	} else {	
116	penulisFeedback.style.display	=
	'none';	
117	}	
118		

119	var penerbit	=
	document.getElementById('penerbit');	
120	var penerbitFeedback	=
	penerbit.nextElementSibling;	
121	if (penerbit.value.trim() === '') {	
122	penerbitFeedback.style.display	=
	'block';	
123	isValid = false;	
124	} else {	
125	penerbitFeedback.style.display	=
	'none';	
126	}	
127		
128	var tahun_terbit	=
	document.getElementById('tahun_terbit');	
129	var tahun_terbitFeedback	=
	tahun_terbit.nextElementSibling;	
130	if (tahun_terbit.value.trim() === '') {	
131	tahun_terbitFeedback.style.display	=
	'block';	
132	isValid = false;	
133	} else {	
134	tahun_terbitFeedback.style.display	=
	'none';	
135	}	
136	return isValid;	
137	}	
138	</script>	
139	</head>	
140		
141	<body>	

```

142     <div class="container">
143         <?php if(session()->getFlashdata('pesan')): ?>
144             <div class="alert" role="alert">
145                 <?= session()->getFlashdata('pesan')
146             ?>
147             </div>
148         <?php endif?>
149         <?php if(session()->getFlashdata('error')): ?>
150             <div class="alert" role="alert">
151                 <?= session()->getFlashdata('error')
152             ?>
153             </div>
154         <?php endif?>
155         <h2>Edit Data Buku</h2>
156         <form                                action="<?=
base_url('/editdata/' . $data['id']) ?>"    method="post"
onsubmit="return validateForm()">
157             <div class="form-group">
158                 <label for="judul">Judul</label>
159                 <input        type="text"        id="judul"
value="<?= $data['judul'] ?>" name="judul">
160                 <div class="invalid-feedback">Judul is
required</div>
161             </div>
162             <div class="form-group">
163                 <label for="penulis">Penulis</label>
164                 <input        type="text"        id="penulis"
value="<?= $data['penulis'] ?>" name="penulis">
165                 <div class="invalid-feedback">Penulis
is required</div>

```

165	</div>
166	<div class="form-group">
167	<label for="penerbit">Penerbit</label>
168	<input type="text" id="penerbit" value="<?=\$data['penerbit'] ?>" name="penerbit">
169	<div class="invalid-feedback">Penerbit is required</div>
170	</div>
171	<div class="form-group">
172	<label for="tahun_terbit">Tahun Terbit</label>
173	<input type="text" id="tahun_terbit" value="<?=\$data['tahun_terbit'] ?>" name="tahun_terbit">
174	<div class="invalid-feedback">Tahun Terbit is required</div>
175	</div>
176	<button type="submit" class="btn btn-primary">Update</button>
177	<a href="<?=\$base_url('/') ?>" class="btn btn-secondary">Batal
178	</form>
179	</div>
180	</body>
181	</html>

Tabel 9. Source Code edit.php

Models/BukuModel.php

1	<?php
2	
3	namespace App\Models;


```
4
5 use CodeIgniter\Model;
6
7 class BukuModel extends Model
8 {
9     protected $DBGroup          = 'default';
10    protected $table             = 'buku';
11    protected $primaryKey        = 'id';
12    protected $useAutoIncrement = true;
13    protected $returnType        = 'array';
14    protected $useSoftDeletes    = false;
15    protected $protectFields     = true;
16    protected $allowedFields      = ["judul",
17    "penulis", "penerbit", "tahun_terbit"];
18
19    protected $useTimestamps = false;
20    protected $dateFormat    = 'datetime';
21    protected $createdField  = 'created_at';
22    protected $updatedField  = 'updated_at';
23    protected $deletedField  = 'deleted_at';
24
25    protected $validationRules      = [];
26    protected $validationMessages  = [];
27    protected $skipValidation       = false;
28    protected $cleanValidationRules = true;
29
30    protected $allowCallbacks = true;
31    protected $beforeInsert    = [];
32    protected $afterInsert     = [];
33    protected $beforeUpdate    = [];
34    protected $afterUpdate     = [];
```

34	protected \$beforeFind	= [];
35	protected \$afterFind	= [];
36	protected \$beforeDelete	= [];
37	protected \$afterDelete	= [];
38	}	

Tabel 10. Source Code BukuModel.php

Models/UserModel.php

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class UserModel extends Model
8	{
9	protected \$DBGroup = 'default';
10	protected \$table = 'user';
11	protected \$primaryKey = 'id';
12	protected \$useAutoIncrement = true;
13	protected \$returnType = 'array';
14	protected \$useSoftDeletes = false;
15	protected \$protectFields = true;
16	protected \$allowedFields = ["username", "email","password"];
17	
18	protected \$useTimestamps = false;
19	protected \$dateFormat = 'datetime';
20	protected \$createdField = 'created_at';
21	protected \$updatedField = 'updated_at';
22	protected \$deletedField = 'deleted_at';

23	
24	protected \$validationRules = [];
25	protected \$validationMessages = [];
26	protected \$skipValidation = false;
27	protected \$cleanValidationRules = true;
28	
29	protected \$allowCallbacks = true;
30	protected \$beforeInsert = [];
31	protected \$afterInsert = [];
32	protected \$beforeUpdate = [];
33	protected \$afterUpdate = [];
34	protected \$beforeFind = [];
35	protected \$afterFind = [];
36	protected \$beforeDelete = [];
37	protected \$afterDelete = [];
38	}

Tabel 11. Source Code UserModel.php

prak601.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: 21 Jun 2024 pada 03.44
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  /*!40101  SET  @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
15  */;
16  /*!40101  SET  @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
17  */;
18  /*!40101  SET  @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
19  */;
20  /*!40101 SET NAMES utf8mb4 */;
21
22  --
23  -- Database: `prak601`
24  --
25
26  -- -----
27
28  --
29  -- Struktur dari tabel `buku`
30  --
31
32  CREATE TABLE `buku` (
33    `id` int(5) NOT NULL,
34    `judul` varchar(255) NOT NULL,
35    `penulis` varchar(255) NOT NULL,
36    `penerbit` varchar(255) NOT NULL,
37    `tahun_terbit` year(4) NOT NULL
38  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
39
40  -- -----
41
42  --
43  -- Struktur dari tabel `migrations`
```

```
44  --
45
46  CREATE TABLE `migrations` (
47      `id` bigint(20) UNSIGNED NOT NULL,
48      `version` varchar(255) NOT NULL,
49      `class` varchar(255) NOT NULL,
50      `group` varchar(255) NOT NULL,
51      `namespace` varchar(255) NOT NULL,
52      `time` int(11) NOT NULL,
53      `batch` int(11) UNSIGNED NOT NULL
54      ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
55
56  --
57  -- Dumping data untuk tabel `migrations`
58  --
59
60  INSERT INTO `migrations` (`id`, `version`, `class`, `group`,
61      `namespace`, `time`, `batch`) VALUES
62      (1, '2023-06-10-142756', 'App\\Database\\Migrations\\Crud',
63      'default', 'App', 1718931535, 1),
64      (2, '2023-06-11-060443', 'App\\Database\\Migrations\\User',
65      'default', 'App', 1718931535, 1);
66
67  -- -----
68
69  --
70  -- Struktur dari tabel `user`
71  --
72
73  CREATE TABLE `user` (
74      `id` int(5) NOT NULL,
```

```

71  `username` varchar(100) NOT NULL,
72  `email` varchar(100) NOT NULL,
73  `password` text NOT NULL
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
74
75  --
76  -- Dumping data untuk tabel `user`
77  --
78
79  INSERT INTO `user` (`id`, `username`, `email`, `password`) VALUES
80  (1,
    'Kholid',
    'Kholid123@gmail.com',
    '$2y$10$7ErMI0gLNConCJc2s7Dx.uLF1CtDs9Vcs6bDNHX31St22zWhat/1G');
81
82  --
83  -- Indexes for dumped tables
84  --
85
86  --
87  -- Indeks untuk tabel `buku`
88  --
89  ALTER TABLE `buku`
90  ADD PRIMARY KEY (`id`);
91
92  --
93  -- Indeks untuk tabel `migrations`
94  --
95  ALTER TABLE `migrations`
96  ADD PRIMARY KEY (`id`);
97
98  --
99  -- Indeks untuk tabel `user`

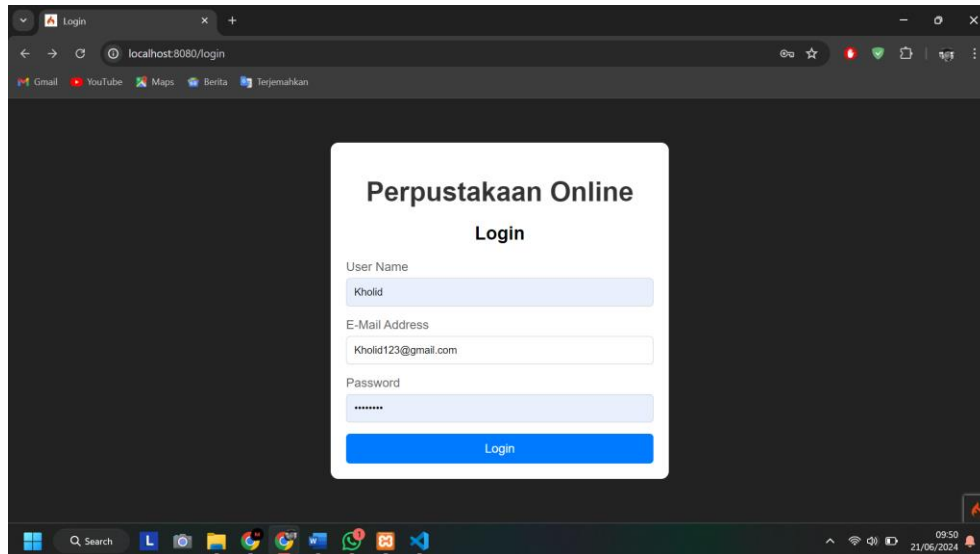
```

```
100 --
101 ALTER TABLE `user`
102     ADD PRIMARY KEY (`id`);
103
104 --
105 -- AUTO_INCREMENT untuk tabel yang dibuang
106 --
107
108 --
109 -- AUTO_INCREMENT untuk tabel `buku`
110 --
111 ALTER TABLE `buku`
112     MODIFY `id` int(5) NOT NULL AUTO_INCREMENT;
113
114 --
115 -- AUTO_INCREMENT untuk tabel `migrations`
116 --
117 ALTER TABLE `migrations`
118     MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
119     AUTO_INCREMENT=3;
120
121 --
122 -- AUTO_INCREMENT untuk tabel `user`
123 --
124 ALTER TABLE `user`
125     MODIFY `id` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
126 COMMIT;
127
128 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
129 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
130 */;
```

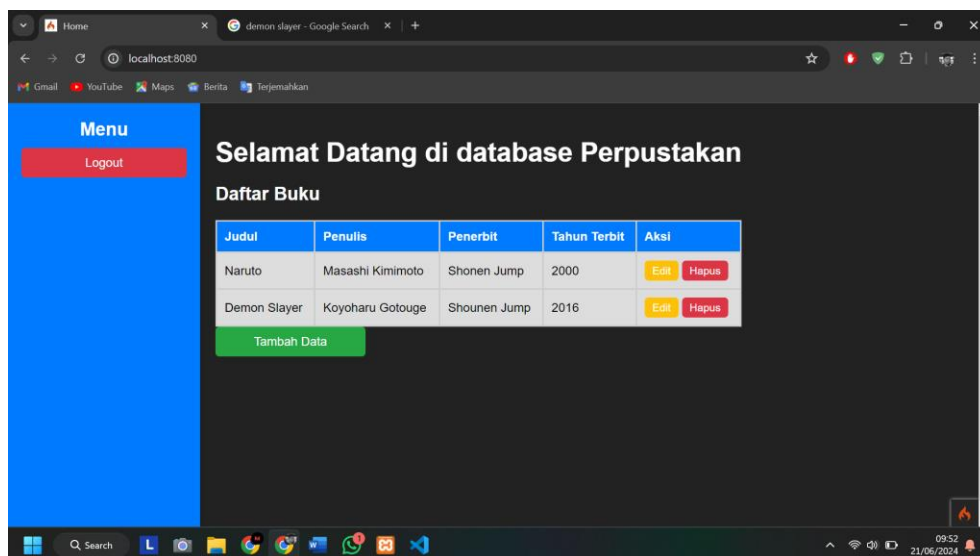
```
129  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Tabel 12. Skema Database prak601.sql

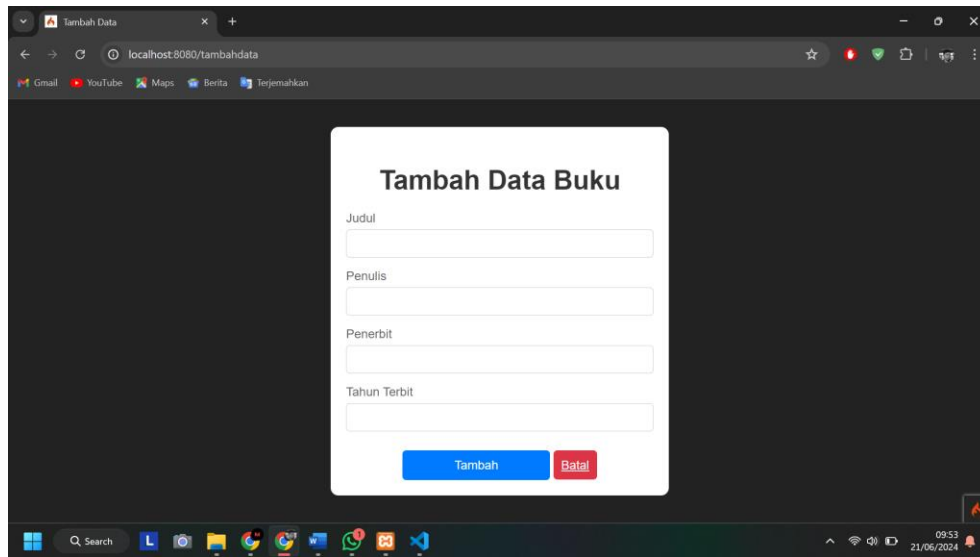
B. Output Program



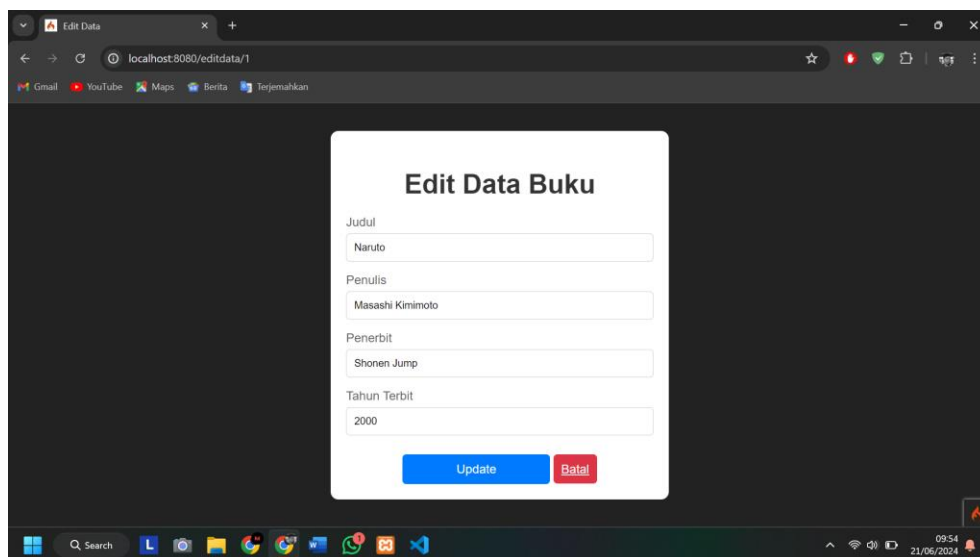
Gambar 2. Screenshot Tampilan Login



Gambar 3. Screenshot Tampilan Home



Gambar 4. Screenshot Tampilan Tambah Data



Gambar 5. Screenshot Tampilan Edit Data

C. Pembahasan

Config/Routes.php

Kode ini mengatur rute-rute (routes) aplikasi web dengan menggunakan layanan `Services::routes()` dari framework CodeIgniter. Namespace default yang digunakan adalah `App\Controllers` dengan controller dan metode default masing-masing adalah `Home` dan `index`. Kode ini juga mengatur agar URI tidak diterjemahkan dengan tanda hubung dan menyiapkan penanganan untuk halaman 404. Ada sekelompok rute yang hanya bisa diakses

jika melewati filter otentikasi (auth), seperti rute untuk menampilkan daftar data (CRUDController::index), menambah data (CRUDController::tambah dan CRUDController::simpan), mengedit data (CRUDController::edit dan CRUDController::update), menghapus data (CRUDController::hapus), dan untuk logout (AuthController::logout). Selain itu, ada juga rute untuk halaman login (AuthController::index dan AuthController::login). Terakhir, jika ada file konfigurasi rute tambahan berdasarkan lingkungan (environment) aplikasi, file tersebut akan dimuat.

Controllers/AuthController.php

AuthController adalah controller yang mengatur proses login dan logout. Metode index() menampilkan halaman login dengan memuat view 'login'. Metode login() mengelola proses autentikasi. Ini melibatkan pengambilan input username, email, dan password dari form login, dan kemudian memeriksa username dalam database menggunakan model UserModel. Jika username ditemukan, kode akan memverifikasi password menggunakan password_verify(). Jika password valid, sesi pengguna akan diatur dengan informasi username, email, dan isLoggedIn, kemudian pengguna akan diarahkan ke halaman utama. Jika tidak valid, pengguna akan diarahkan kembali ke halaman login dengan pesan kesalahan. Metode logout() menghancurkan sesi pengguna dan mengarahkan kembali ke halaman login.

Controllers/BaseController.php

BaseController adalah kelas abstrak yang mewarisi dari kelas Controller di CodeIgniter. Kelas ini mengatur inisialisasi dasar yang diperlukan oleh semua controller dalam aplikasi. Properti \$request dapat berupa CLIRequest atau IncomingRequest, yang mewakili jenis permintaan HTTP yang diterima. Properti \$helpers adalah array yang dapat digunakan untuk memuat helper-helper yang dibutuhkan. Metode initController() digunakan untuk inisialisasi controller dengan mengambil objek RequestInterface, ResponseInterface, dan LoggerInterface. Metode ini memanggil metode initController() dari parent (Controller) untuk memastikan bahwa inisialisasi standar CodeIgniter tetap dijalankan.

Controllers/CRUDController.php

CRUDController adalah controller yang menangani operasi CRUD untuk entitas buku. Metode index() mengambil semua data buku dari model BukuModel dan menampilkannya pada view index. Metode tambah() menampilkan halaman untuk menambah data buku baru. Metode simpan() mengambil data buku dari input pengguna dan menyimpannya ke database menggunakan BukuModel. Metode edit(\$id) mengambil data buku berdasarkan ID dan menampilkannya pada view edit untuk diedit. Metode update(\$id) memperbarui data buku yang ada berdasarkan ID dengan data baru dari input pengguna. Metode hapus(\$id)

menghapus data buku berdasarkan ID dari database. Setelah operasi selesai, pengguna akan diarahkan kembali ke halaman utama.

Filters/AuthFilter.php

AuthFilter adalah kelas yang mengimplementasikan FilterInterface dan digunakan untuk memeriksa status otentikasi pengguna sebelum mereka dapat mengakses bagian tertentu dari aplikasi. Metode before() akan dijalankan sebelum permintaan diproses. Jika sesi tidak memiliki kunci isLoggedIn, filter akan mengatur pesan kesalahan dan mengarahkan pengguna ke halaman login. Metode after() didefinisikan tetapi tidak melakukan apa-apa dan hanya ada untuk melengkapi persyaratan dari FilterInterface. Filter ini memastikan bahwa hanya pengguna yang telah masuk yang dapat mengakses rute yang dilindungi.

Views/login.php

Halaman login ini dirancang dengan HTML, CSS, dan JavaScript untuk aplikasi web. Pada bagian <head>, terdapat pengaturan meta dan style CSS untuk memastikan tampilan halaman yang responsif dan rapi. Halaman ini memiliki container pusat berwarna putih dengan border-radius, yang berisi form login. Form ini memiliki input untuk username, email, dan password, dengan validasi menggunakan JavaScript yang memeriksa apakah input sudah diisi dengan benar. Jika tidak, pesan kesalahan akan ditampilkan di bawah input terkait. Selain itu, terdapat fitur untuk menampilkan pesan flash (flashdata) yang dikirim dari server jika ada kesalahan login sebelumnya, menggunakan PHP. Tombol "Login" mengirim form menggunakan metode POST ke URL login aplikasi. CSS memastikan tampilan yang modern dan responsif, sementara JavaScript menyediakan validasi sisi klien sebelum form dikirim.

Views/index.php

Halaman ini memiliki dua bagian utama: sidebar dan konten utama. Sidebar di sebelah kiri memiliki warna latar belakang biru dan menampilkan menu dengan tombol logout. Konten utama menampilkan pesan sambutan dan tabel daftar buku. Tabel ini berisi kolom untuk judul buku, penulis, penerbit, tahun terbit, serta aksi yang dapat dilakukan pada setiap entri buku seperti edit dan hapus. Tombol 'Tambah Data' berada di bawah tabel untuk menambah entri baru. Desain halaman menggunakan CSS untuk memastikan tampilan yang responsif dan konsisten, dengan warna kontras antara latar belakang gelap dan teks terang untuk meningkatkan keterbacaan. Aksi seperti edit dan hapus diatur menggunakan hyperlink dan form dengan metode POST yang diintegrasikan ke dalam framework PHP, memungkinkan interaksi yang dinamis dengan database.

Views/create.php

Halaman HTML di atas digunakan untuk menambah data buku dalam aplikasi perpustakaan online. Halaman ini menggunakan HTML, CSS, dan JavaScript untuk validasi input sebelum data dikirim. Desainnya responsif dengan latar belakang gelap dan teks putih untuk meningkatkan keterbacaan. Form mengirimkan data ke URL /tambahdata dengan metode POST. Pesan kesalahan atau berhasil ditampilkan menggunakan flash data. Tombol "Tambah" mengirimkan form setelah validasi, sedangkan "Batal" mengarahkan kembali ke halaman utama.

Views/edit.php

Halaman HTML di atas digunakan untuk mengedit data buku dalam aplikasi perpustakaan online. Halaman ini memiliki desain yang responsif dengan latar belakang gelap dan menggunakan warna putih untuk teksnya agar mudah dibaca. Form ini mengirimkan data ke URL /editdata/{id} dengan metode POST untuk mengupdate informasi buku. Setiap input di dalam form sudah diisi dengan data yang ada sebelumnya menggunakan nilai dari variabel \$data. Validasi JavaScript dilakukan sebelum mengirimkan data, dan pesan kesalahan atau notifikasi berhasil ditampilkan menggunakan flash data dari session PHP. Tombol "Update" mengirimkan form setelah validasi, sedangkan "Batal" mengarahkan kembali ke halaman utama.

Models/BukuModel.php

Model BukuModel dalam framework CodeIgniter digunakan untuk mengelola data buku dalam database. Model ini terhubung dengan tabel buku dan mendefinisikan kunci utama sebagai id dengan penomoran otomatis. Properti \$allowedFields menentukan kolom yang dapat dimanipulasi, yaitu "judul", "penulis", "penerbit", dan "tahun_terbit". Model ini tidak secara otomatis mengelola timestamp (\$useTimestamps diatur sebagai false) dan tidak menerapkan validasi data bawaan. Pengguna dapat menggunakan callback sebelum atau setelah operasi CRUD untuk menyesuaikan perilaku model sesuai kebutuhan aplikasi.

Models/UserModel.php

Model UserModel digunakan untuk mengelola data pengguna dalam tabel user di dalam database. Model ini terhubung dengan grup database default (default) dan telah dikonfigurasi dengan berbagai properti seperti nama tabel (user), kunci utama (id), penggunaan auto-increment untuk kunci utama (true), jenis data yang diharapkan dari hasil query (array), serta field-field yang diizinkan untuk dimanipulasi seperti username, email, dan password. Penggunaan timestamp untuk kolom created_at, updated_at, dan deleted_at dinonaktifkan (false). Validasi data dan callback sebelum dan sesudah operasi database juga disediakan untuk memungkinkan penggunaan tambahan seperti validasi kustom dan tindakan sebelum atau setelah operasi database.

prak601.sql

SQL ini mencakup struktur untuk dua tabel utama: buku dan user. Tabel buku menyimpan informasi judul, penulis, penerbit, dan tahun terbit buku. Sementara itu, tabel user digunakan untuk menyimpan data pengguna dengan informasi seperti username, email, dan password yang di-hash menggunakan bcrypt. Selain itu, terdapat tabel migrations untuk melacak versi migrasi database yang telah dieksekusi dalam aplikasi CodeIgniter.

Tautan Git

Berikut adalah tautan untuk semua source code yang telah dibuat.

https://github.com/MKsafr/Pemrograman_Web_II.git