

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Muh. Haekal Barera NIM. 2210817310008

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 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 : Muh. Haekal Barera
NIM : 2210817310008

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	6
A. Source Code	7
B. Output Program	51
C. Pembahasan	53
LINK GITHUB.....	69

DAFTAR GAMBAR

Gambar Modul 6. 1 Tampilan register.....	51
Gambar Modul 6. 2 Tampilan login.....	51
Gambar Modul 6. 3 Tampilan home	52
Gambar Modul 6. 4 Tampilan tambah buku	52
Gambar Modul 6. 5 Tampilan edit.....	53

DAFTAR TABEL

Table 1. Code Routes.php	7
Table 2. Code Buku.php di Controller	11
Table 3. Code Home.php	12
Table 4. Code Login.php.....	14
Table 5. Code Register.php	16
Table 6. Code AuthFilter.php.....	17
Table 7. Code vw_add.php.....	21
Table 8. Code vw_buku.php	29
Table 9. Code vw_edit.php	34
Table 10. Code vw_login.php	38
Table 11. Code vw_register.php	42
Table 12. Code BukuModel.php	44
Table 13. Code UserModel.php	44
Table 14. Code Buku.php di Seeds	45
Table 15. Code UserSeeder.php.....	46
Table 16. Code perpustakaan.sql	50

SOAL

Soal Praktikum:

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. Contoh database soal 1

Simpan dengan nama projek di Github: PRAK601

A. Source Code

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');
	\$routes->get('/register', 'Register::index');
	\$routes->post('/register/process', 'Register::process');
	\$routes->get('/login', 'Login::index');
	\$routes->post('/login/process', 'Login::process');
	\$routes->get('/logout', 'Login::logout');
	\$routes->get('/buku/create', 'Buku::create');
	\$routes->post('buku/store', 'Buku::store');
	\$routes->post('buku/update/', 'Buku::update', ['filter'
	=> 'auth']);
	\$routes->get('buku/edit/', 'Buku::edit', ['filter' =>
	'auth']);
	\$routes->get('buku/delete/', 'Buku::delete', ['filter'
	=> 'auth']);

Table 1. Code Routes.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      * @var Model
11      */
12     protected $model;
13
14     public function __construct()
15     {
16         $this->model = new BukuModel();
17         $this->helpers = ['form', 'url'];
18     }
19
20     public function login()
21     {
22         $data = [
23             'posts' => $this->model->paginate(10),
24             'pager' => $this->model->pager,
25             'title' => 'POST LIST'
26         ];
27
28         return view('vw_login', $data);
29     }
30
31     public function index()
32     {
33         if (!session()->get('logged_in')) {
34             return redirect()->to(base_url('/login'));
35         }
36         $data = [
37             'posts' => $this->model->paginate(10),
38             'pager' => $this->model->pager,
39             'title' => 'POST LIST'
40         ];
41
42         return view('vw_buku', $data);
```



```

41     }
42
43     public function create()
44     {
45         // Check if user is authenticated
46         if (!session()->get('logged_in')) {
47             return redirect()->to(base_url('/login'));
48         }
49         $data = ['title' => 'Create new post'];
50
51         return view('vw_add', $data);
52     }
53
54     public function store()
55     {
56         $data = $this->request->getPost(['judul',
57 'penulis', 'penerbit', 'tahun_terbit']);
58
59         if (! $this->validateData($data, $this->model-
60 >validationRules)) {
61             return $this->create();
62         }
63
64         $post = $this->validator->getValidated();
65
66         $save = $this->model->save([
67             'judul' => $post['judul'],
68             'penulis' => $post['penulis'],
69             'penerbit' => $post['penerbit'],
70             'tahun_terbit' => $post['tahun_terbit'],
71         ]);
72
73         session()->setFlashdata('success', 'Post has been
74 added successfully. ');
75         return redirect()->to(base_url('home'));
76     }
77
78     public function edit($id)
79     {
80         // Check if user is authenticated
81         if (!session()->get('logged_in')) {
82             return redirect()->to(base_url('/login'));
83         }
84
85         $post = $this->model->find($id);

```

```

81         if (empty($post)) {
82             session()->setFlashdata('error','Post not
83 found');
84             return redirect()->back();
85         }
86         $data = [
87             'title' => 'Edit Post',
88             'post' => $post
89         ];
90         return view('vw_edit', $data);
91     }
92
93     public function update($id)
94     {
95         // Check if user is authenticated
96         if (!session()->get('logged_in')) {
97             return redirect()->to(base_url('/login'));
98         }
99         $post = $this->model->find($id);
100
101         if (empty($post)) {
102             session()->setFlashdata('error','Post not
103 found');
104             return redirect()->back();
105         }
106
107         $data = $this->request->getPost(['judul',
108 'penulis', 'penerbit', 'tahun_terbit']);
109
110         if (! $this->validateData($data, $this->model-
111 >validationRules)) {
112             return $this->create();
113         }
114
115         $updatedPost = $this->validator->getValidated();
116
117         $update = $this->model->update($post['id'],
118 $updatedPost);
119
120         if ($update) {
121             session()->setFlashdata('success', 'Post has
122 been updated successfully');
123             return redirect()->to(base_url('home'));

```

121	} else {
122	session()->setFlashdata('error', 'Some
123	problems occured, please try again.');
124	return redirect()->back();
125	}
126	}
127	public function delete(\$id)
128	{
129	// Check if user is authenticated
130	if (!session()->get('logged_in')) {
131	return redirect()->to(base_url('home'));
132	}
133	
134	if (empty(\$id)) {
135	return redirect()->to(base_url('home'));
136	}
137	
138	\$delete = \$this->model->delete(\$id);
139	
140	if (\$delete) {
141	session()->setFlashdata('success', 'Post has
142	been removed successfully.');
143	return redirect()->to(base_url('home'));
144	} else {
145	session()->setFlashdata('error', 'Some
146	problems occured, please try again.');
147	return redirect()->to(base_url('home'));
148	}
149	}
150	}
151	
152	
153	
154	
155	

Table 2. Code Buku.php di Controller

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. Code Home.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', $username)-
20 >first();
21
22         if ($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', 'Username
34 & Password Salah');
35                 return redirect()->to(base_url('login'));
36             }
37         } else {
38             session()->setFlashdata('error', 'Username &
39 Password Salah');
40             return redirect()->to(base_url('login'));
41         }
42     }
43
44     public function logout()
```

41	{
42	session()->destroy();
43	return redirect()->to(base_url('login'));
44	}
45	}

Table 4. Code Login.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         // Validate input
17         if (!$this->validate([
18             'username' => [
19                 'rules' =>
20                 'required|min_length[4]|max_length[20]|is_unique[users.username]',
21                 'errors' => [
22                     'required' => '{field} Harus diisi',
23                     'min_length' => '{field} Minimal 4 Karakter',
24                     'max_length' => '{field} Maksimal 20 Karakter',
25                     'is_unique' => 'Username sudah digunakan
26                     sebelumnya'
27                 ]
28             ],
29             'password' => [
30                 'rules' =>
31                 'required|min_length[0]|max_length[50]',
32                 'errors' => [
33                     'required' => '{field} Harus diisi',
34                     'min_length' => '{field} Minimal 0 Karakter',
35                     'max_length' => '{field} Maksimal 50 Karakter',
36                 ]
37             ],
38             'password_conf' => [
39                 'rules' => 'matches[password]',
40                 'errors' => [
41                     'matches' => 'Konfirmasi Password tidak sesuai
42                     dengan password',
43                 ]
44             ]
45         ])) {
46             return view('vw_register');
47         }
48         // Proses register
49         $usersModel = new UsersModel();
50         $usersModel->save($this->request->getPost('username'), $this->request->getPost('password'));
51         return view('vw_register');
52     }
53 }

```

```

41     ],
42     'email' => [
43                                     'rules'          =>
44 'required|min_length[4]|max_length[100]|valid_email',
45     'errors' => [
46         'required' => '{field} Harus diisi',
47         'min_length' => '{field} Minimal 4 Karakter',
48         'max_length' => '{field} Maksimal 100 Karakter',
49         'valid_email' => 'Email tidak valid'
50     ]
51 ],
52 )) {
53     session()->setFlashdata('error', $this->validator-
54 >listErrors());
55     return redirect()->back()->withInput();
56 }
57
58 $users = new UsersModel();
59
60 // Hash the password before storing it
61 $hashedPassword = password_hash($this->request-
62 >getVar('password'), PASSWORD_DEFAULT);
63
64 $users->insert([
65     'username' => $this->request->getVar('username'),
66     'password' => $hashedPassword,
67     'email' => $this->request->getVar('email')
68 ]);
69
70 return redirect()->to('/login');
71 }
72 }

```


1	<?php
2	// app/Filters/AuthFilter.php
3	
4	namespace App\Filters;
5	
6	use CodeIgniter\HTTP\RequestInterface;
7	use CodeIgniter\HTTP\ResponseInterface;
8	use CodeIgniter\Filters\FilterInterface;
9	class AuthFilter implements FilterInterface
10	{
11	public function before(RequestInterface \$request,
12	\$args = null)
13	{
14	// Check if the user is logged in
15	if (!session()->has('user_id')) {
16	// If not logged in, redirect to the login
17	page return redirect()->to('/login');
18	}
19	}
20	
21	public function after(RequestInterface \$request,
22	ResponseInterface \$response, \$args = null)
23	{
24	// Do nothing after the request
25	}
	}

Table 6. Code AuthFilter.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	<style>
5	body {
6	background-color: #121212;
7	background-repeat: no-repeat;
8	background-attachment: fixed;
9	background-position: center;
10	background-size: cover;
11	color: #ffffff;
12	}
13	
14	.card {
15	background-color: #1e1e1e;
16	color: #ffffff;
17	}
18	
19	.card-header {
20	background-color: #6200ea;
21	}
22	
23	.form-control {
24	background-color: #2c2c2c;
25	color: #ffffff;
26	border: 1px solid #444;
27	}
28	
29	.form-control::placeholder {
30	color: #888;
31	}
32	
33	.form-group label {
34	color: #ffffff;
35	}
36	
37	.btn-success {
38	background-color: #bb86fc;
39	border-color: #bb86fc;
40	color: #000000;

41	}
42	
43	.btn-success:hover {
44	background-color: #3700b3;
45	border-color: #3700b3;
46	color: #ffffff;
47	}
48	
49	.btn-secondary {
50	background-color: #333;
51	border-color: #333;
52	color: #ffffff;
53	}
54	
55	.btn-secondary:hover {
56	background-color: #555;
57	border-color: #555;
58	color: #ffffff;
59	}
60	
61	.alert-danger {
62	background-color: #ff4444;
63	color: #ffffff;
64	border: none;
65	}
66	</style>
67	
68	<div class="container mt-5">
69	<div class="row justify-content-center">
70	<div class="col-md-8">
71	<div class="card shadow-sm">
72	<div class="card-header text-white">
73	<h4>Tambah</h4>
74	</div>
75	<div class="card-body">
76	<?php if (session()-
77	>getFlashdata('error')) : ?>
78	<div class="alert alert-danger">
	<?php echo session()-
	>getFlashdata('error'); ?>

```

79         </div>
80     <?php endif; ?>
81
82     <?= validation_list_errors() ?>
83
84     <?= form_open('buku/store'); ?>
85     <div class="form-group">
86         <label for="judul">Judul</label>
87         <input type="text" class="form-
control" id="judul" name="judul" placeholder="Masukkan
Judul">
88     </div>
89     <div class="form-group">
90         <label
for="penulis">Penulis</label>
91         <input type="text" class="form-
control" id="penulis" name="penulis"
placeholder="Masukkan Penulis">
92     </div>
93     <div class="form-group">
94         <label
for="penerbit">Penerbit</label>
95         <input type="text" class="form-
control" id="penerbit" name="penerbit"
placeholder="Masukkan Penerbit">
96     </div>
97     <div class="form-group">
98         <label for="tahun_terbit">Tahun
Terbit</label>
99         <input type="number" class="form-
control" id="tahun_terbit" name="tahun_terbit"
placeholder="Masukkan Tahun Terbit">
100     </div>
101     <div class="form-group text-right">
102         <button class="btn btn-
success">Simpan</button>
103         <a href="<?= base_url('home') ?>"
class="btn btn-secondary">Kembali</a>
104     </div>
105     <?= form_close(); ?>

```

114	</div>
115	</div>
116	</div>
117	</div>
118	</div>
119	
120	<?= \$this->endSection() ?>
121	
122	<?= \$this->section('extra-js') ?>
123	<!-- include summernote css/js -->
124	<link
125	href="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dis
126	t/summernote-bs4.min.css" rel="stylesheet">
127	<script
128	src="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist
129	/summernote-bs4.min.js"></script>
130	<script>
131	\$(document).ready(function() {
132	\$('#post_content').summernote({
133	tabsize: 2,
134	height: 500
135	});
136	});
137	</script>
138	<?= \$this->endSection() ?>

Table 7. Code vw_add.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	<style>
5	body {
6	background-color: #121212;
7	background-repeat: no-repeat;
8	background-attachment: fixed;
9	background-position: center;
10	background-size: cover;
11	color: #ffffff;
12	}
13	
14	.navbar-light {
15	background-color: #1f1f1f;
16	}
17	
18	.navbar-light .navbar-brand,
19	.navbar-light .navbar-nav .nav-link {
20	color: #ffffff;
21	}
22	
23	.navbar-toggler {
24	border-color: rgba(255, 255, 255, 0.1);
25	}
26	
27	.navbar-toggler-icon {
28	background-image:
29	url("data:image/svg+xml;charset=utf8,%3Csvg
30	viewBox='0 0 30 30'
31	xmlns='http://www.w3.org/2000/svg'%3E%3Cpath
32	stroke='rgba%2855, 55, 55, 0.7%29' stroke-width='2'
33	
34	
35	
36	
37	
38	
39	
40	

```
41 stroke-linecap='round' stroke-miterlimit='10' d='M4
42 7h22M4 15h22M4 23h22'/%3E%3C/svg%3E");
43     }
44
45     .card {
46         background-color: #1e1e1e;
47         color: #ffffff;
48     }
49
50     .card-header {
51         background-color: #6200ea;
52     }
53
54     .btn {
55         border: none;
56     }
57
58     .btn-sm {
59         background-color: #bb86fc;
60         color: #000000;
61     }
62
63     .btn-sm:hover {
64         background-color: #3700b3;
65         color: #ffffff;
66     }
67
68     .alert-success {
69         background-color: #00c851;
70         color: #ffffff;
71         border: none;
72     }
73
74     .alert-danger {
75         background-color: #ff4444;
76         color: #ffffff;
77         border: none;
78     }
79
80     .table {
```

```
81         background-color: #1e1e1e;
82         color: #ffffff;
83     }
84
85     .table thead {
86         background-color: #6200ea;
87         color: #ffffff;
88     }
89
90     .table-bordered,
91     .table-bordered td,
92     .table-bordered th {
93         border-color: #444;
94     }
95
96     .pagination .page-item .page-link {
97         background-color: #1e1e1e;
98         color: #ffffff;
99         border-color: #6200ea;
100     }
101
102     .pagination .page-item .page-link:hover {
103         background-color: #6200ea;
104         color: #ffffff;
105         border-color: #3700b3;
106     }
107 </style>
108
109 <!-- Navbar -->
110 <nav class="navbar navbar-expand-lg navbar-light bg-
111 light shadow-sm">
112     <div class="container">
113         <a          class="navbar-brand"          href="<?=
114 base_url('/') ?>">My Library</a>
115         <button          class="navbar-toggler"
116 type="button"          data-toggle="collapse"          data-
117
118
119
120
```



```

121 target="#navbarNav" aria-controls="navbarNav" aria-
122 expanded="false" aria-label="Toggle navigation">
123     <span class="navbar-toggler-
124 icon"></span>
125     </button>
126     <div class="collapse navbar-collapse"
127 id="navbarNav">
128         <ul class="navbar-nav ml-auto">
129             <?php if (session()-
130 >get('logged_in')) : ?>
131                 <li class="nav-item">
132                     <a class="nav-link"
133 href="#">Welcome, <?= session()->get('username');
134 ?></a>
135                 </li>
136                 <li class="nav-item">
137                     <a class="nav-link"
138 href="<?= base_url('login/logout'); ?>">Logout</a>
139                 </li>
140             <?php else : ?>
141                 <li class="nav-item">
142                     <a class="nav-link"
143 href="<?= base_url('login'); ?>">Login</a>
144                 </li>
145                 <li class="nav-item">
146                     <a class="nav-link"
147 href="<?= base_url('register'); ?>">Register</a>
148                 </li>
149             <?php endif; ?>
150         </ul>
151     </div>
152 </div>
153 </nav>
154
155 <div class="container mt-5">
156     <div class="row justify-content-center">
157         <div class="col-md-10">
158             <div class="card shadow-sm">
159                 <div class="card-header text-
160 white">

```

```

161         <h5 class="mb-0">List Buku</h5>
162         <a href="<?=
163 base_url('buku/create'); ?>" class="btn btn-sm
164 float-right">Tambah Data</a>
165     </div>
166     <div class="card-body">
167         <?php if (session() -
168 >getFlashdata('success')) : ?>
169             <div class="alert alert-
170 success">
171                 <?= session() -
172 >getFlashdata('success'); ?>
173             </div>
174             <?php endif; ?>
175
176         <?php if (session() -
177 >getFlashdata('error')) : ?>
178             <div class="alert alert-
179 danger">
180                 <?= session() -
181 >getFlashdata('error'); ?>
182             </div>
183             <?php endif; ?>
184
185         <table class="table table-
186 bordered table-striped table-hover">
187             <thead class="text-center">
188                 <tr>
189                     <th
190 scope="col">Judul</th>
191                     <th
192 scope="col">Penulis</th>
193                     <th
194 scope="col">Penerbit</th>
195                     <th
196 scope="col">Tahun Terbit</th>
197                     <th
198 scope="col">Action</th>
199                 </tr>
200             </thead>

```

201	<tbody>
202	<?php if
203	(!empty(\$posts) && is_array(\$posts)) : ?>
204	<?php foreach
205	(\$posts as \$row) : ?>
	<tr>
	<td><?=
	\$row['judul']; ?></td>
	<td><?=
	\$row['penulis']; ?></td>
	<td><?=
	\$row['penerbit']; ?></td>
	<td><?=
	\$row['tahun_terbit']; ?></td>
	<td
	class="text-center">
	<form
	onsubmit="return confirm('Apakah Anda Yakin ?');"

	<pre> action="<?= base_url('buku/delete/' . \$row['id']); ?>" method="POST"> <input type="hidden" name="{csrf_token}" value="{csrf_hash}"> <input type="hidden" name="_method" value="DELETE"> <a href="<?= base_url('buku/edit/' . \$row['id']); ?>" class="btn btn-warning btn-sm">Edit <button type="submit" class="btn btn-danger btn- sm">Delete</button> </form> </td> </tr> </table> </pre>
--	---

	<pre> <script> \$(document).ready(function() { \$('.pagination li').addClass('page-item'); \$('.pagination li a').addClass('page- link'); }); </script> <?= \$this->endSection() ?> </pre>
--	---

Table 8. Code vw_buku.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	<style>
5	body {
6	background-color: #121212;
7	background-repeat: no-repeat;
8	background-attachment: fixed;
9	background-position: center;
10	background-size: cover;
11	color: #ffffff;
12	}
13	.card {
14	background-color: #1e1e1e;
15	color: #ffffff;
16	}
17	.card-header {
18	background-color: #6200ea;
19	}
20	
21	.form-control {
22	background-color: #2c2c2c;
23	color: #ffffff;
24	border: 1px solid #444;
25	}
26	.form-control::placeholder {
27	color: #888;
28	}
29	
30	.form-group label {
31	color: #ffffff;
32	}
33	
34	.btn-success {
35	background-color: #bb86fc;
36	border-color: #bb86fc;
37	color: #000000;
38	}
39	.btn-success:hover {
40	background-color: #3700b3;

```

41         border-color: #3700b3;
42         color: #ffffff;
43     }
44
45     .btn-secondary {
46         background-color: #333;
47         border-color: #333;
48         color: #ffffff;
49     }
50
51     .btn-secondary:hover {
52         background-color: #555;
53         border-color: #555;
54         color: #ffffff;
55     }
56
57     .alert-danger {
58         background-color: #ff4444;
59         color: #ffffff;
60         border: none;
61     }
62 </style>
63
64 <div class="container mt-5">
65     <div class="row justify-content-center">
66         <div class="col-md-8">
67             <div class="card shadow-sm">
68                 <div class="card-header text-white">
69                     <h4>Edit</h4>
70                 </div>
71                 <div class="card-body">
72
73                     <?php if (session()->getFlashdata('error'))
74 : ?>
75
76                         <div class="alert alert-danger">
77                             <?=
78                                 session()-
79                                 >getFlashdata('error'); ?>
80                             </div>
81
82                     <?php endif; ?>
83
84                     <?= validation_list_errors() ?>
85
86                     <?=
87                         form_open('buku/update/'
88
89
90 $post['id']); ?>

```

81	<input <="" name="_method" td="" type="hidden"/>
82	value="PUT">
83	
84	<div ><="" class="form-group" td=""></div>
85	<label >judul<="" for="judul" label><="" td=""></label>
86	<input <="" class="form-control" td="" type="text"/>
87	id="judul" name="judul" placeholder="Masukkan Judul" value="<?=\$post['judul'] ?>">
88	</div>
89	<div ><="" class="form-group" td=""></div>
90	<label >penulis<="" for="penulis" label><="" td=""></label>
91	<input <="" class="form-control" td="" type="text"/>
92	id="penulis" name="penulis" placeholder="Masukkan Penulis"
93	value="<?=\$post['penulis'] ?>">
94	</div>
95	<div ><="" class="form-group" td=""></div>
96	<label >penerbit<="" for="penerbit" label><="" td=""></label>
97	<input <="" class="form-control" td="" type="text"/>
98	id="penerbit" name="penerbit" placeholder="Masukkan Penerbit"
99	value="<?=\$post['penerbit'] ?>">
100	</div>
101	<div ><="" class="form-group" td=""></div>
102	<label >tahun<="" for="tahun_terbit" td=""></label>
103	Terbit</label>
104	<input <="" class="form-control" id="tahun_terbit" name="tahun_terbit" td="" type="number"/>
105	
106	
107	
108	
109	
110	
111	
112	
113	
114	
115	
116	
117	
118	
119	
120	


```

121 placeholder="Masukkan Tahun Terbit" value="<?=  

122 $post['tahun_terbit'] ?>">  

123 </div>  

124 <div class="form-group text-right">  

125 <button class="btn btn-  

126 success">Update</button>  

127 <a href="<?=  

128 class="btn btn-secondary">Kembali</a>  

129 </div>  

130 <?= form_close(); ?>  

131 </div>  

132 </div>  

133 </div>  

134 </div>  

135 </div>  

136 <?= $this->endSection() ?>  

137 <?= $this->section('extra-js') ?>  

138 <!-- include summernote css/js -->  

139 <link  

140 href="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist/summe  

141 rnote-bs4.min.css" rel="stylesheet">  

142 <script  

143 src="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist/summer  

144 note-bs4.min.js"></script>  

145 <script>  

146 $(document).ready(function() {  

147 $('#post_content').summernote({  

148 tabsize: 2,  

149 height: 500,  

150 styleTags: [  

151 'p', 'blockquote', 'pre', 'h1', 'h2', 'h3',  

152 'h4', 'h5', 'h6'  

153 ],  

154 toolbar: [  

155 ['style', ['style']],  

156 ['font', ['bold', 'italic', 'underline',  

157 'clear']],  

158 ['fontname', ['fontname']],  

159 ['color', ['color']],  

160 ['para', ['ul', 'ol', 'paragraph']],  

161 ['table', ['table']],

```

	<pre> ['insert', ['link', 'picture', 'video']], ['view', ['fullscreen', 'codeview', 'help']]] }); }); </script> <?= \$this->endSection() ?></pre>
--	--

Table 9. Code `vw_edit.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=1.0">
8	<title>Login</title>
9	<style>
10	body {
11	display: flex;
12	justify-content: center;
13	align-items: center;
14	min-height: 100vh;
15	padding: 0;
16	margin: 0;
17	font-family: Arial, sans-serif;
18	background-color: #121212;
19	background-repeat: no-repeat;
20	background-attachment: fixed;
21	background-position: center;
22	background-size: cover;
23	}
24	.form-signin {
25	width: 100%;
26	max-width: 330px;
27	padding: 15px;
28	margin: auto;
29	background: #1e1e1e;
30	border-radius: 8px;
31	box-shadow: 0 4px 6px rgba(0, 0, 0, 1);
32	position: relative;
33	color: #ffffff;
34	text-align: center;
35	}
36	.form-signin img {
37	margin-bottom: 20px;
38	}
39	.form-signin h1 {
40	margin-bottom: 10px;
	font-size: 24px;

```
41         font-weight: normal;
42         color: #ffffff;
43     }
44
45     .form-signin p {
46         margin-bottom: 20px;
47         font-size: 14px;
48         color: #b3b3b3;
49     }
50
51     .form-signin input[type="text"],
52     .form-signin input[type="password"] {
53         width: 100%;
54         padding: 10px;
55         margin-bottom: 10px;
56         border: 1px solid #333;
57         border-radius: 4px;
58         box-sizing: border-box;
59         background-color: #333;
60         color: #ffffff;
61     }
62
63     .form-signin button {
64         width: 100%;
65         padding: 10px;
66         background-color: #6200ea;
67         color: white;
68         border: none;
69         border-radius: 4px;
70         cursor: pointer;
71         margin-bottom: 10px;
72         transition: background-color 0.3s ease;
73     }
74
75     .form-signin button:hover {
76         background-color: #3700b3;
77     }
78
79     .btn-register {
80         display: block;
81         width: 100%;
82         padding: 10px;
83         font-size: 14px;
84         color: white;
```

```

81         cursor: pointer;
82         background: #03dac6;
83         border: none;
84         border-radius: 4px;
85         margin-top: 10px;
86         transition: background-color 0.3s ease;
87     }
88     .btn-register:hover {
89         background-color: #018786;
90     }
91
92     .form-signin .alert {
93         padding: 10px;
94         margin-bottom: 10px;
95         border-radius: 4px;
96         color: #856404;
97         background-color: rgba(255, 253, 220, 0.1);
98         border: 1px solid #ffeeba;
99         display: none;
100        position: relative;
101    }
102
103    .form-signin .text-muted {
104        color: #6c757d;
105        margin-top: 20px;
106    }
107    </style>
108    </head>
109    <body>
110        <main class="form-signin">
111            <!-- Alert untuk informasi saat halaman dimuat -->
112            <div class="alert" id="initial-alert" style="display:
113            block;">
114                Login terlebih dahulu!
115            </div>
116
117            <!-- Alert untuk kesalahan login -->
118            <div class="alert" style="display: <?php echo
119            (session()->getFlashdata('error')) ? 'block' : 'none'; ?>";
120                <?= session()->getFlashdata('error'); ?>
121            </div>

```

121	
122	<form method="post" action="<?= base_url();
123	?>/login/process">
124	<?= csrf_field(); ?>
125	<h1 class="h3 mb-3 fw-normal">Login</h1>
126	<p>Register dulu apabila belum memiliki akun</p>
127	
128	<input type="text" name="username" id="username"
129	placeholder="Username" required autofocus>
130	<input type="password" name="password"
131	id="password" placeholder="Password" required>
132	
133	<button type="submit">Login</button>
134	<button type="button" class="btn-register"
135	onclick="window.location.href='<?php echo
136	base_url('register'); ?>'>Register</button>
137	</form>
138	</main>
139	</body>
	</html>

Table 10. Code vw_login.php

1	<!doctype html>
2	<html lang="en" class="h-100">
3	
4	<head>
5	<meta charset="utf-8">
6	<meta name="viewport" content="width=device-width,
7	initial-scale=1">
8	<title>Register</title>
9	
10	<!-- Custom styles for this template -->
11	<style>
12	body,
13	html {
14	margin: 0;
15	padding: 0;
16	font-family: Arial, sans-serif;
17	height: 100%;
18	display: flex;
19	flex-direction: column;
20	background-color: black;
21	background-repeat: no-repeat;
22	background-attachment: fixed;
23	background-position: center;
24	background-size: cover;
25	background-color: #121212;
26	padding: 20px; /* Add padding to body */
27	color: #ffffff; /* Default text color */
28	}
29	
30	.container {
31	max-width: 600px;
32	margin: auto;
33	padding: 20px;
34	background: #1e1e1e;
35	border-radius: 8px;
36	box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
37	margin-top: 40px;
38	color: #ffffff;
39	}
40	
	h1 {
	font-size: 24px;
	margin-bottom: 20px;
	text-align: center;

41	color: #ffffff;
42	}
43	
44	.form-label {
45	display: block;
46	margin-bottom: 5px;
47	font-weight: bold;
48	color: #ffffff;
49	}
50	.form-control {
51	width: 100%;
52	padding: 10px;
53	margin-bottom: 10px;
54	border: 1px solid #333;
55	border-radius: 4px;
56	box-sizing: border-box; /* Ensure padding is
57	included in width calculation */
58	background-color: #333;
59	color: #ffffff;
60	}
61	.btn {
62	width: 100%;
63	padding: 10px;
64	background-color: #6200ea;
65	color: white;
66	border: none;
67	border-radius: 4px;
68	cursor: pointer;
69	transition: background-color 0.3s ease;
70	}
71	.btn:hover {
72	background-color: #3700b3;
73	}
74	
75	.alert {
76	padding: 10px;
77	margin-bottom: 10px;
78	border-radius: 4px;
79	color: #856404;
80	background-color: rgba(255, 253, 220, 0.1);
	border: 1px solid #ffeeba;


```

81     }
82
83     .footer {
84         margin-top: auto;
85         padding: 10px 0;
86         text-align: center;
87         background-color: #1e1e1e;
88         color: #ffffff;
89     }
90
91     .footer .container {
92         max-width: none;
93     }
94 </style>
95 </head>
96 <body>
97     <main class="flex-shrink-0">
98         <div class="container">
99             <h1>Register Form</h1>
100             <hr />
101             <?php      if      (!empty(session()-
102 >getFlashdata('error')) : ?>
103                 <div class="alert">
104                     <h4>Periksa Entrian Form</h4>
105                     <?=      session()-
106 >getFlashdata('error'); ?>
107                 </div>
108                 <?php endif; ?>
109                 <form method="post" action="<?= base_url();
110 ?>/register/process">
111                     <?= csrf_field(); ?>
112                     <div class="mb-3">
113                         <label for="username" class="form-
114 label">Username</label>
115                         <input type="text" class="form-
116 control" id="username" name="username" required>
117                     </div>
118                     <div class="mb-3">
119                         <label for="password" class="form-
120 label">Password</label>
121                         <input type="password" class="form-
122 control" id="password" name="password" required>

```

121	</div>
122	<div class="mb-3">
123	<label for="password_conf"
124	class="form-label">Confirm Password</label>
125	<input type="password" class="form-
126	control" id="password_conf" name="password_conf"
127	required>
128	</div>
129	<div class="mb-3">
130	<label for="email" class="form-
131	label">Email</label>
132	<input type="email" class="form-
133	control" id="email" name="email" required>
134	</div>
135	<div class="mb-3">
136	<button type="submit"
137	class="btn">Register</button>
	</div>
	</form>
	<hr />
	</div>
	</main>
	<footer class="footer">
	<div class="container">
	© 2023 Your Company
	</div>
	</footer>
	</body>
	</html>

Table 11. Code vw_register.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 string.'
22             ]
23         ],
24         'penulis' => [
25             'rules' => 'required|string',
26             'errors' => [
27                 'required' => 'Penulis harus diisi.',
28                 'string' => 'Penulis harus berupa
29 string.'
30             ]
31         ],
32         'penerbit' => [
33             'rules' => 'required|string',
34             'errors' => [
35                 'required' => 'Penerbit harus diisi.',
36                 'string' => 'Penerbit harus berupa
37 string.'
38             ]
39         ],
40         'tahun_terbit' => [
41             'rules' =>
42 'required|integer|greater_than[1800]|less_than[2024]',
43             'errors' => [

```


1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;
6	
7	class Buku extends Seeder
8	{
9	public function run()
10	{
11	\$this->db->table('buku')->insert([
12	'judul' => 'Laut Bercerita',
13	'penulis' => 'Leila S. Chudori',
14	'penerbit' => 'Gramedia',
15	'tahun_terbit' => '2017'
16]);
17	\$this->db->table('buku')->insert([
18	'judul' => 'Tentang Kamu',
19	'penulis' => 'Tere Liye',
20	'penerbit' => 'Republika',
21	'tahun_terbit' => '2016'
22]);
23	}
24	}
25	

Table 14. Code Buku.php di Seeds

1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;
6	
7	class UserSeeder extends Seeder
8	{
9	public function run()
10	{
11	\$this->db->table('users')->insert([
12	'username' => 'Farly',
13	'email' => 'farlyhaydy123@gmail.com',
14	'password' => password_hash('123',
15	PASSWORD_DEFAULT),
16]);
17	}

Table 15. Code UserSeeder.php

```

1  -- phpMyAdmin SQL Dump
2  -- version 5.2.1
3  -- https://www.phpmyadmin.net/
4  --
5  -- Host: 127.0.0.1
6  -- Waktu pembuatan: 11 Jun 2024 pada 19.57
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: `perpustakaan`
24 --
25
26 --
27 -- Struktur dari tabel `buku`
28 --
29 CREATE TABLE `buku` (
30   `id` bigint(20) UNSIGNED NOT NULL,
31   `judul` varchar(255) NOT NULL,
32   `penulis` varchar(255) NOT NULL,
33   `penerbit` varchar(255) NOT NULL,
34   `tahun_terbit` year(4) DEFAULT NULL
35 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
36 COLLATE=utf8mb4_general_ci;
37
38 --
39 -- Dumping data untuk tabel `buku`
40 --

```

```

41 INSERT INTO `buku` (`id`, `judul`, `penulis`, `penerbit`,
42 `tahun_terbit`) VALUES
43 (15, 'Laut Bercerita', 'Leila S. Chudori', 'Gramedia', '2017'),
44 (16, 'Tentang Kamu', 'Tere Liye', 'Republika', '2016');
45
46 -----
47 --
48 -- Struktur dari tabel `migrations`
49 --
50
51 CREATE TABLE `migrations` (
52   `id` bigint(20) UNSIGNED NOT NULL,
53   `version` varchar(255) NOT NULL,
54   `class` varchar(255) NOT NULL,
55   `group` varchar(255) NOT NULL,
56   `namespace` varchar(255) NOT NULL,
57   `time` int(11) NOT NULL,
58   `batch` int(11) UNSIGNED NOT NULL
59 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
60 COLLATE=utf8mb4_general_ci;
61 --
62 -- Dumping data untuk tabel `migrations`
63 --
64
65 INSERT INTO `migrations` (`id`, `version`, `class`, `group`,
66 `namespace`, `time`, `batch`) VALUES
67 (1, '2024-06-03-103310', 'App\\Database\\Migrations\\Users',
68 'default', 'App', 1718081782, 1),
69 (2, '2024-06-03-122804', 'App\\Database\\Migrations\\Buku',
70 'default', 'App', 1718081782, 1),
71 (4, '2024-06-11-173406', 'App\\Database\\Migrations\\Users',
72 'default', 'App', 1718127395, 3);
73
74 -----
75 --
76 -- Struktur dari tabel `users`
77 --
78
79 CREATE TABLE `users` (
80   `id` int(10) UNSIGNED NOT NULL,
81   `username` varchar(255) NOT NULL,

```



```

81     `email` varchar(255) NOT NULL,
82     `password` text NOT NULL,
83     `created_at` datetime DEFAULT NULL,
84     `updated_at` datetime DEFAULT NULL
85 )          ENGINE=InnoDB          DEFAULT CHARSET=utf8mb4
86 COLLATE=utf8mb4_general_ci;
87 --
88 -- Dumping data untuk tabel `users`
89 --
90
91 INSERT INTO `users` (`id`, `username`, `email`, `password`,
92 `created_at`, `updated_at`) VALUES
93 (4, 'Farly', 'farlyhaydy123@gmail.com',
94 '$2y$10$xYhsftGqi132/1Y8Qrypge2VGyR2bS0TEBFpXTa4sXUKI4GwnEoh2',
95 NULL, NULL);
96 --
97 -- Indexes for dumped tables
98 --
99
100 --
101 -- Indeks untuk tabel `buku`
102 --
103 ALTER TABLE `buku`
104     ADD PRIMARY KEY (`id`);
105 --
106 -- Indeks untuk tabel `migrations`
107 --
108 ALTER TABLE `migrations`
109     ADD PRIMARY KEY (`id`);
110 --
111 -- Indeks untuk tabel `users`
112 --
113 ALTER TABLE `users`
114     ADD PRIMARY KEY (`id`),
115     ADD UNIQUE KEY `email` (`email`);
116 --
117 --
118 -- AUTO_INCREMENT untuk tabel yang dibuang
119 --
120

```

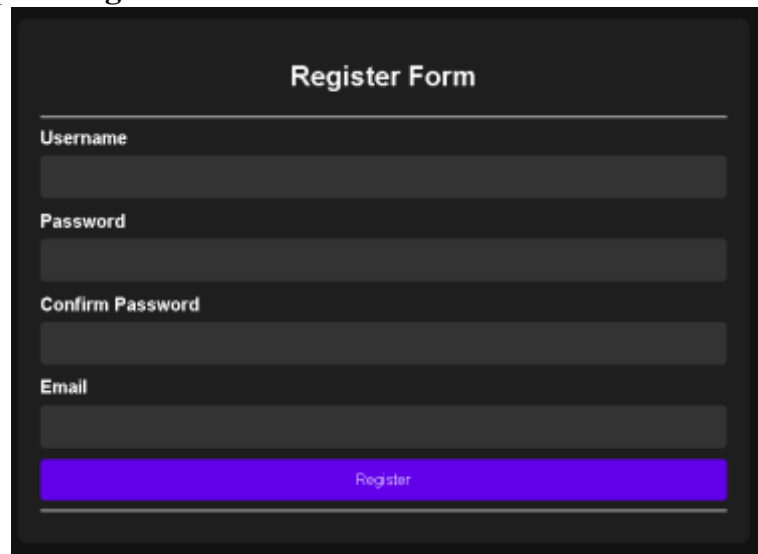
```

121  --
122  -- AUTO_INCREMENT untuk tabel `buku`
123  --
124  ALTER TABLE `buku`
125      MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
126      AUTO_INCREMENT=19;
127  --
128  -- AUTO_INCREMENT untuk tabel `migrations`
129  --
130  ALTER TABLE `migrations`
131      MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
132      AUTO_INCREMENT=5;
133  --
134  -- AUTO_INCREMENT untuk tabel `users`
135  --
136  ALTER TABLE `users`
137      MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
138      AUTO_INCREMENT=5;
139  COMMIT;
140
141  /*!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 */;

```

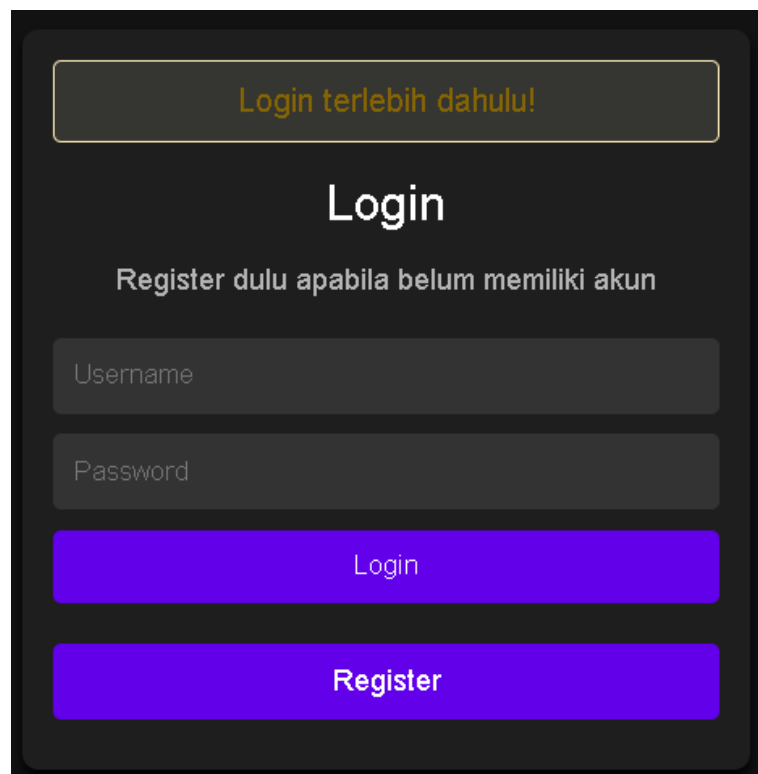
Table 16. Code perpustakaan.sql

B. Output Program



The image shows a 'Register Form' with a dark background. It contains four input fields labeled 'Username', 'Password', 'Confirm Password', and 'Email'. Below these fields is a red button labeled 'Register'.

Gambar Modul 6. 1 Tampilan register



The image shows a 'Login' screen with a dark background. At the top, there is a red box with the text 'Login terlebih dahulu!'. Below this, the word 'Login' is displayed in large white text. Underneath, there is a line of text: 'Register dulu apabila belum memiliki akun'. There are two input fields labeled 'Username' and 'Password'. Below these fields are two red buttons: 'Login' and 'Register'.

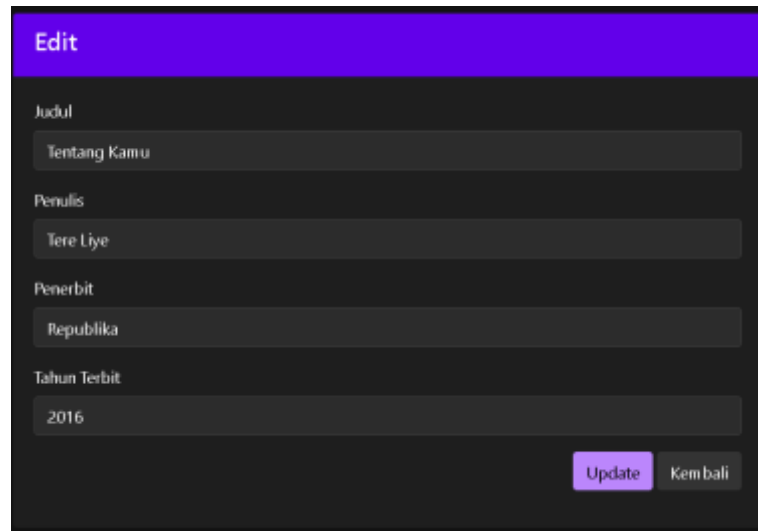
Gambar Modul 6. 2 Tampilan login

List Buku				
				Tambah Data
Judul	Penulis	Penerbit	Tahun Terbit	Action
terang Kamu	Ise Iye	Republika	2016	Edit Hapus
1				

Gambar Modul 6. 3 Tampilan home

Tambah	
Judul	<input type="text" value="Masukkan Judul"/>
Penulis	<input type="text" value="Masukkan Penulis"/>
Penerbit	<input type="text" value="Masukkan Penerbit"/>
Tahun Terbit	<input type="text" value="Masukkan Tahun Terbit"/>
<input type="button" value="Simpan"/> <input type="button" value="Kembali"/>	

Gambar Modul 6. 4 Tampilan tambah buku



Gambar Modul 6. 5 Tampilan edit

C. Pembahasan

- Routes.php

Kode tersebut ditulis menggunakan framework CodeIgniter untuk mengatur routing aplikasi web. Mari kita bahas fungsinya secara umum:

1. Auto Routing (`$routes->setAutoRoute(true);`)

Baris ini secara opsional mengaktifkan auto routing. Artinya, CodeIgniter akan mencoba mengenali controller dan method berdasarkan struktur URL. Namun, pada kode ini terlihat pengaturan manual dilakukan pada baris selanjutnya, jadi fungsinya kemungkinan dinonaktifkan.

2. Route Definisi

- `$routes->get('/', 'Buku::login');` ○ Ini mengatur routing untuk path /. Ketika pengguna mengakses halaman utama (/), aplikasi akan memanggil controller Buku dengan method login.
- `$routes->get('home', 'Buku::index');`
 - Routing untuk path /home. Controller Buku dengan method index akan dijalankan. Biasanya method index digunakan untuk menampilkan halaman utama setelah login.

- **\$routes->get('/register', 'Register::index');**
 - Routing untuk path /register. Controller Register dengan method index akan dijalankan.
- Baris selanjutnya mengatur routing terkait proses registrasi dan login dengan controller Register dan Login.
- **\$routes->get('/logout', 'Login::logout');**
 - Routing untuk path /logout. Controller Login dengan method logout akan dijalankan untuk proses keluar (logout).

3. Routing Buku • **\$routes->get('/buku/create', 'Buku::create');**

- Routing untuk path /buku/create. Controller Buku dengan method create akan dijalankan, kemungkinan untuk menampilkan form pembuatan buku baru.
- **\$routes->post('buku/store', 'Buku::store');**
 - Routing untuk method POST ke path /buku/store. Controller Buku dengan method store kemungkinan digunakan untuk memproses data submit form pembuatan buku baru.
- **\$routes->post('buku/update/', 'Buku::update', ['filter' => 'auth']);**
 - Routing untuk method POST ke path dengan format /buku/update/ diikuti angka (ID buku). Method update pada controller Buku akan dijalankan. Perhatikan bahwa route ini dilindungi dengan filter auth, artinya pengguna harus login terlebih dahulu untuk mengaksesnya.
- **\$routes->get('buku/edit/', 'Buku::edit', ['filter' => 'auth']);**
 - Routing untuk path dengan format /buku/edit/ diikuti angka (ID buku). Method edit pada controller Buku akan dijalankan untuk menampilkan form edit buku. Route ini juga dilindungi dengan filter auth.
- **\$routes->get('buku/delete/', 'Buku::delete', ['filter' => 'auth']);**
 - Routing untuk path dengan format /buku/delete/ diikuti angka (ID buku). Method delete pada controller Buku akan dijalankan untuk menghapus buku. Route ini juga dilindungi dengan filter auth.

- **Buku.php di controller**

Kode tersebut merupakan class controller bernama Buku di aplikasi My Library yang ditulis dengan framework CodeIgniter. Class ini menangani operasi terkait manajemen buku. Mari kita bahas fungsinya secara umum

1. Struktur Class

- Class Buku merupakan child class dari class Controller bawaan CodeIgniter.
- Properti `$model` dideklarasikan untuk menampung objek model yang digunakan untuk interaksi dengan database.
- Konstruktor (`__construct`) menginisialisasi objek model `BukuModel` dan helper yang dibutuhkan (form dan url).

2. Method-method

- **login()**: Method ini menampilkan halaman login. Sepertinya ada kesalahan pemanggilan model `BukuModel` untuk menampilkan daftar post (mungkin seharusnya menggunakan model lain).
- **index()**: Method ini menampilkan halaman daftar buku. Method ini melakukan pengecekan apakah user sudah login (menggunakan session), jika belum akan di redirect ke halaman login. Kemudian method ini mengambil data buku dengan pagination (10 data per halaman) dan mengirimkannya ke view `vw_buku` bersama dengan judul halaman.
- **create()**: Method ini menampilkan halaman create buku baru. Method ini juga melakukan pengecekan apakah user sudah login, jika belum akan di redirect ke halaman login. Terakhir, method ini mengirimkan judul halaman ke view `vw_add`.
- **store()**: Method ini menangani proses penyimpanan data buku baru. Method ini mengambil data buku yang di submit dari form dan melakukan validasi data. Jika validasi gagal, user akan dikembalikan ke halaman `create()`. Jika validasi sukses, data buku akan disimpan ke database menggunakan model `BukuModel`. Setelah berhasil disimpan, method ini menampilkan flashdata sukses dan redirect ke halaman home.
- **edit(\$id)**: Method ini menampilkan halaman edit buku. Method ini mengambil data buku berdasarkan id dari database dan melakukan pengecekan apakah data buku ditemukan. Jika tidak ditemukan, method ini menampilkan flashdata error dan redirect ke halaman sebelumnya. Jika data ditemukan, method ini mengirimkan data buku dan judul halaman ke view `vw_edit`.

- **update(\$id):** Method ini menangani proses update data buku. Method ini mengambil data buku yang di submit dari form dan melakukan validasi data. Jika validasi gagal, user akan dikembalikan ke halaman edit. Jika validasi sukses, data buku akan di update di database menggunakan model BukuModel. Method ini kemudian mengecek apakah update berhasil dan menampilkan flashdata sesuai hasil update (berhasil/gagal) dan redirect ke halaman home.
- **delete(\$id):** Method ini menangani proses delete buku. Method ini melakukan pengecekan apakah user sudah login dan id buku tersedia. Jika tidak valid, method ini akan redirect ke halaman home. Selanjutnya, method ini menghapus data buku berdasarkan id dari database menggunakan model BukuModel. Terakhir, method ini mengecek apakah delete berhasil dan menampilkan flashdata sesuai hasil delete (berhasil/gagal) dan redirect ke halaman home.

- **Home.php**

Kode tersebut menunjukkan controller Home yang memperluas class BaseController dari framework CodeIgniter.

Fungsi Utama:

- Controller ini memiliki satu method publik yaitu index().
- Method index() tidak memiliki parameter dan mengembalikan string.
- Di dalam method index(), terdapat baris `return view('vw_buku');`.
 - Baris ini menunjukkan bahwa controller ini akan menampilkan view dengan nama vw_buku.

- **Login.php**

Kode tersebut merupakan class controller bernama Login di aplikasi MyLibrary yang ditulis dengan framework CodeIgniter. Class ini menangani proses login dan logout user.

1. Struktur Class

- Class Login merupakan child class dari class BaseController bawaan CodeIgniter.

2. Method-method

- **index():** Method ini menampilkan halaman login. User akan diarahkan ke view vw_login.
- **process():** Method ini menangani proses login user.
 - Method ini pertama-tama membuat objek model UsersModel untuk interaksi dengan database.
 - Kemudian, method ini mengambil username dan password yang di submit dari form login.
 - Selanjutnya, method ini mencari data user berdasarkan username dari database.
 - Jika data user ditemukan:
 - ✦ Method ini memverifikasi password yang di submit dengan password yang tersimpan di database menggunakan fungsi password_verify.
 - ✦ Jika password valid, method ini menyimpan username dan flag logged_in ke dalam session.
 - ✦ Terakhir, user akan di redirect ke halaman home.
 - Jika data user tidak ditemukan atau password salah, method ini akan menampilkanflashdata error "Username & Password Salah" dan redirect user kembali ke halaman login.
- **logout():** Method ini menangani proses logout user.
 - Method ini memanggil fungsi session()->destroy() untuk menghapus semua data session yang terkait dengan user.
 - Terakhir, user akan di redirect ke halaman login.

- Register.php

Kode tersebut ditulis menggunakan framework CodeIgniter untuk menangani proses registrasi pengguna. Mari kita bahas fungsinya secara umum:

1. Method Index (Form Registrasi)

- Method index kemungkinan digunakan untuk menampilkan halaman registrasi.
- Method ini mengembalikan view vw_register.

2. Method Process (Registrasi Akun Baru)

- Method process menangani proses submit form registrasi.
- Diawali dengan validasi form input menggunakan library validate.
 - Username:
 - ✦ Diwajibkan diisi (required).
 - ✦ Minimal 4 karakter (min_length[4]).
 - ✦ Maksimal 20 karakter (max_length[20]).
 - ✦ Harus unik (is_unique[users.username]), tidak boleh ada user lain dengan username yang sama.
 - Password:
 - ✦ Diwajibkan diisi (required).
 - ✦ Minimal 0 karakter (min_length[0]).
 - ✦ Maksimal 50 karakter (max_length[50]).
 - Konfirmasi Password:
 - ✦ Harus sama dengan password (matches[password]).
 - Email:
 - ✦ Diwajibkan diisi (required).
 - ✦ Minimal 4 karakter (min_length[4]).
 - ✦ Maksimal 100 karakter (max_length[100]).
 - ✦ Harus berupa email yang valid (valid_email).
- Jika validasi gagal:
 - Flashdata session untuk menampilkan pesan error berisi daftar error validasi dibuat dengan \$this->validator->listErrors().
 - Pengguna dialihkan kembali ke halaman registrasi (redirect()->back()) beserta data form yang sudah diisi sebelumnya (withInput()).
- Jika validasi berhasil:
 - Model UsersModel diinisiasi untuk interaksi dengan database.
 - Password yang diinputkan diubah menjadi hash menggunakan function password_hash sebelum disimpan ke database. Teknik hashing

disarankan untuk keamanan password. ○ Data user baru (username, password hash, dan email) disimpan ke database menggunakan method insert pada model UsersModel.

- Setelah registrasi berhasil, pengguna akan dialihkan ke halaman login (redirect()->to('/login')).

- **AuthFilter.php**

Kode tersebut ditulis menggunakan framework CodeIgniter untuk membuat filter autentikasi pengguna. Mari kita bahas fungsinya secara umum:

1. Class dan Interface

- Class ini bernama AuthFilter dan mengimplementasi interface FilterInterface dari CodeIgniter.

2. Method before

- Method before dieksekusi sebelum request controller dijalankan.
- Di dalam method ini dilakukan pengecekan apakah user sudah login.
- Pengecekan dilakukan dengan memeriksa keberadaan data user_id di session.
- Jika user_id tidak ditemukan di session (artinya user belum login):
 - Filter akan menghentikan proses controller dan mengalihkan pengguna ke halaman login (redirect()->to('/login')).

3. Method after

- Method after dieksekusi setelah request controller dijalankan dan menghasilkan response.
- Namun pada kode ini, method after terlihat kosong. Artinya filter tidak melakukan tindakan apapun setelah proses controller selesai.

- **vw_add.php**

Kode tersebut tampaknya merupakan bagian dari halaman web untuk menambah data buku baru. Mari kita bahas komponennya secara umum:

1. Layout dan Konten

- File ini menggunakan layout layouts/main yang kemungkinan berisi struktur dasar halaman web seperti header, footer, dan navigasi.
- Konten utama ditempatkan di dalam section content.

2. Styling

- Terdapat style CSS di bagian awal untuk mengatur background image pada seluruh halaman.

3. Form Tambah Buku

- Terdapat formulir untuk menambahkan data buku baru.
- Formulir ini menggunakan library form helper dari CodeIgniter.
- Formulir di submit ke URL buku/store menggunakan method form_open.
- Elemen formulir meliputi:
 - Judul buku (text input).
 - Penulis buku (text input).
 - Penerbit buku (text input).
 - Tahun terbit buku (number input).
 - Tombol submit "Simpan" dan tombol kembali "Kembali" ke halaman home.

4. Pesan Error

- Kode ini mengecek keberadaanflashdata session dengan key "error".
- Jika ada, maka akan menampilkan pesan error di dalam alert berwarna bahaya (danger).

5. Validasi Error

- Kode ini menggunakan function validation_list_errors untuk menampilkan pesan error validasi jika ada.

6. Summernote Text Editor

- Bagian akhir kode memuat script untuk menyertakan library Summernote.
- Summernote digunakan sebagai text editor WYSIWYG untuk mengisi konten buku.
- Script ini menginisialisasi Summernote pada element dengan ID post_content dan mengatur beberapa konfigurasi seperti ukuranの高さ (takasa - height) menjadi 500px.

- **vw_buku.php**

Struktur dan Konten HTML

Extend dan Section: Menggunakan `<?= $this->extend('layouts/main') ?>` dan `<?= $this->section('content') ?>` untuk memanfaatkan layout utama dan menetapkan bagian konten pada layout tersebut.

Navbar: Menampilkan navbar dengan logo dan menu navigasi. Jika pengguna sudah login (`session()->get('logged_in')`), maka menampilkan nama pengguna dan opsi logout. Jika belum login, menampilkan opsi login dan register.

CSS Styling

Body dan Komponen: Mengatur latar belakang, warna teks, dan beberapa komponen seperti navbar (`navbar-light`) dan tombol (`btn`, `btn-sm`).

PHP Integration

Flashdata: Menampilkan pesan success atau error menggunakan `session()->getFlashdata('success')` dan `session()->getFlashdata('error')`.

Data Buku: Menampilkan data buku dari variabel `$posts` dalam tabel dengan judul, penulis, penerbit, tahun terbit, dan opsi edit serta delete. Menggunakan `foreach` untuk iterasi data.

JavaScript Section

Pagination Styling: Menggunakan JavaScript untuk menambahkan kelas Bootstrap pada elemen pagination agar tampilan sesuai dengan tema (`.pagination li` dan `.pagination li a`).

Interaksi dengan Pengguna

Edit dan Delete: Setiap baris pada tabel memiliki tombol Edit dan Delete yang mengarahkan ke halaman edit buku atau menghapus data buku. Tombol Delete memiliki konfirmasi sebelum mengirimkan form.

- **vw_edit.php**

Struktur HTML dan PHP

Extend dan Section: Menggunakan `<?= $this->extend('layouts/main') ?>` dan `<?= $this->section('content') ?>` untuk memperluas dan menetapkan bagian konten pada layout utama.

Card Container: Menggunakan Bootstrap untuk merancang tata letak yang responsif. Card ini digunakan untuk mengelompokkan elemen-elemen terkait dengan edit form.

Form Edit: Menggunakan `form_open('buku/update/' . $post['id'])` untuk membuka form yang mengarah ke endpoint update buku dengan method PUT. Input-input seperti judul, penulis, penerbit, dan tahun terbit diisi dengan nilai yang sudah ada (`value="<?= $post['judul'] ?>"`) untuk menampilkan nilai yang sudah ada sebelumnya.

Flashdata dan Validasi: Menggunakan `session()->getFlashdata('error')` untuk menampilkan pesan error jika ada, dan `validation_list_errors()` untuk menampilkan daftar error validasi jika ada.

Styling dengan CSS

Body dan Komponen: Mengatur latar belakang, warna teks, dan tata letak menggunakan CSS untuk memastikan tampilan yang sesuai dengan tema aplikasi perpustakaan.

JavaScript untuk WYSIWYG Editor

Summernote Editor: Menggunakan Summernote untuk menyediakan editor teks yang kaya fitur. Ini termasuk mengatur ukuran tab, toolbar dengan pilihan gaya teks, penambahan gambar, dan fitur-fitur lainnya untuk memudahkan pengguna dalam mengedit konten.

Interaksi dengan Pengguna

Button Actions: Tombol "Update" digunakan untuk mengirimkan form yang telah diisi untuk menyimpan perubahan data buku. Tombol "Kembali" mengarahkan pengguna kembali ke halaman utama setelah melakukan proses edit.

- **vw_login.php**

HTML Structure

DOCTYPE dan Head Section: Mendefinisikan DOCTYPE HTML5 dan bagian `<head>` yang berisi meta tag untuk pengaturan karakter set, viewport, dan judul halaman.

Body: Bagian utama dari halaman yang berisi semua konten yang akan ditampilkan kepada pengguna. Diatur untuk berada di tengah-tengah layar dengan background berupa gambar yang diatur agar rapi dan sesuai dengan tema.

CSS Styling

Body Styling: Menggunakan CSS untuk mengatur tata letak body dengan `display: flex`, `justify-content: center`, dan `align-items: center` untuk memposisikan elemen-elemen secara tengah-tengah di layar.

Form .form-signin: Mengatur form login yang dibungkus dalam sebuah card dengan latar belakang hitam dan border-radius untuk membuat sudut melengkung. Shadow dan warna teks putih untuk menonjolkan konten form.

Alerts .alert: Dua bagian alert yang ditampilkan secara kondisional:

#initial-alert: Alert yang selalu ditampilkan saat halaman dimuat.

Alert untuk kesalahan login (session()->getFlashdata('error')), yang ditampilkan jika terdapat kesalahan saat login.

Form Inputs and Buttons: Mengatur input teks (username dan password) dengan background berwarna gelap dan border yang halus. Tombol login dengan warna latar belakang ungu tua (#6200ea) dan transisi warna saat hover. Tombol register dengan warna latar belakang hijau muda (#03dac6) yang berubah warna saat dihover.

Interaksi JavaScript

Register Button: Digunakan onclick untuk mengarahkan pengguna ke halaman register dengan window.location.href.

- vw_register.php

HTML Structure

DOCTYPE dan Head Section: Mendefinisikan DOCTYPE HTML5 dan bagian <head> yang berisi meta tag untuk pengaturan karakter set, viewport, dan judul halaman.

Body: Bagian utama dari halaman yang berisi semua konten yang akan ditampilkan kepada pengguna. Diatur untuk berada di tengah-tengah layar dengan background yang rapi dan sesuai dengan tema.

CSS Styling

Body Styling: Menggunakan CSS untuk mengatur tata letak body dengan display: flex, flex-direction: column, dan height: 100% untuk mengisi tinggi layar penuh. Background menggunakan gambar yang diatur agar rapi dan sesuai dengan tema aplikasi.

Container .container: Mengatur container untuk form registrasi dengan maksimum lebar 600px, padding, background berwarna gelap (#1e1e1e), dan border-radius untuk membuat sudut melengkung. Shadow dan warna teks putih untuk menonjolkan konten form.

Form Styling: Mengatur input teks (username, password, confirm password, email) dengan background berwarna gelap (#333), border yang halus, dan padding yang sesuai. Tombol "Register" dengan latar belakang ungu tua (#6200ea) dan transisi warna saat hover untuk efek visual yang lebih interaktif.

Alert Styling .alert: Bagian alert yang ditampilkan jika ada kesalahan dalam pengisian formulir. Memiliki padding, border-radius, dan warna latar belakang yang sesuai dengan tema aplikasi.

Footer .footer: Bagian footer dengan teks yang berada di bagian bawah halaman, menggunakan background dan warna teks yang konsisten dengan tema aplikasi.

Interaksi PHP

Flashdata Display: PHP digunakan untuk menampilkan pesan error jika terdapat kesalahan dalam pengisian form. Menggunakan session()->getFlashdata('error') untuk menampilkan pesan yang relevan di dalam alert.

- **BukuModel.php**

Kode tersebut mendefinisikan model BukuModel yang mewakili tabel buku di database. Model ini mempermudah interaksi dengan data buku di aplikasi. Mari kita bahas komponennya secara umum:

1. Class dan Namespace

- Kode diawali dengan deklarasi namespace App\Models yang menunjukkan lokasi class model.
- Deklarasi class BukuModel extends Model mendefinisikan class BukuModel yang mewarisi class Model dari CodeIgniter.

2. Atribut Model

- protected \$table = 'buku';: Menentukan nama tabel di database yang diwakili oleh model ini, yaitu buku.
- protected \$allowedFields = ['judul', 'penulis', 'penerbit', 'tahun_terbit'];: Menentukan daftar nama kolom yang diizinkan untuk diakses dan dimanipulasi oleh model.
- protected \$createdField = 'created_at';: Menentukan nama kolom yang menyimpan waktu pembuatan data.

- `protected $updatedAt = 'updated_at';` Menentukan nama kolom yang menyimpan waktu pembaruan data.
- `protected $deletedField = 'deleted_at';` Menentukan nama kolom yang menyimpan waktu penghapusan data (jika soft delete diaktifkan).
- `protected $validationRules = [...];` Menentukan aturan validasi untuk masing-masing kolom.
 - Setiap kolom memiliki array aturan yang terdiri dari rules dan errors.
 - rules berisi daftar aturan validasi yang dipisahkan dengan |.
 - errors berisi pesan error untuk setiap aturan validasi.
- `protected $skipValidation = false;` Menentukan apakah validasi data otomatis harus dilewati.

3. Aturan Validasi

Model ini mendefinisikan aturan validasi untuk kolom judul, penulis, penerbit, dan tahun_terbit:

- **Judul:**
 - Harus diisi (required).
 - Harus berupa string (string).
- **Penulis:**
 - Harus diisi (required).
 - Harus berupa string (string).
- **Penerbit:**
 - Harus diisi (required).
 - Harus berupa string (string).
- **Tahun Terbit:**
 - Harus diisi (required).
 - Harus berupa bilangan bulat (integer).
 - Harus lebih besar dari 1800 (`greater_than[1800]`).
 - Harus lebih kecil dari 2024 (`less_than[2024]`).

- UsersModel.php

Kode tersebut mendefinisikan model UsersModel yang mewakili tabel users di database. Model ini mempermudah interaksi dengan data pengguna di aplikasi. Mari kita bahas komponennya secara umum:

1. Class dan Namespace

- Kode diawali dengan deklarasi namespace `App\Models` yang menunjukkan lokasi class model.
- Deklarasi class `UsersModel extends Model` mendefinisikan class `UsersModel` yang mewarisi class `Model` dari `CodeIgniter`.

2. Atribut Model

- `protected $table = "users";`: Menentukan nama tabel di database yang diwakili oleh model ini, yaitu `users`.
- `protected $returnType = "object";`: Menentukan tipe data yang dikembalikan oleh metode model. Dalam hal ini, model akan mengembalikan objek.
- `protected $useTimestamps = true;`: Mengaktifkan penandaan waktu `created_at` dan `updated_at` secara otomatis untuk setiap data yang ditambahkan atau diperbarui.

protected \$allowedFields = ['username', 'password', 'email']; Menentukan daftar nama kolom yang diizinkan untuk diakses dan dimanipulasi oleh model.

- **Buku.php di Seeds**

Kode tersebut mendefinisikan class seeder bernama Buku yang digunakan untuk mengisi data awal ke dalam tabel buku di database. Class seeder ini mewarisi class Seeder dari CodeIgniter.

1. Struktur Class

- Deklarasi namespace App\Database\Seeds; menunjukkan lokasi class seeder.
- Deklarasi class Buku extends Seeder mendefinisikan class Buku yang mewarisi class Seeder.
- Method run() berisi kode yang akan dijalankan untuk mengisi data ke tabel.

2. Memasukkan Data ke Tabel Buku

- Method run() menggunakan method insert() dari class DB untuk memasukkan data ke tabel buku.
- Dua baris insert() digunakan untuk memasukkan dua data buku yang berbeda.
- Setiap baris insert() berisi array asosiatif yang memetakan nama kolom dengan nilai data yang ingin dimasukkan.
 - Kolom judul, penulis, penerbit, dan tahun_terbit diisi dengan data yang sesuai.

- **UserSeeder.php**

Kode tersebut mendefinisikan class seeder bernama UserSeeder yang digunakan untuk mengisi data awal ke dalam tabel users di database. Class seeder ini mewarisi class Seeder dari CodeIgniter.

1. Struktur Class

- Deklarasi namespace App\Database\Seeds; menunjukkan lokasi class seeder.
- Deklarasi class UserSeeder extends Seeder mendefinisikan class UserSeeder yang mewarisi class Seeder.

Method run() berisi kode yang akan dijalankan untuk mengisi data ke tabel.

2. Memasukkan Data ke Tabel Users

- Method run() menggunakan method insert() dari class DB untuk memasukkan data ke tabel users.
- Satu baris insert() digunakan untuk memasukkan satu data pengguna.
- Baris insert() berisi array asosiatif yang memetakan nama kolom dengan nilai data yang ingin dimasukkan:

- Kolom username diisi dengan nilai "Farly".
- Kolom email diisi dengan nilai "farlyhaydy123@gmail.com".
- Kolom password diisi dengan nilai hash dari string "123" menggunakan fungsi password_hash(). Algoritma hash yang digunakan adalah PASSWORD_DEFAULT.

- perpustakaan.sql

File tersebut berisi kode untuk membuat struktur dan mengisi data awal ke dalam sebuah database perpustakaan. Mari kita bahas bagian-bagiannya:

1. Informasi Header

--- phpMyAdmin SQL Dump - Menunjukkan ini adalah file dump database yang dibuat oleh phpMyAdmin. --- Host: 127.0.0.1 - Alamat server database (localhost). --- Waktu pembuatan --- Tanggal dan waktu pembuatan file dump. --- Versi server - Versi server database (MariaDB). --- Versi PHP - Versi interpreter PHP yang digunakan.

2. Perintah SQL

-SET SQL_MODE - Mengatur mode operasi untuk server database. -START TRANSACTION - Memulai transaksi database (perubahan data bersifat sementara sampai commit). -SET time_zone - Mengatur zona waktu database.

3. Dump Struktur Database perpustakaan

-CREATE TABLE - Perintah untuk membuat tabel-tabel di database perpustakaan. - Tabel buku menyimpan informasi buku (id, judul, penulis, penerbit, tahun_terbit). - Tabel migrations menyimpan history migrasi database (versi, class, group, namespace, time, batch). - Tabel users menyimpan informasi pengguna (id, username, email, password, created_at, updated_at).

4. Dump Data Tabel

-INSERT INTO - Perintah untuk memasukkan data awal ke dalam tabel. - Tabel buku berisi data awal 2 buah buku. - Tabel migrations berisi history migrasi yang pernah dilakukan (3 record). - Tabel users berisi data awal 1 pengguna dengan username "Farly". Password disimpan dalam bentuk hash.

5. Indexes dan Auto Increment

-ALTER TABLE - Perintah untuk mengatur indeks dan auto increment pada tabel. - Setiap tabel memiliki primary key untuk identifikasi data. - Tabel users memiliki unique key pada kolom email untuk memastikan email unik. - Perintah AUTO_INCREMENT mengatur nilai awal dan penambahan otomatis untuk kolom id pada setiap tabel.

6. Commit Transaksi

-COMMIT - Menyimpan perubahan data yang dilakukan pada transaksi sebelumnya.

LINK GITHUB

https://github.com/Vainn11/Prak_Web_2/tree/b4af3f7197d0a76126fb49215269dcde20945c63/Modul%206