

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Fatih Alif Ghifari

NIM. 2210817310014

**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 : Fatih Alif Ghifari
NIM : 2210817310014

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	48
C. Pembahasan	50
LINK GITHUB	67

DAFTAR GAMBAR

Gambar Modul 6. 1 Tampilan register.....	48
Gambar Modul 6. 2 Tampilan login.....	48
Gambar Modul 6. 3 Tampilan home	49
Gambar Modul 6. 4 Tampilan tambah buku	49
Gambar Modul 6. 5 Tampilan edit.....	50

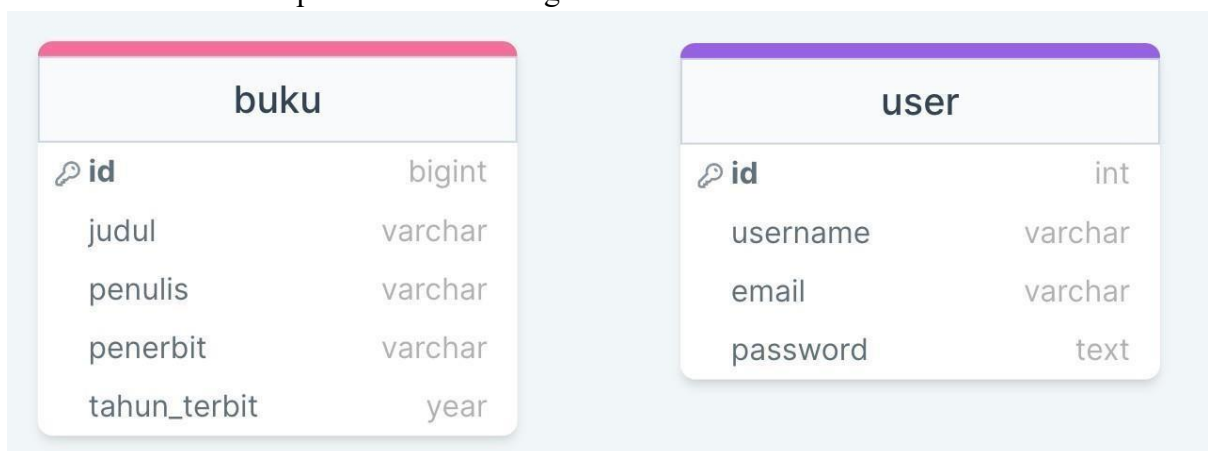
DAFTAR TABEL

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.....	20
Table 8. Code vw_buku.php	26
Table 9. Code vw_edit.php	29
Table 10. Code vw_login.php	33
Table 11. Code vw_register.php	38
Table 12. Code BukuModel.php	40
Table 13. Code UserModel.php	40
Table 14. Code Buku.php di Seeds	41
Table 15. Code UserSeeder.php.....	42
Table 16. Code perpustakaan.sql	47

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 occurred, 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	\$delete = \$this->model->delete(\$id);
138	
139	if (\$delete) {
140	session()->setFlashdata('success', 'Post has
141	been removed successfully.');
142	return redirect()->to(base_url('home'));
143	} else {
144	session()->setFlashdata('error', 'Some
145	problems occurred, please try again.');
146	return redirect()->to(base_url('home'));
147	}
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 if (password_verify($password, $dataUser-
25 >password)) {
26                 session()->set([
27                     'username' => $dataUser->username,
28                     'logged_in' => true
29                 ]);
30                 return redirect()->to(base_url('home'));
31             } else {
32                 session()->setFlashdata('error', 'Username &
33 Password Salah');
34                 return redirect()->to(base_url('login'));
35             }
36         } else {
37             session()->setFlashdata('error', 'Username &
38 Password Salah');
39             return redirect()->to(base_url('login'));
40         }
41
42     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'), $this->request->getPost('password_conf'));
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     $users = new UsersModel();
58     // Hash the password before storing it
59     $hashedPassword = password_hash($this->request-
60 >getVar('password'), PASSWORD_DEFAULT);
61
62     $users->insert([
63         'username' => $this->request->getVar('username'),
64         'password' => $hashedPassword,
65         'email' => $this->request->getVar('email')
66     ]);
67     return redirect()->to('/login');
68 }
}

```

Table 5. Code Register.php

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	
10	class AuthFilter implements FilterInterface
11	{
12	public function before(RequestInterface \$request,
13	\$arguments = null)
14	{
15	// Check if the user is logged in
16	if (!session()->has('user_id')) {
17	// If not logged in, redirect to the login
18	page
19	return redirect()->to('/login');
20	}
21	}
22	public function after(RequestInterface \$request,
23	ResponseInterface \$response, \$arguments = null)
24	{
25	// Do nothing after the request
	}
	}

Table 6. Code AuthFilter.php

```

1 <?= $this->extend('layouts/main') ?>
2
3 <?= $this->section('content') ?>
4 <style>
5     body {
6
7                                     background-image:
8 url('https://images.pexels.com/photos/694740/pexels-photo-
9 694740.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1');
10         background-repeat: no-repeat;
11         background-attachment: fixed;
12         background-position: center;
13         background-size: cover;
14     }
15 </style>
16
17 <div class="container mt-5">
18     <div class="row justify-content-center">
19         <div class="col-md-8">
20             <div class="card shadow-sm">
21                 <div class="card-header text-white"
22 style="background-color: #c2724f;">
23                     <h4>Tambah</h4>
24                 </div>
25                 <div class="card-body">
26                     <?php if (session()->getFlashdata('error'))
27 : ?>
28                         <div class="alert alert-danger">
29                             <?php echo session()-
30 >getFlashdata('error'); ?>
31                         </div>
32                     <?php endif; ?>
33
34                     <?= validation_list_errors() ?>
35
36                     <?= form_open('buku/store'); ?>
37                     <div class="form-group">
38                         <label for="judul">Judul</label>
39                         <input type="text" class="form-control"
40 id="judul" name="judul" placeholder="Masukkan Judul">
41                     </div>
42                     <div class="form-group">
43                         <label for="penulis">Penulis</label>
44                         <input type="text" class="form-control"
45 id="penulis" name="penulis" placeholder="Masukkan Penulis">

```

41	</div>
42	<div class="form-group">
43	<label for="penerbit">Penerbit</label>
44	<input type="text" class="form-control"
45	id="penerbit" name="penerbit" placeholder="Masukkan Penerbit">
46	</div>
47	<div class="form-group">
48	<label for="tahun_terbit">Tahun
49	Terbit</label>
50	<input type="number" class="form-
51	control" id="tahun_terbit" name="tahun_terbit"
52	placeholder="Masukkan Tahun Terbit">
53	</div>
54	<div class="form-group text-right">
55	<button class="btn btn-success"
56	style="background-color: #c2724f">Simpan</button>
57	<a href="<?= base_url('home') ?>"
58	class="btn btn-secondary" style="background-color:
59	#733e27">Kembali
60	</div>
61	<?= form_close(); ?>
62	</div>
63	</div>
64	</div>
65	<?= \$this->endSection() ?>
66	<?= \$this->section('extra-js') ?>
67	<!-- include summernote css/js -->
68	<link
69	href="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist/summ
70	ernote-bs4.min.css" rel="stylesheet">
71	<script
72	src="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist/summe
	rnote-bs4.min.js"></script>
	<script>
	\$(document).ready(function() {
	\$('#post_content').summernote({
	tabsize: 2,
	height: 500
	});
	});

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

Table 7. Code vw_add.php

1	<pre><?= \$this->extend('layouts/main') ?></pre>
2	
3	<pre><?= \$this->section('content') ?></pre>
4	<pre><style></pre>
5	<pre> body {</pre>
6	<pre> background-image:</pre>
7	<pre>url('https://images.pexels.com/photos/877971/pexels-</pre>
8	<pre>photo-</pre>
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	

```

38 877971.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1
39 ');
40     background-repeat: no-repeat;
41     background-attachment: fixed;
42     background-position: center;
43     background-size: cover;
44 }
45 .btn-tambah-data {
46     background-color: #c2724f;
47     color: white;
48     transition: background-color 0.3s ease;
49 }
50
51 .btn-tambah-data:hover {
52     background-color: #a5593e;
53     color: white;
54 }
55
56 .card {
57     background-color: rgba(255, 255, 255, 0.9);
58 }
59
60 .card-header {
61     background-color: rgba(115, 62, 39, 0.9);
62 }
63
64 .table {
65     background-color: rgba(255, 255, 255, 0.8);
66 }
67
68 .table thead {
69     background-color: rgba(194, 114, 79, 0.9);
70 }
71
72 </style>
73
74 <!-- Navbar -->
75 <nav class="navbar navbar-expand-lg navbar-light bg-light
76 shadow-sm">
77     <div class="container">
78         <a class="navbar-brand" href="<?= base_url('/')
79 ?>">My Library</a>
80         <button class="navbar-toggler" type="button" data-
81 toggle="collapse" data-target="#navbarNav" aria-

```

```

78 controls="navbarNav"      aria-expanded="false"      aria-
79 label="Toggle navigation">
80     <span class="navbar-toggler-icon"></span>
81     </button>
82     <div class="collapse navbar-collapse"
83 id="navbarNav">
84         <ul class="navbar-nav ml-auto">
85             <?php if (session()->get('logged_in')) :
86             ?>
87                 <li class="nav-item">
88                     <a class="nav-link"
89 href="#">Welcome, <?= session()->get('username'); ?></a>
90                 </li>
91                 <li class="nav-item">
92                     <a class="nav-link" href="<?=
93 base_url('login/logout'); ?>">Logout</a>
94                 </li>
95                 <?php else : ?>
96                     <li class="nav-item">
97                         <a class="nav-link" href="<?=
98 base_url('login'); ?>">Login</a>
99                     </li>
100                     <li class="nav-item">
101                         <a class="nav-link" href="<?=
102 base_url('register'); ?>">Register</a>
103                     </li>
104                 <?php endif; ?>
105             </ul>
106         </div>
107     </div>
108 </nav>
109
110 <div class="container mt-5">
111     <div class="row justify-content-center">
112         <div class="col-md-10">
113             <div class="card shadow-sm">
114                 <div class="card-header text-white">
115                     <h5 class="mb-0">List Buku</h5>
116                     <a href="<?= base_url('buku/create');
117 ?>" class="btn btn-sm float-right btn-tambah-data">Tambah
118 Data</a>
119                 </div>
120                 <div class="card-body">

```

```

10                                     <?php if (session()-
9 >getFlashdata('success')) : ?>
11                                     <div class="alert alert-success">
0                                     <?= session()-
11 >getFlashdata('success'); ?>
1                                     </div>
11                                     <?php endif; ?>
2
11                                     <?php if (session()-
3 >getFlashdata('error')) : ?>
11                                     <div class="alert alert-danger">
11                                     <?= session()-
4 >getFlashdata('error'); ?>
11                                     </div>
5                                     <?php endif; ?>
11
6                                     <table class="table table-bordered
11 table-striped table-hover">
7                                     <thead class="text-center">
11                                     <tr>
8                                     <th scope="col">Judul</th>
11                                     <th scope="col">Penulis</th>
9                                     <th scope="col">Penerbit</th>
12                                     <th scope="col">Tahun
0 Terbit</th>
12                                     <th scope="col">Action</th>
1                                     </tr>
12                                     </thead>
2                                     <tbody>
12                                     <?php if (!empty($posts) &&
3 is_array($posts)) : ?>
12                                     <?php foreach ($posts as
4 $row) : ?>
12                                     <tr>
5                                     <td><?=
12 $row['judul']; ?></td>
6                                     <td><?=
12 $row['penulis']; ?></td>
7                                     <td><?=
12 $row['penerbit']; ?></td>
8                                     <td><?=
12 $row['tahun_terbit']; ?></td>

```

12	<td class="text-
9	center">
13	<form
0	onsubmit="return confirm('Apakah Anda Yakin ?');"
13	
1	
13	
2	
13	
3	
13	
4	
13	
5	
13	
6	
13	
7	
13	
8	
13	
9	
14	
0	
14	
1	
14	
2	
14	
3	
14	
4	
14	
5	


```

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</a>
                                <button
type="submit"          class="btn          btn-danger          btn-
sm">Delete</button>
                                </form>
                                </td>
                                </tr>
                                <?php endforeach; ?>
                                <?php else : ?>
                                <tr>
                                <td colspan="5"
class="text-center">No post found.</td>
                                </tr>
                                <?php endif; ?>
                                </tbody>
                                </table>

                                <div class="d-flex justify-content-
center">
                                <?= $pager->links(); ?>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                <?= $this->endSection() ?>

                                <?= $this->section('extra-js') ?>
                                <script>
                                $(document).ready(function() {
                                $('.pagination li').addClass('page-
item').css('background-color', '#000000');
                                $('.pagination li a').addClass('page-
link').css('background-color', '#000000');

```

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

Table 8. Code `vw_buku.php`

1	<pre><?= \$this->extend('layouts/main') ?></pre>
2	
3	<pre><?= \$this->section('content') ?></pre>
4	<pre><style></pre>
5	<pre> body {</pre>
6	<pre> background-image:</pre>
7	<pre>url('https://images.unsplash.com/photo-1688320006653-</pre>
8	<pre>72c493042675?q=80&w=1887&auto=format&fit=crop&ixlib=rb-</pre>
9	
1	
0	

```

4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVuZDB8fHx8fA
%3D%3D');

        background-repeat: no-repeat;
        background-attachment: fixed;
        background-position: center;
        background-size: cover;
    }
</style>

<div class="container mt-5">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card shadow-sm">
                <div class="card-header text-white"
style="background-color: #c2724f;">
                    <h4>Edit</h4>
                </div>
                <div class="card-body">

                    <?php if (session()-
>getFlashdata('error')) : ?>
                        <div class="alert alert-danger">
                            <?= session()-
>getFlashdata('error'); ?>
                        </div>
                    <?php endif; ?>

                    <?= validation_list_errors() ?>

                    <?= form_open('buku/update/' .
$post['id']); ?>
                    <input type="hidden" name="_method"
value="PUT">

                    <div class="form-group">
                        <label for="judul">Judul</label>
                        <input type="text" class="form-
control" id="judul" name="judul" placeholder="Masukkan
Judul" value="<?= $post['judul'] ?>">
                    </div>
                    <div class="form-group">
                        <label for="penulis">Penulis</label>

```

```

<input type="text" class="form-
control" id="penulis" name="penulis" placeholder="Masukkan
Penulis" value="<?= $post['penulis'] ?>">
</div>
<div class="form-group">
<label
for="penerbit">Penerbit</label>
<input type="text" class="form-
control" id="penerbit" name="penerbit"
placeholder="Masukkan Penerbit" value="<?=
$post['penerbit'] ?>">
</div>
<div class="form-group">
<label for="tahun_terbit">Tahun
Terbit</label>
<input type="number" class="form-
control" id="tahun_terbit" name="tahun_terbit"
placeholder="Masukkan Tahun Terbit" value="<?=
$post['tahun_terbit'] ?>">
</div>
<div class="form-group text-right">
<button class="btn btn-success"
style="background-color: #c2724f">Update</button>
<a href="<?= base_url('home') ?>"
class="btn btn-secondary" style="background-color:
#733e27">Kembali</a>
</div>
<?= form_close(); ?>
</div>
</div>
</div>
</div>
</div>
<?= $this->endSection() ?>

<?= $this->section('extra-js') ?>
<!-- include summernote css/js -->
<link
href="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist
/summernote-bs4.min.css" rel="stylesheet">

```

	<pre> <script src="https://cdn.jsdelivr.net/npm/summernote@0.8.15/dist/ summernote-bs4.min.js"></script> <script> \$(document).ready(function() { \$('#post_content').summernote({ tabsize: 2, height: 500 }); }); </script> <?= \$this->endSection() ?> </pre>
--	--

Table 9. Code vw_edit.php

1	<!DOCTYPE html>
2	<html lang="id">
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-image:
19	url('https://images.pexels.com/photos/2041540/pexels-
20	photo-
21	
22	
23	
24	
25	
26	
27	
28	

```
29 2041540.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=
30 1');
31         background-repeat: no-repeat;
32         background-attachment: fixed;
33         background-position: center;
34         background-size: cover;
35     }
36     .form-signin {
37         width: 100%;
38         max-width: 330px;
39         padding: 15px;
40         margin: auto;
41         background: rgba(255, 255, 255, 0.8);
42         border-radius: 8px;
43         box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
44         position: relative;
45     }
46     .form-signin img {
47         margin-bottom: 20px;
48     }
49
50     .form-signin h1 {
51         margin-bottom: 10px;
52         font-size: 24px;
53         font-weight: normal;
54     }
55
56     .form-signin p {
57         margin-bottom: 20px;
58         font-size: 14px;
59         color: #6c757d;
60     }
61
62     .form-signin input[type="text"],
63     .form-signin input[type="password"] {
64         width: 100%;
65         padding: 10px;
66         margin-bottom: 10px;
67         border: 1px solid #ccc;
68         border-radius: 4px;
69         box-sizing: border-box;
70         background: rgba(255, 255, 255, 0.8);
```

69	}
70	
71	.form-signin button {
72	width: 100%;
73	padding: 10px;
74	background-color: #c2724f;
75	color: white;
76	border: none;
77	border-radius: 4px;
78	cursor: pointer;
79	margin-bottom: 10px;
80	}
81	.form-signin button:hover {
82	background-color: #733e27;
83	}
84	.form-signin .alert {
85	padding: 10px;
86	margin-bottom: 10px;
87	border-radius: 4px;
88	color: #c2724f;
89	background-color: rgba(255, 253, 220, 0.8);
90	border: 1px solid #ffeeba;
91	display: none;
92	}
93	
94	.form-signin .text-muted {
95	color: #6c757d;
96	margin-top: 20px;
97	}
98	
99	.btn-register {
10	width: 100%;
10	padding: 10px;
0	background-color: #c2724f;
10	color: white;
1	border: none;
10	border-radius: 4px;
2	cursor: pointer;
10	text-align: center;
3	}
10	
	.btn-register:hover {

```

4         background-color: #733e27;
10     }
5     </style>
10 </head>
6
10 <body>
7     <main class="form-signin">
10         <!-- Alert untuk informasi saat halaman dimuat -->
8     >
10         <div class="alert" id="initial-alert"
9 style="display: block;">
11             Login terlebih dahulu!
0         </div>
11
1     <!-- Alert untuk kesalahan login -->
11         <div class="alert" style="display: <?php echo
2 (session()->getFlashdata('error')) ? 'block' : 'none';
11 ?>;">
3             <?= session()->getFlashdata('error'); ?>
11         </div>
4
11         <form method="post" action="<?= base_url();
5 ?>/login/process">
6             <?= csrf_field(); ?>
11             <h1 class="h3 mb-3 fw-normal">Login</h1>
6             <p>Register dulu apabila belum memiliki
11 akun</p>
7
11             <input type="text" name="username"
8 id="username" placeholder="Username" required autofocus>
11             <input type="password" name="password"
9 id="password" placeholder="Password" required>
12
12             <button type="submit">Login</button>
0         </form>
12
1     <!-- Tombol Register -->
12         <button class="btn-register"
2 onclick="window.location.href='<?php
12 base_url('register'); ?>'>Register</button>
3         </main>
12 </body>

```


4	
12	</html>
5	
12	
6	
12	
7	
12	
8	
12	
9	
13	
0	
13	
1	
13	
2	
13	
3	
13	
4	
13	
5	
13	
6	
13	
7	
13	
8	
13	
9	

Table 10. Code vw_login.php

```
1 <!doctype html>
2 <html lang="id" 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-image:
21 url('https://images.pexels.com/photos/2908984/pexels-
22 photo-
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
```

```
41 2908984.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=
42 1');
43         background-repeat: no-repeat;
44         background-attachment: fixed;
45         background-position: center;
46         background-size: cover;
47         background-color: #f5f5f5;
48         padding: 20px;
49     }
50     .container {
51         max-width: 600px;
52         margin: auto;
53         padding: 20px;
54         background: rgba(255, 255, 255, 0.8);
55         border-radius: 8px;
56         box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
57         margin-top: 40px;
58     }
59     h1 {
60         font-size: 24px;
61         margin-bottom: 20px;
62         text-align: center;
63     }
64
65     .form-label {
66         display: block;
67         margin-bottom: 5px;
68         font-weight: bold;
69     }
70     .form-control {
71         width: 100%;
72         padding: 10px;
73         margin-bottom: 10px;
74         border: 1px solid #ccc;
75         border-radius: 4px;
76         box-sizing: border-box;
77         background: rgba(255, 255, 255, 0.8);
78     }
79     .btn {
80         width: 100%;
```

```

81         padding: 10px;
82         background-color: #c2724f;
83         color: white;
84         border: none;
85         border-radius: 4px;
86         cursor: pointer;
87         margin-bottom: 10px;
88     }
89     .btn:hover {
90         background-color: #733e27;
91     }
92
93     .alert {
94         padding: 10px;
95         margin-bottom: 10px;
96         border-radius: 4px;
97         color: #856404;
98         background-color: #fff3cd;
99         border: 1px solid #ffeeba;
100    }
101
102    .footer {
103        margin-top: auto;
104        padding: 10px 0;
105        text-align: center;
106        background-color: #e9ecef;
107    }
108
109    .footer .container {
110        max-width: none;
111    }
112    </style>
113    </head>
114
115    <body>
116
117        <main class="flex-shrink-0">
118            <div class="container">
119                <h1>Register Form</h1>
120                <hr />
121                <?php    if    (!empty(session()-
122    >getFlashdata('error')))) : ?>
123                    <div class="alert">

```

```

0          <h4>Periksa Entrian Form</h4>
11          <?= session()->getFlashdata('error');
1      ?>
11          </div>
2          <?php endif; ?>
11          <form method="post" action="<?= base_url();
3      ?>/register/process">
11          <?= csrf_field(); ?>
4          <div class="mb-3">
11              <label for="username" class="form-
label">Username</label>
5              <input type="text" class="form-control"
11          id="username" name="username" required>
6          </div>
11          <div class="mb-3">
7              <label for="password" class="form-
label">Password</label>
11              <input type="password" class="form-
8          control" id="password" name="password" required>
11          </div>
9          <div class="mb-3">
12              <label for="password_conf" class="form-
0          label">Confirm Password</label>
12              <input type="password" class="form-
1          control" id="password_conf" name="password_conf"
12          required>
2          </div>
12          <div class="mb-3">
3              <label for="email" class="form-
label">Email</label>
12              <input type="email" class="form-
4          control" id="email" name="email" required>
12          </div>
5          <div class="mb-3">
12              <button type="submit"
6          class="btn">Register</button>
12          </div>
7          </form>
12          <button class="btn"
8          onclick="window.location.href='<?=
base_url();
12          ?>/login'">Kembali ke Login</button>
9          <hr />
13      </div>
          </main>

```

0	</body>
13	
1	</html>
13	
2	
13	
3	
13	
4	
13	
5	
13	
6	
13	
7	

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 string.'
29            ]
30        ],
31        'penerbit' => [
32            'rules' => 'required|string',
33            'errors' => [
34                'required' => 'Penerbit harus diisi.',
35                'string' => 'Penerbit harus berupa
36        string.'
37            ]
38        ],
39        'tahun_terbit' => [
40            'rules' =>
41            'required|integer|greater_than[1800]|less_than[2024]',
42            'errors' => [
43                'required' => 'Tahun terbit harus
44            diisi.',
```

41	<code>'integer' => 'Tahun terbit harus berupa</code>
42	<code>angka.',</code>
43	<code>'greater_than' => 'Tahun terbit harus lebih</code>
44	<code>besar dari 1800.',</code>
45	<code>'less_than' => 'Tahun terbit harus lebih</code>
46	<code>kecil dari 2024.'</code>
47	<code>]</code>
48	<code>],</code>
49	<code>]);</code>
50	<code>protected \$skipValidation = false;</code>
	<code>}</code>

Table 12. Code BukuModel.php

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

Table 13. Code UserModel.php

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 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8mb4 */;
18
19 --
20 -- Database: `perpustakaan`
21 --
22 --
23 -- -----
24 --
25 --
26 --
27 -- Struktur dari tabel `buku`
28 --
29
30 CREATE TABLE `buku` (
31   `id` bigint(20) UNSIGNED NOT NULL,
32   `judul` varchar(255) NOT NULL,
33   `penulis` varchar(255) NOT NULL,
34   `penerbit` varchar(255) NOT NULL,
35   `tahun_terbit` year(4) DEFAULT NULL
36 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
37 COLLATE=utf8mb4_general_ci;
38
39 --
40 -- Dumping data untuk tabel `buku`
41 --

```

```

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

```

```

81 --
82 -- Struktur dari tabel `users`
83 --
84
85 CREATE TABLE `users` (
86   `id` int(10) UNSIGNED NOT NULL,
87   `username` varchar(255) NOT NULL,
88   `email` varchar(255) NOT NULL,
89   `password` text NOT NULL,
90   `created_at` datetime DEFAULT NULL,
91   `updated_at` datetime DEFAULT NULL
92 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
93 COLLATE=utf8mb4_general_ci;
94
95 --
96 -- Dumping data untuk tabel `users`
97 --
98 INSERT INTO `users` (`id`, `username`, `email`,
99 `password`, `created_at`, `updated_at`) VALUES
100 (4, 'Farly', 'farlyhaydy123@gmail.com',
101 '$2y$10$xYhsftGqi132/1Y8Qrypge2VGyR2bS0TEBFpXTa4sXUKI4Gw
102 nEoh2', NULL, NULL);
103
104 --
105 -- Indexes for dumped tables
106 --
107
108 --
109 -- Indeks untuk tabel `buku`
110 --
111 ALTER TABLE `buku`
112   ADD PRIMARY KEY (`id`);
113
114 --
115 -- Indeks untuk tabel `migrations`
116 --
117 ALTER TABLE `migrations`
118   ADD PRIMARY KEY (`id`);
119
120 --
121 -- Indeks untuk tabel `users`
122 --
123 ALTER TABLE `users`

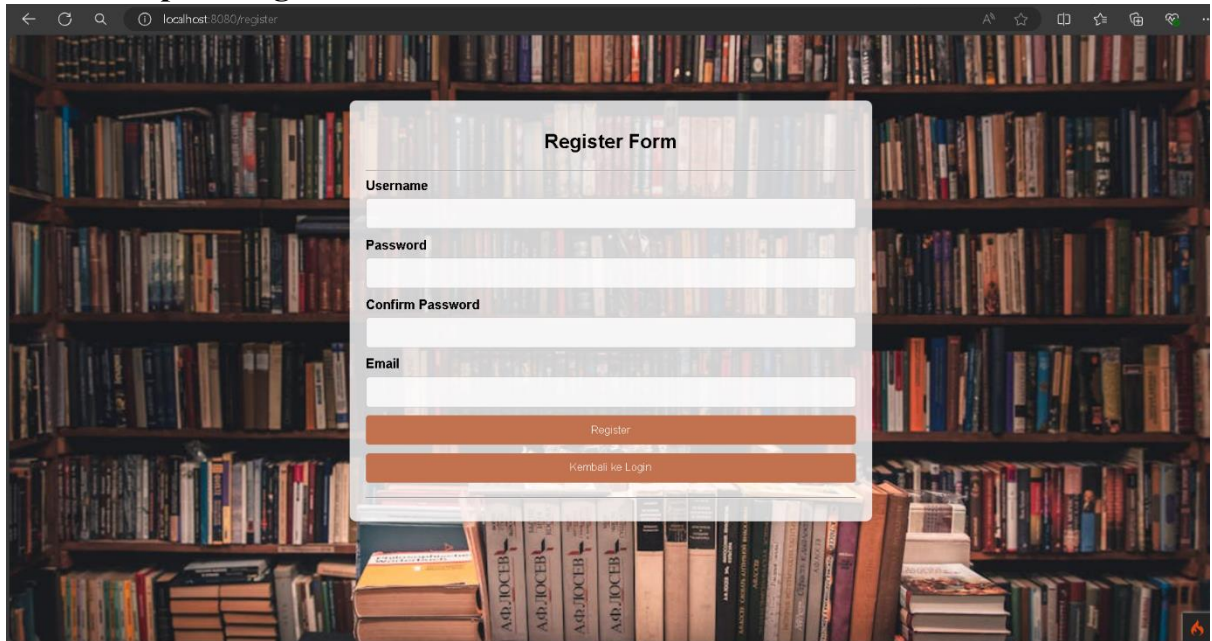
```

0	ADD PRIMARY KEY (`id`),	
11	ADD UNIQUE KEY `email` (`email`);	
1		
11	--	
2	-- AUTO_INCREMENT untuk tabel yang dibuang	
11	--	
3		
11	--	
4	-- AUTO_INCREMENT untuk tabel `buku`	
11	--	
11	ALTER TABLE `buku`	
5	MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,	
11	AUTO_INCREMENT=19;	
6		
11	--	
7	-- AUTO_INCREMENT untuk tabel `migrations`	
11	--	
11	ALTER TABLE `migrations`	
8	MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,	
11	AUTO_INCREMENT=5;	
9		
12	--	
0	-- AUTO_INCREMENT untuk tabel `users`	
12	--	
1	ALTER TABLE `users`	
12	MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,	
2	AUTO_INCREMENT=5;	
12	COMMIT;	
3		
12	/*!40101	SET
4	CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;	
12	/*!40101	SET
5	CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;	
12	/*!40101	SET
6	COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;	
12		
7		
12		
8		
12		
9		
13		

0	
13	
1	
13	
2	
13	
3	
13	
4	
13	
5	
13	
6	
13	
7	
13	
8	
13	
9	
14	
0	
14	
1	

Table 16. Code perpustakaan.sql

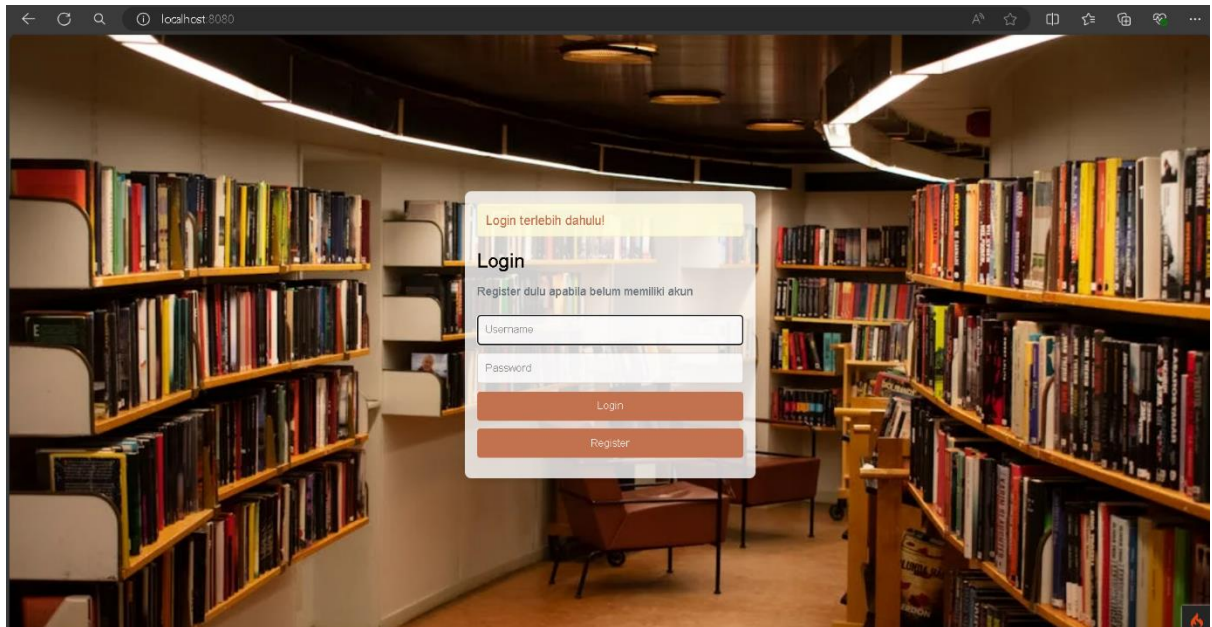
B. Output Program



The screenshot displays a web browser window with the address bar showing 'localhost:8080/register'. The background is a high-resolution image of a library with rows of bookshelves filled with books. Overlaid on this background is a semi-transparent white 'Register Form'. The form contains the following elements:

- Register Form** (Title)
- Username** (Label) with a text input field.
- Password** (Label) with a text input field.
- Confirm Password** (Label) with a text input field.
- Email** (Label) with a text input field.
- Register** (Orange button).
- Kembali ke Login** (Orange button).

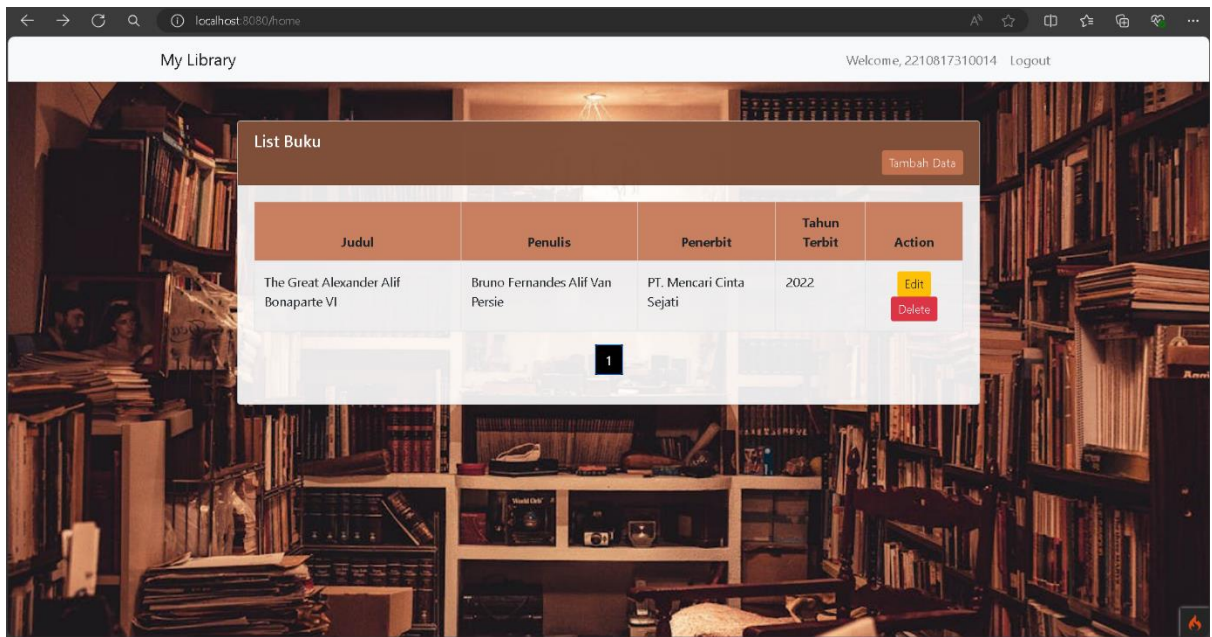
Gambar Modul 6. 1 Tampilan register



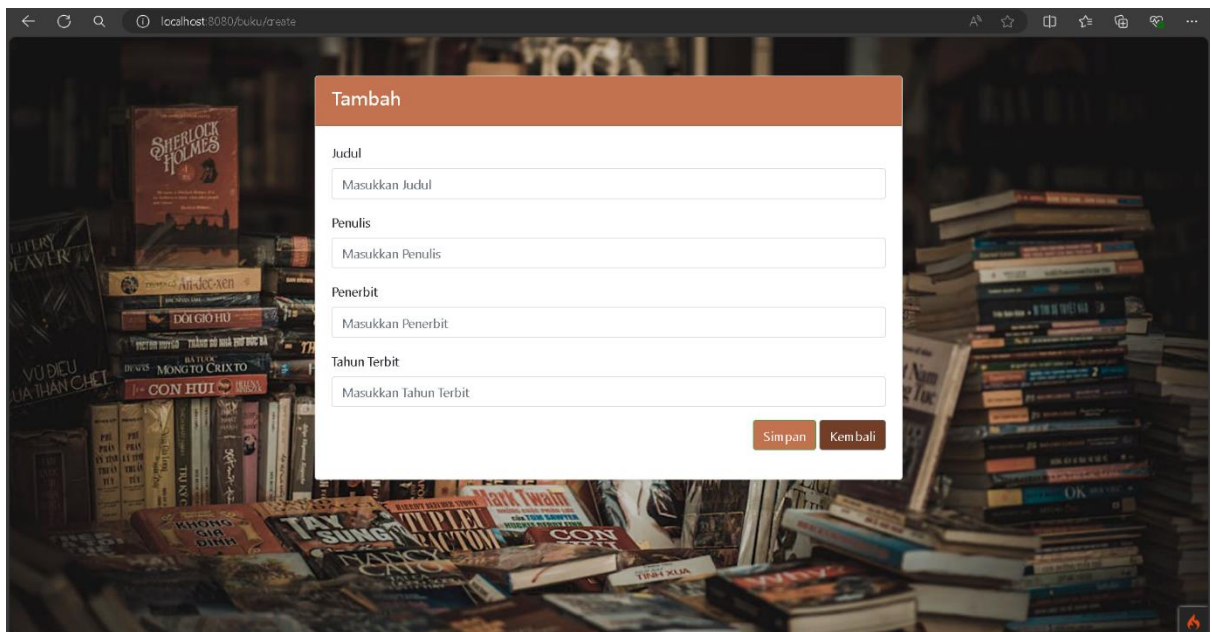
The screenshot shows a web browser window with the address bar at 'localhost:8080'. The background is a photograph of a library interior, featuring bookshelves and a reading area with armchairs. A semi-transparent white 'Login' overlay is centered on the screen. The overlay contains the following elements:

- Login terlebih dahulu** (Yellow banner message).
- Login** (Section title).
- Register dulu apabila belum memiliki akun** (Text prompt).
- Username** (Label) with a text input field.
- Password** (Label) with a text input field.
- Login** (Orange button).
- Register** (Orange button).

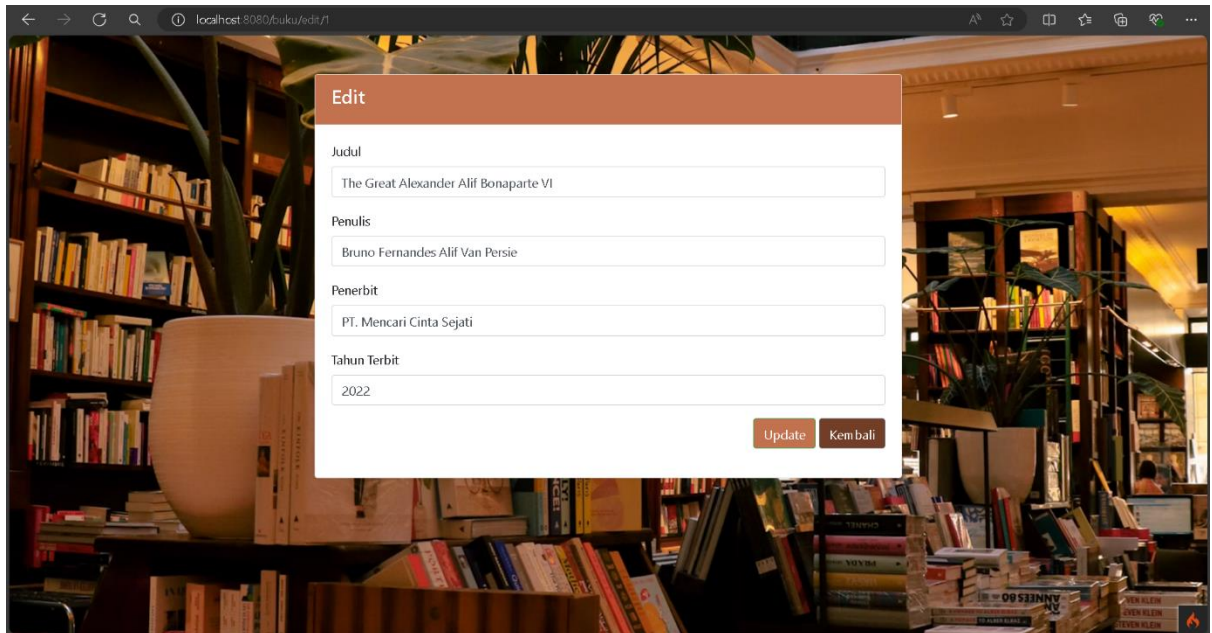
Gambar Modul 6. 2 Tampilan login



Gambar Modul 6. 3 Tampilan home



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 menampilkanflashdata 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 menampilkanflashdata 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 menampilkanflashdata 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 menampilkanflashdata 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の高さい (takasa - height) menjadi 500px.

- **vw_buku.php**

Kode tersebut menampilkan daftar buku beserta formulir 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. Navbar

- Terdapat navbar di bagian atas yang kemungkinan dibuat dengan library Bootstrap.
- Navbar berisi link ke halaman home, dan link kondisional untuk Login/Register atau Logout tergantung apakah user sudah login atau belum (diperiksa menggunakan `session logged_in`).

3. Styling

- Kode CSS diawal untuk mengatur background image pada seluruh halaman.
- Kode CSS tambahan di akhir `section extra-js` untuk memberi style pada elemen paginasi (menambahkan class dan warna background).

4. List Buku

- Bagian utama konten menampilkan tabel berisi daftar buku.
- Data buku kemungkinan berasal dari variable `$posts`.
- Tabel dilengkapi dengan header untuk Judul, Penulis, Penerbit, Tahun Terbit, dan Action.
- Kolom Action berisi link untuk Edit dan tombol untuk Delete data buku (menggunakan form dengan method DELETE).
 - Tombol Delete dilengkapi dengan konfirmasi sebelum menghapus data.

5. Paginasi

- Di bawah tabel terdapat elemen untuk navigasi paginasi.
- Kode menggunakan variable \$pager untuk menampilkan link halaman.

6. Tambah Data Buku

- Terdapat link "Tambah Data" di bagian atas tabel untuk menuju halaman form tambah buku baru.

- vw_edit.php

Kode tersebut menampilkan formulir untuk mengedit data buku. 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. Form Edit Buku

- Terdapat formulir untuk mengedit data buku.
- Formulir ini menggunakan library form helper dari CodeIgniter.
- Formulir di submit ke URL buku/update/ diikuti dengan ID buku yang akan diedit (menggunakan method PUT).
 - Perhatikan penggunaan form_open dengan menyertakan method PUT.
- Formulir diisi dengan data buku yang sudah ada (variable \$post).
- Elemen formulir meliputi:
 - Judul buku (text input).
 - Penulis buku (text input).
 - Penerbit buku (text input).
 - Tahun terbit buku (number input).
 - Tombol submit "Update" dan tombol kembali "Kembali" ke halaman home.

3. Pesan Error

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

4. Validasi Error

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

5. Summernote Text Editor (Tidak Digunakan)

- Bagian akhir kode memuat script untuk menyertakan library Summernote.
- Komentar pada script menunjukkan kemungkinan Summernote digunakan sebagai text editor WYSIWYG.

- **vw_login.php**

Kode tersebut menampilkan halaman login untuk aplikasi MyLibrary. Mari kita bahas komponen-komponennya:

1. Struktur HTML

- File tersebut merupakan file HTML standar yang berisi struktur halaman web.
- Bagian `<head>` berisi informasi meta dan stylesheet untuk mengatur tampilan halaman.
- Bagian `<body>` berisi konten utama halaman web.

2. Form Login

- Bagian `<main>` berisi form login dengan class `form-signin`.
 - Form ini menggunakan method POST untuk mengirimkan data login ke URL `<?= base_url(); ?>/login/process`.
 - Form dilengkapi dengan token keamanan `<?= csrf_field(); ?>`.
 - Form memiliki field untuk username dan password.
 - Tombol Login digunakan untuk submit form.

3. Alert Awal

- Div dengan class alert dan id initial-alert berisi pesan "Login terlebih dahulu!" yang ditampilkan secara default.

4. Alert Kesalahan Login

- Div dengan class alert kedua menampilkan pesan error dari session flashdata (jika ada).
 - Script PHP digunakan untuk mengecek keberadaan flashdata error dan menampilkan alert secara kondisional.

5. Link Registrasi

- Span dengan class btn-register menampilkan text "Register" yang berfungsi sebagai link ke halaman registrasi.
 - Script Javascript digunakan untuk mengatur link ke halaman registrasi menggunakan `base_url('register')`.

6. Penjelasan kode alert

- Kode ini menampilkan dua alert secara terpisah, satu untuk informasi awal dan satu untuk menampilkan pesan kesalahan login dari session flashdata.
- Kode memanfaatkan session flashdata untuk menampilkan pesan error login secara dinamis.

- vw_register.php

Kode tersebut menampilkan formulir untuk registrasi akun baru di aplikasi MyLibrary. Mari kita bahas komponennya secara umum:

1. Struktur HTML

- Kode diawali dengan deklarasi DOCTYPE dan tag `<html>` standar.
- Bagian `<head>` berisi meta informasi dan link ke stylesheet CSS untuk mengatur tampilan halaman registrasi.
- Bagian `<body>` berisi konten utama yaitu formulir registrasi dibungkus dengan tag `<main>`.

Terdapat footer di bagian bawah dengan informasi tambahan.

2. Form Registrasi

- Formulir registrasi menggunakan class container dan berada di dalam tag `<main>`.
- Formulir menggunakan method POST dan action ke URL `<?= base_url(); ?>/register/process` untuk memproses data registrasi.
- Formulir dilengkapi dengan token CSRF untuk keamanan.
- Elemen formulir meliputi:
 - Judul "Register Form".
 - Username input field.
 - Password input field.
 - Confirm Password input field.
 - Email input field.
 - Tombol "Register" untuk submit formulir.
- Kode ini mengecek keberadaanflashdata session dengan key "error".
 - Jika ada, maka akan menampilkan pesan error di dalam alert berwarna kuning dengan judul "Periksa Entrian Form".

3. Styling CSS

- Kode CSS mengatur tampilan keseluruhan halaman registrasi, meliputi:
 - Gaya container formulir.
 - Gaya elemen formulir seperti label, input text, button, dan alert.
 - Penggunaan background image untuk seluruh halaman.
 - Gaya untuk bagian footer.

- 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 $updatedField = '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 masingmasing 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/lifghi/Praktikum-Pemrograman-Web2/tree/main/Modul6/PRAK601>