

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Farlyhaydy H.Djalil

NIM. 2210817210006

**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 : Farlyhaydy H.Djalil
NIM : 2210817320006

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

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

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI.....	3
DAFTAR GAMBAR	4
DAFTAR TABEL.....	5
SOAL 1	6
A. Source Code	7
B. Output Program	41
C. Pembahasan	43
D. Tautan Git.....	61

DAFTAR GAMBAR

Gambar 1. Contoh database soal 1	6
Gambar 2. Screenshot Hasil Jawaban Soal tampilan register	41
Gambar 3. Screenshot Hasil Jawaban Soal 1 tampilan login.....	41
Gambar 4. Screenshot Hasil Jawaban Soal 1 tampilan home	42
Gambar 5. Screenshot Hasil Jawaban Soal 1 tampilan tambah buku	42
Gambar 6. Screenshot Hasil Jawaban Soal 1 tampilan edi atau mengupdatet buku...	43

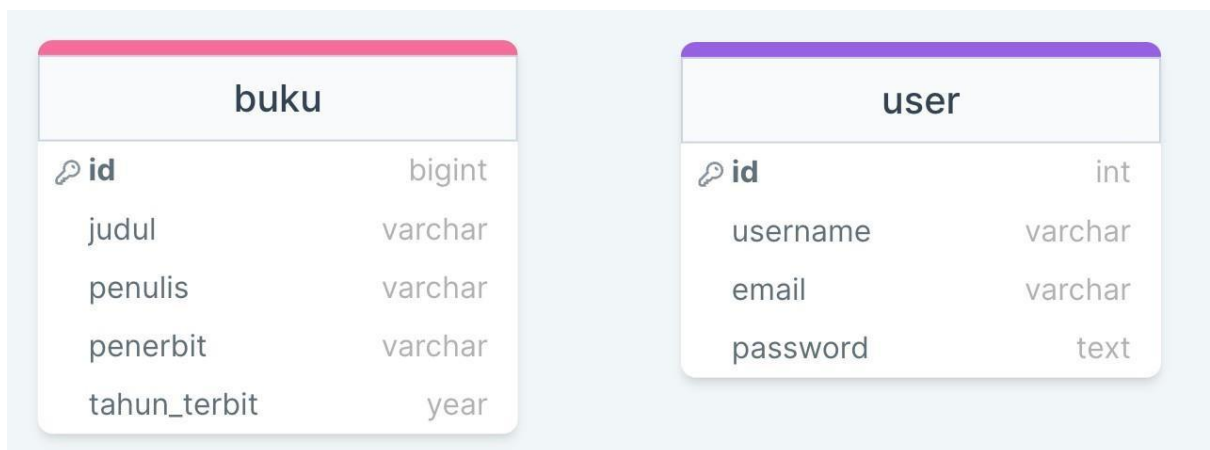
DAFTAR TABEL

Tabel 1. Source Code Soal 1 Routes.php.....	7
Tabel 2. Source Code Soal 1 Buku.php di Controller.....	11
Tabel 3. Source Code Soal 1 Home.php	12
Tabel 4. Source Code Soal 1 Login.php	13
Tabel 5. Source Code Soal 1 Register.php.....	15
Tabel 6. Source Code Soal 1 AuthFilter.php	16
Tabel 7. Source Code Soal 1 vw_add.php	19
Tabel 8. Source Code Soal 1 vw_buku.php	23
Tabel 9. Source Code Soal 1 vw_edit.php	26
Tabel 10. Source Code Soal 1 vw_login.php.....	29
Tabel 11. Source Code Soal 1 vw_register.php	33
Tabel 12. Source Code Soal 1 BukuModel.php.....	35
Tabel 13. Source Code Soal 1 UsersModel.php	35
Tabel 14. Source Code Soal 1 Buku.php di Seeds.....	36
Tabel 15. Source Code Soal 1 UserSeeder.php.....	36
Tabel 16. Source Code Soal 1 perpustakaan.sql	40

SOAL 1

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

Routes.php

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

Tabel 1. Source Code Soal 1 Routes.php

Buku.php

1	<?php namespace App\Controllers;
2	
3	use App\Models\BukuModel;
4	use CodeIgniter\Controller;
5	use App\Models\PostModel;
6	
7	class Buku extends <u>Controller</u>
8	{
9	/**
10	*

```

11      * @var Model
12      */
13      protected $model;
14
15      public function __construct()
16      {
17          $this->model = new BukuModel();
18          $this->helpers = ['form', 'url'];
19
20      }
21
22      public function login()
23      {
24          $data = [
25              'posts' => $this->model-
26              >paginate(10),
27              'pager' => $this->model->pager,
28              'title' => 'POST LIST'
29          ];
30
31          return view('vw_login', $data);
32      }
33
34      public function index()
35      {
36          $data = [
37              'posts' => $this->model-
38              >paginate(10),
39              'pager' => $this->model->pager,
40              'title' => 'POST LIST'
41          ];
42
43          return view('vw_buku', $data);
44      }
45
46      public function create()
47      {
48          // Check if user is authenticated
49          if (!session()->get('logged_in')) {
50              return redirect()-
51              >to(base_url('/login'));
52          }
53
54          $data = ['title' => 'Create new post'];

```



```

51
52         return view('vw_add', $data);
53     }
54
55     public function store()
56     {
57         $data = $this->request-
58 >getPost(['judul', 'penulis', 'penerbit',
59 'tahun_terbit']);
60
61         if (! $this->validateData($data, $this-
62 >model->validationRules)) {
63             return $this->create();
64         }
65
66         $post = $this->validator-
67 >getValidated();
68
69         $save = $this->model->save([
70             'judul' => $post['judul'],
71             'penulis' => $post['penulis'],
72             'penerbit' => $post['penerbit'],
73             'tahun_terbit' =>
74 $post['tahun_terbit'],
75         ]);
76
77         session()->setFlashdata('success', 'Post
78 has been added successfully.');
```

```

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

```

94         return redirect()->back();
95     }

```

```

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

```

121	} else {
122	session()->setFlashdata('error', 'Some problems occurred, please try again.');
123	return redirect()->back();
124	}
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()-
136	>to(base_url('home'));
137	}
138	\$delete = \$this->model->delete(\$id);
139	
140	if (\$delete) {
141	session()->setFlashdata('success', 'Post has been removed successfully.');
142	return redirect()-
143	>to(base_url('home'));
144	} else {
145	session()->setFlashdata('error', 'Some problems occurred, please try again.');
146	return redirect()-
147	>to(base_url('home'));
148	}
149	}
150	}

Tabel 2. Source Code Soal 1 Buku.php di Controller

Home.php

1	<?php
2	
3	namespace App\Controllers;
4	

5	class Home extends <u>BaseController</u>
6	{
7	public function index(): string
8	{
9	return view('vw_buku');
10	}
11	}

Tabel 3. Source Code Soal 1 Home.php

Login.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Models\UsersModel;
6	
7	class Login extends <u>BaseController</u>
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-
18	>getVar('username');
19	\$password = \$this->request-
20	>getVar('password');
21	\$dataUser = \$users->where('username',
22	\$username)->first();
23	
24	if (\$dataUser) {
25	// Check the password using object
26	property syntax
27	if (password_verify(\$password,
28	\$dataUser->password)) {
29	session()->set([
30	'username' => \$dataUser-
31	>username,

26	'logged_in' => true
27]);
28	return redirect()-
29	>to(base_url('home')));
30	} else {
31	session()->setFlashdata('error',
32	'Username & Password Salah');
33	return redirect()->back();
34	}
35	} else {
36	session()->setFlashdata('error',
37	'Username & Password Salah');
38	return redirect()->back();
39	}
40	}
41	public function logout()
42	{
43	session()->destroy();
44	return redirect()->to('/login');
	}
	}

Tabel 4. Source Code Soal 1 Login.php

Register.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Models\UsersModel;
6	
7	class Register extends <u>BaseController</u>
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[use
rs.username]',
21                 'errors' => [
22                     'required' => '{field} Harus
23                     diisi',
24                     'min_length' => '{field} Minimal
25                     4 Karakter',
26                     'max_length' => '{field}
27                     Maksimal 20 Karakter',
28                     'is_unique' => 'Username sudah
29                     digunakan sebelumnya'
30                 ]
31             ],
32             'password' => [
33                 'rules' =>
34                 'required|min_length[0]|max_length[50]',
35                 'errors' => [
36                     'required' => '{field} Harus
37                     diisi',
38                     'min_length' => '{field} Minimal
39                     0 Karakter',
40                     'max_length' => '{field}
41                     Maksimal 50 Karakter',
42                 ]
43             ],
44             'password_conf' => [
45                 'rules' => 'matches[password]',
46                 'errors' => [
47                     'matches' => 'Konfirmasi
48                     Password tidak sesuai dengan password',
49                 ]
50             ],
51             'email' => [
52                 'rules' =>
53                 'required|min_length[4]|max_length[100]|valid_email'
54             ],
55             'errors' => [
56                 'required' => '{field} Harus
57                 diisi',

```

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

Tabel 5. Source Code Soal 1 Register.php

AuthFilter.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 <u>FilterInterface</u>
11	{
12	public function before(RequestInterface
	\$request, \$arguments = 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 page
17	return redirect()->to('/login');
18	}
19	}
20	
21	public function after(RequestInterface
	\$request, ResponseInterface \$response,
	\$arguments = null)
22	{
23	// Do nothing after the request
24	}
25	}

Tabel 6. Source Code Soal 1 AuthFilter.php

vw_add.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	<style>
5	body {
6	background-image:
	url('https://images.unsplash.com/photo-
	1688320006653-
	72c493042675?q=80&w=1887&auto=format&fit=crop&ixlib=
	rb-
	4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8f
7	Hx8fA%3D%3D');
8	background-repeat: no-repeat;
9	background-attachment: fixed;
10	background-position: center;
11	background-size: cover;


```

12     }
13     </style>
14
15 <div class="container mt-5">
16     <div class="row justify-content-center">
17         <div class="col-md-8">
18             <div class="card shadow-sm">
19                 <div class="card-header text-white"
20 style="background-color: #f0768b;">
19                     <h4>Tambah</h4>
20                 </div>
21                 <div class="card-body">
22                     <?php if (session()-
23 >getFlashdata('error')) : ?>
24                         <div class="alert alert-
25 danger">
26                             <?php echo session()-
27 >getFlashdata('error'); ?>
28                         </div>
29                     <?php endif; ?>
30
31                     <?= validation_list_errors() ?>
32
33                     <?= form_open('buku/store'); ?>
34                     <div class="form-group">
35                         <label
36 for="judul">Judul</label>
37                         <input type="text"
38 class="form-control" id="judul" name="judul"
39 placeholder="Masukkan Judul">
40                     </div>
41                     <div class="form-group">
42                         <label
43 for="penulis">Penulis</label>
44                         <input type="text"
45 class="form-control" id="penulis" name="penulis"
46 placeholder="Masukkan Penulis">
47                     </div>
48                     <div class="form-group">
49                         <label
50 for="penerbit">Penerbit</label>

```

```

41         <input type="text"
class="form-control" id="penerbit" name="penerbit"
placeholder="Masukkan Penerbit">
42     </div>
43     <div class="form-group">
44         <label
for="tahun_terbit">Tahun Terbit</label>
45         <input type="number"
class="form-control" id="tahun_terbit"
name="tahun_terbit" placeholder="Masukkan Tahun
Terbit">
46     </div>
47     <div class="form-group text-
right">
48         <button class="btn btn-
success" style="background-color:
#dc3545">Simpan</button>
49         <a href="<?=
base_url('home') ?>" class="btn btn-secondary"
style="background-color: #000000">Kembali</a>
50     </div>
51     <?= form_close(); ?>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57
58 <?= $this->endSection() ?>
59
60 <?= $this->section('extra-js') ?>
61 <!-- include summernote css/js -->
62 <link
href="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/
/dist/summernote-bs4.min.css" rel="stylesheet">
63 <script
src="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/
/dist/summernote-bs4.min.js"></script>
64 <script>
65     $(document).ready(function() {
66         $('#post_content').summernote({
67             tabsize: 2,
68             height: 500

```

69	});
70	});
71	</script>
72	<?= \$this->endSection() ?>

Tabel 7. Source Code Soal 1 vw_add.php

vw_buku.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	<style>
5	body {
6	background-image:
	url('https://images.unsplash.com/photo-
	1661097410573-
	16d926ec1ec7?q=80&w=2070&auto=format&fit=crop&ixlib
	=rb-
	4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8
	fHx8fA%3D%3D');
7	background-repeat: no-repeat;
8	background-attachment: fixed;
9	background-position: center;
10	background-size: cover;
11	}
12	</style>
13	
14	<!-- Navbar -->
15	<nav class="navbar navbar-expand-lg navbar-light
	bg-light shadow-sm">
16	<div class="container">
17	<a class="navbar-brand" href="<?=
	base_url('/') ?>">My Library
18	<button class="navbar-toggler"
	type="button" data-toggle="collapse" data-
	target="#navbarNav" aria-controls="navbarNav" aria-
	expanded="false" aria-label="Toggle navigation">
19	<span class="navbar-toggler-
	icon">
20	</button>
21	<div class="collapse navbar-collapse"
	id="navbarNav">

```

22         <ul class="navbar-nav ml-auto">
23             <?php if (session()-
>get('logged_in')) : ?>
24                 <li class="nav-item">
25                     <a class="nav-link"
href="#">Welcome, <?= session()->get('username');
?></a>
26                 </li>
27                 <li class="nav-item">
28                     <a class="nav-link"
href="<?= base_url('login/logout'); ?>">Logout</a>
29                 </li>
30             <?php else : ?>
31                 <li class="nav-item">
32                     <a class="nav-link"
href="<?= base_url('login'); ?>">Login</a>
33                 </li>
34                 <li class="nav-item">
35                     <a class="nav-link"
href="<?= base_url('register'); ?>">Register</a>
36                 </li>
37             <?php endif; ?>
38         </ul>
39     </div>
40 </div>
41 </nav>
42
43 <div class="container mt-5">
44     <div class="row justify-content-center">
45         <div class="col-md-10">
46             <div class="card shadow-sm">
47                 <div class="card-header text-white"
style="background-color: #f0768b;">
48                     <h5 class="mb-0">List Buku</h5>
49                     <a href="<?=
base_url('buku/create'); ?>" class="btn btn-sm
float-right" style="background-color: #dc3545;
color: white;">Tambah Data</a>
50                 </div>
51                 <div class="card-body">
52                     <?php if (session()-
>getFlashdata('success')) : ?>

```

```

53         <div class="alert alert-
success">
54             <?= session() -
>getFlashdata('success'); ?>
55         </div>
56         <?php endif; ?>
57
58         <?php if (session() -
>getFlashdata('error')) : ?>
59             <div class="alert alert-
danger">
60                 <?= session() -
>getFlashdata('error'); ?>
61             </div>
62             <?php endif; ?>
63
64             <table class="table table-
bordered table-striped table-hover">
65                 <thead class="text-center"
style="background-color: #6f00ff;">
66                     <tr>
67                         <th
scope="col">Judul</th>
68                         <th
scope="col">Penulis</th>
69                         <th
scope="col">Penerbit</th>
70                         <th
scope="col">Tahun Terbit</th>
71                         <th
scope="col">Action</th>
72                     </tr>
73                 </thead>
74                 <tbody>
75                     <?php if
(!empty($posts) && is_array($posts)) : ?>
76                         <?php foreach
($posts as $row) : ?>
77                             <tr>
78                                 <td><?=
$row['judul']; ?></td>
79                                 <td><?=
$row['penulis']; ?></td>

```

```

80 $row['penerbit']; ?></td>                                <td><?=
81                                     <td><?=
82 $row['tahun_terbit']; ?></td>                                <td
83 class="text-center">                                       <form
84                                     onsubmit="return confirm('Apakah Anda Yakin ?');"
85                                     action="<?= base_url('buku/delete/' . $row['id']);
86                                     ?>" method="POST">
87                                     <in
88 put type="hidden" name="{csrf_token}"
89 value="{csrf_hash}">
90                                     <in
91 put type="hidden" name="_method" value="DELETE">
92                                     <a
93 href="<?= base_url('buku/edit/' . $row['id']); ?>"
94 class="btn btn-warning btn-sm">Edit</a>
95                                     <bu
96 tton type="submit" class="btn btn-danger btn-
97 sm">Delete</button>
98                                     </form>
99                                     </td>
100                                    </tr>
101                                    <?php endforeach;
102 ?>
103                                    <?php else : ?>
104                                    <tr>
105                                        <td colspan="5"
106 class="text-center">No post found.</td>
107                                    </tr>
108                                    <?php endif; ?>
109                                </tbody>
110                            </table>
111
112                            <div class="d-flex justify-
113 content-center">
114                                <?= $pager->links(); ?>
115                            </div>
116                        </div>
117                    </div>
118                </div>
119            </div>
120        </div>

```

```

107 </div>
108
109 <?= $this->endSection() ?>
110
111 <?= $this->section('extra-js') ?>
112 <script>
113     $(document).ready(function() {
114         $('.pagination li').addClass('page-
item').css('background-color', '#000000');
115         $('.pagination li a').addClass('page-
link').css('background-color', '#000000');
116     });
117 </script>
118 <?= $this->endSection() ?>

```

Tabel 8. Source Code Soal 1 vw_buku.php

vw_edit.php

```

1 <?= $this->extend('layouts/main') ?>
2
3 <?= $this->section('content') ?>
4 <style>
5     body {
6         background-image:
url('https://images.unsplash.com/photo-
1688320006653-
72c493042675?q=80&w=1887&auto=format&fit=crop&ixlib=
rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8f
Hx8fA%3D%3D');
7         background-repeat: no-repeat;
8         background-attachment: fixed;
9         background-position: center;
10        background-size: cover;
11    }
12 </style>
13
14 <div class="container mt-5">
15     <div class="row justify-content-center">
16         <div class="col-md-8">
17             <div class="card shadow-sm">
18                 <div class="card-header text-white"
style="background-color: #f0768b;">
19                     <h4>Edit</h4>

```

```

20         </div>
21         <div class="card-body">
22
23             <?php if (session()-
24 >getFlashdata('error')) : ?>
25                 <div class="alert alert-
26 danger">
27                     <?= session()-
28 >getFlashdata('error'); ?>
29                 </div>
30             <?php endif; ?>
31
32             <?= validation_list_errors() ?>
33
34             <?= form_open('buku/update/' .
35 $post['id']); ?>
36
37             <input type="hidden"
38 name="_method" value="PUT">
39
40             <div class="form-group">
41                 <label
42 for="judul">Judul</label>
43                 <input type="text"
44 class="form-control" id="judul" name="judul"
45 placeholder="Masukkan Judul" value="<?=
$post['judul'] ?>">
46             </div>
47
48             <div class="form-group">
49                 <label
50 for="penulis">Penulis</label>
51                 <input type="text"
52 class="form-control" id="penulis" name="penulis"
53 placeholder="Masukkan Penulis" value="<?=
$post['penulis'] ?>">
54             </div>
55
56             <div class="form-group">
57                 <label
58 for="penerbit">Penerbit</label>
59                 <input type="text"
60 class="form-control" id="penerbit" name="penerbit"
61 placeholder="Masukkan Penerbit" value="<?=
62 $post['penerbit'] ?>">
63             </div>

```



```

46         <div class="form-group">
47             <label
for="tahun_terbit">Tahun Terbit</label>
48             <input type="number"
class="form-control" id="tahun_terbit"
name="tahun_terbit" placeholder="Masukkan Tahun
Terbit" value="<?= $post['tahun_terbit'] ?>">
49             </div>
50             <div class="form-group text-
right">
51                 <button class="btn btn-
success" style="background-color:
#dc3545">Update</button>
52                 <a href="<?=
base_url('home') ?>" class="btn btn-secondary"
style="background-color: #000000">Kembali</a>
53             </div>
54             <?= form_close(); ?>
55
56         </div>
57     </div>
58 </div>
59 </div>
60 </div>
61
62 <?= $this->endSection() ?>
63
64 <?= $this->section('extra-js') ?>
65 <!-- include summernote css/js -->
66 <link
href="https://cdn.jsdelivr.net/npm/summernote@0.8.15
/dist/summernote-bs4.min.css" rel="stylesheet">
67 <script
src="https://cdn.jsdelivr.net/npm/summernote@0.8.15/
dist/summernote-bs4.min.js"></script>
68 <script>
69     $(document).ready(function() {
70         $('#post_content').summernote({
71             tabsize: 2,
72             height: 500
73         });
74     });
75 </script>

```

76	<?= \$this->endSection() ?>
----	-----------------------------

Tabel 9. Source Code Soal 1 vw_edit.php

vw_login.php

1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta name="viewport" content="width=device-
	width, initial-scale=1.0">
7	<title>Login · MyLibrary</title>
8	<style>
9	body {
10	display: flex;
11	justify-content: center;
12	align-items: center;
13	min-height: 100vh;
14	padding: 0;
15	margin: 0;
16	font-family: Arial, sans-serif;
17	background-image:
	url('https://plus.unsplash.com/premium_photo-
	1677567996070-
	68fa4181775a?q=80&w=2072&auto=format&fit=crop&ixlib=
	rb-
	4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8f
	Hx8fA%3D%3D');
18	background-repeat: no-repeat;
19	background-attachment: fixed;
20	background-position: center;
21	background-size: cover;
22	}
23	
24	.form-signin {
25	width: 100%;
26	max-width: 330px;
27	padding: 15px;
28	margin: auto;
29	background: #fff;
30	border-radius: 8px;
31	box-shadow: 0 4px 6px rgba(0, 0, 0,
	0.1);

```
32         position: relative;
33     }
34
35     .form-signin img {
36         margin-bottom: 20px;
37     }
38
39     .form-signin h1 {
40         margin-bottom: 10px;
41         font-size: 24px;
42         font-weight: normal;
43     }
44
45     .form-signin p {
46         margin-bottom: 20px;
47         font-size: 14px;
48         color: #6c757d;
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 #ccc;
57         border-radius: 4px;
58         box-sizing: border-box;
59     }
60
61     .form-signin button {
62         width: 50%;
63         padding: 10px;
64         background-color: #f0768b;
65         color: white;
66         border: none;
67         border-radius: 4px;
68         cursor: pointer;
69     }
70
71     .form-signin button:hover {
72         background-color: #f0768b;
73     }
74
```

```
75     .form-signin .alert {
76         padding: 10px;
77         margin-bottom: 10px;
78         border-radius: 4px;
79         color: #856404;
80         background-color: #fff3cd;
81         border: 1px solid #ffeeba;
82         position: relative;
83     }
84
85     .form-signin .alert .btn-close {
86         position: absolute;
87         top: 10px;
88         right: 10px;
89         background: none;
90         border: none;
91         font-size: 16px;
92         cursor: pointer;
93     }
94
95     .form-signin .text-muted {
96         color: #6c757d;
97         margin-top: 20px;
98     }
99
100    .btn-register {
101        position: absolute;
102        bottom: 10px;
103        right: 10px;
104        font-size: 12px;
105        color: #007bff;
106        cursor: pointer;
107    }
108
109    .btn-register:hover {
110        text-decoration: underline;
111    }
112    </style>
113 </head>
114
115 <body>
116
117     <main class="form-signin">
```

```

118      <?php if (!empty(session()-
>getFlashdata('error')) || session()-
>get('requires_login')) : ?>
119          <div class="alert">
120              <?php if (session()-
>get('requires_login')) : ?>
121                  Login terlebih dahulu!
122              <?php else : ?>
123                  <?= session()-
>getFlashdata('error'); ?>
124              <?php endif; ?>
125              <button type="button" class="btn-
close" aria-label="Close"
onclick="this.parentElement.style.display='none';">&
times;</button>
126          </div>
127          <?php endif; ?>
128          <form method="post" action="<?= base_url();
?>/login/process">
129              <?= csrf_field(); ?>
130              <h1 class="h3 mb-3 fw-normal">Login</h1>
131              <p>Register dulu apabila belum memiliki
akun</p>
132
133              <input type="text" name="username"
id="username" placeholder="Username" required
autofocus>
134              <input type="password" name="password"
id="password" placeholder="Password" required>
135
136              <button type="submit">Login</button>
137          </form>
138          <span class="btn-register"
onclick="window.location.href='<?php echo
base_url('register'); ?>'">Register</span>
139      </main>
140
141 </body>
142
143 </html>

```

Tabel 10. Source Code Soal 1 vw_login.php

vw_register.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, initial-scale=1">
7     <title>Register · MyLibrary</title>
8
9     <!-- Custom styles for this template -->
10    <style>
11        body,
12        html {
13            margin: 0;
14            padding: 0;
15            font-family: Arial, sans-serif;
16            height: 100%;
17            display: flex;
18            flex-direction: column;
19            background-image:
url('https://images.unsplash.com/photo-
1457369804613-
52c61a468e7d?q=80&w=2070&auto=format&fit=crop&ixlib
=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8
fHx8fA%3D%3D');
20            background-repeat: no-repeat;
21            background-attachment: fixed;
22            background-position: center;
23            background-size: cover;
24            background-color: #f5f5f5;
25            padding: 20px; /* Add padding to body
*/
26        }
27
28        .container {
29            max-width: 600px;
30            margin: auto;
31            padding: 20px;
32            background: #fff;
33            border-radius: 8px;
```

34	box-shadow: 0 4px 6px rgba(0, 0, 0,
35	0.1);
36	margin-top: 40px;
37	}
38	h1 {
39	font-size: 24px;
40	margin-bottom: 20px;
41	text-align: center;
42	}
43	
44	.form-label {
45	display: block;
46	margin-bottom: 5px;
47	font-weight: bold;
48	}
49	
50	.form-control {
51	width: 100%;
52	padding: 10px;
53	margin-bottom: 10px;
54	border: 1px solid #ccc;
55	border-radius: 4px;
56	box-sizing: border-box; /* Ensure
	padding is included in width calculation */
57	}
58	
59	.btn {
60	width: 100%;
61	padding: 10px;
62	background-color: #f0768b;;
63	color: white;
64	border: none;
65	border-radius: 4px;
66	cursor: pointer;
67	}
68	
69	.btn:hover {
70	background-color: #f0768b;;
71	}
72	
73	.alert {
74	padding: 10px;

```

75         margin-bottom: 10px;
76         border-radius: 4px;
77         color: #856404;
78         background-color: #fff3cd;
79         border: 1px solid #ffeeba;
80     }
81
82     .footer {
83         margin-top: auto;
84         padding: 10px 0;
85         text-align: center;
86         background-color: #e9ecef;
87     }
88
89     .footer .container {
90         max-width: none;
91     }
92 </style>
93 </head>
94
95 <body>
96
97     <main class="flex-shrink-0">
98         <div class="container">
99             <h1>Register Form</h1>
100             <hr />
101             <?php if (!empty(session()-
>getFlashdata('error')) : ?>
102                 <div class="alert">
103                     <h4>Periksa Entrian Form</h4>
104                     <?= session()-
>getFlashdata('error'); ?>
105                 </div>
106             <?php endif; ?>
107             <form method="post" action="<?=
base_url(); ?>/register/process">
108                 <?= csrf_field(); ?>
109                 <div class="mb-3">
110                     <label for="username"
class="form-label">Username</label>
111                     <input type="text" class="form-
control" id="username" name="username" required>
112                 </div>

```


113	<div class="mb-3"> </div>
114	<div class="mb-3"> <div class="form-label">Password</div> <div class="form-control" id="password" name="password" required=""> </div> </div>
115	<div class="mb-3"> <div class="form-label">Confirm Password</div> <div class="form-control" id="password_conf" name="password_conf" required=""> </div> </div>
116	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
117	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
118	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
119	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
120	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
121	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
122	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
123	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
124	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
125	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
126	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
127	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
128	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
129	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
130	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
131	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
132	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
133	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>
134	<div class="mb-3"> <div class="form-label">Email</div> <div class="form-control" id="email" name="email" required=""> </div> </div>

Tabel 11. Source Code Soal 1 vw_register.php

BukuModel.php

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class BukuModel extends Model

```

8 {
9     protected $table = 'buku';
10    protected $allowedFields = ['judul', 'penulis' ,
    'penerbit', 'tahun_terbit'];
11    protected $createdField = 'created_at';
12    protected $updatedField = 'updated_at';
13    protected $deletedField = 'deleted_at';
14
15    protected $validationRules = ([
16        'judul' => [
17            'rules' => 'required|string',
18            'errors' => [
19                'required' => 'Judul harus diisi.',
20                'string' => 'Judul harus berupa
string.'
21            ]
22        ],
23        'penulis' => [
24            'rules' => 'required|string',
25            'errors' => [
26                'required' => 'Penulis harus
diisi.',
27                'string' => 'Penulis harus berupa
string.'
28            ]
29        ],
30        'penerbit' => [
31            'rules' => 'required|string',
32            'errors' => [
33                'required' => 'Penerbit harus
diisi.',
34                'string' => 'Penerbit harus berupa
string.'
35            ]
36        ],
37        'tahun_terbit' => [
38            'rules' =>
'required|integer|greater_than[1800]|less_than[2024]
',
39            'errors' => [
40                'required' => 'Tahun terbit harus
diisi.',

```

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

Tabel 12. Source Code Soal 1 BukuModel.php

UsersModel.php

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class UsersModel extends <u>Model</u>
8	{
9	protected \$table = "users";
10	protected \$returnType = "object";
11	protected \$useTimestamps = true;
12	protected \$allowedFields = ['username',
13	'password', 'email'];
14	}

Tabel 13. Source Code Soal 1 UsersModel.php

Buku.php

1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;
6	
7	class Buku extends <u>Seeder</u>
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
18         $this->db->table('buku')->insert([
19             'judul' => 'Tentang Kamu',
20             'penulis' => 'Tere Liye',
21             'penerbit' => 'Republika',
22             'tahun_terbit' => '2016'
23         ]);
24     }
25 }

```

Tabel 14. Source Code Soal 1 Buku.php di Seeds

UserSeeder.php

```

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' =>
14             'farlyhaydy123@gmail.com',
15             'password' => password_hash('123',
16             PASSWORD_DEFAULT),
17         ]);
18     }
19 }

```

Tabel 15. Source Code Soal 1 UserSeeder.php

perpustakaan.sql

```

1  -- phpMyAdmin SQL Dump
2  -- version 5.2.1
3  -- https://www.phpmyadmin.net/

```

```

4      --
5      -- Host: 127.0.0.1
6      -- Waktu pembuatan: 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
15     /*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
16     /*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
17     /*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
18     /*!40101 SET NAMES utf8mb4 */;
19
20     --
21     -- Database: `perpustakaan`
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
    COLLATE=utf8mb4_general_ci;
37
38     --

```

```

39 -- Dumping data untuk tabel `buku`
40 --
41
42 INSERT INTO `buku` (`id`, `judul`, `penulis`,
43 `penerbit`, `tahun_terbit`) VALUES
44 (15, 'Laut Bercerita', 'Leila S. Chudori',
45 'Gramedia', '2017'),
46 (16, 'Tentang Kamu', 'Tere Liye', 'Republika',
47 '2016');
48
49 -----
50 -----
51
52 --
53 -- Struktur dari tabel `migrations`
54 --
55
56 CREATE TABLE `migrations` (
57   `id` bigint(20) UNSIGNED NOT NULL,
58   `version` varchar(255) NOT NULL,
59   `class` varchar(255) NOT NULL,
60   `group` varchar(255) NOT NULL,
61   `namespace` varchar(255) NOT NULL,
62   `time` int(11) NOT NULL,
63   `batch` int(11) UNSIGNED NOT NULL
64 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
65 COLLATE=utf8mb4_general_ci;
66
67 --
68 -- Dumping data untuk tabel `migrations`
69 --
70
71 INSERT INTO `migrations` (`id`, `version`, `class`,
72 `group`, `namespace`, `time`, `batch`) VALUES
73 (1, '2024-06-03-103310',
74 'App\\Database\\Migrations\\Users', 'default',
75 'App', 1718081782, 1),
76 (2, '2024-06-03-122804',
77 'App\\Database\\Migrations\\Buku', 'default',
78 'App', 1718081782, 1),
79 (4, '2024-06-11-173406',
80 'App\\Database\\Migrations\\Users', 'default',
81 'App', 1718127395, 3);

```

```

70
71 -- -----
72 -----
73 --
74 -- Struktur dari tabel `users`
75 --
76
77 CREATE TABLE `users` (
78   `id` int(10) UNSIGNED NOT NULL,
79   `username` varchar(255) NOT NULL,
80   `email` varchar(255) NOT NULL,
81   `password` text NOT NULL,
82   `created_at` datetime DEFAULT NULL,
83   `updated_at` datetime DEFAULT NULL
84 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
  COLLATE=utf8mb4_general_ci;
85
86 --
87 -- Dumping data untuk tabel `users`
88 --
89
90 INSERT INTO `users` (`id`, `username`, `email`,
91   `password`, `created_at`, `updated_at`) VALUES
92   (4, 'Farly', 'farlyhaydy123@gmail.com',
93     '$2y$10$xYhsftGqi132/1Y8Qrypge2VGyR2bS0TEBFpXTa4sXU
94     KI4GwnEoh2', NULL, NULL);
95
96
97 --
98 -- Indexes for dumped tables
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`);

```

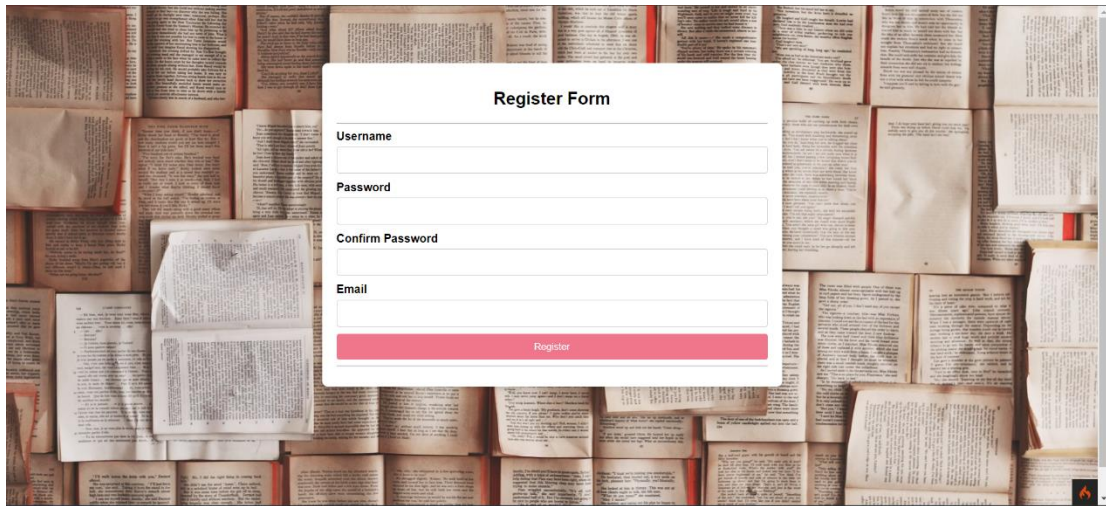
```

108
109 --
110 -- Indeks untuk tabel `users`
111 --
112 ALTER TABLE `users`
113     ADD PRIMARY KEY (`id`),
114     ADD UNIQUE KEY `email` (`email`);
115
116 --
117 -- AUTO_INCREMENT untuk tabel yang dibuang
118 --
119
120 --
121 -- AUTO_INCREMENT untuk tabel `buku`
122 --
123 ALTER TABLE `buku`
124     MODIFY `id` bigint(20) UNSIGNED NOT NULL
125     AUTO_INCREMENT, AUTO_INCREMENT=19;
126
127 --
128 -- AUTO_INCREMENT untuk tabel `migrations`
129 --
130 ALTER TABLE `migrations`
131     MODIFY `id` bigint(20) UNSIGNED NOT NULL
132     AUTO_INCREMENT, AUTO_INCREMENT=5;
133
134 --
135 -- AUTO_INCREMENT untuk tabel `users`
136 --
137 ALTER TABLE `users`
138     MODIFY `id` int(10) UNSIGNED NOT NULL
139     AUTO_INCREMENT, AUTO_INCREMENT=5;
140 COMMIT;
141
142 /*!40101 SET
143 CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
144 /*!40101 SET
145 CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
146 */;
147 /*!40101 SET
148 COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

Tabel 16. Source Code Soal 1 perpustakaan.sql

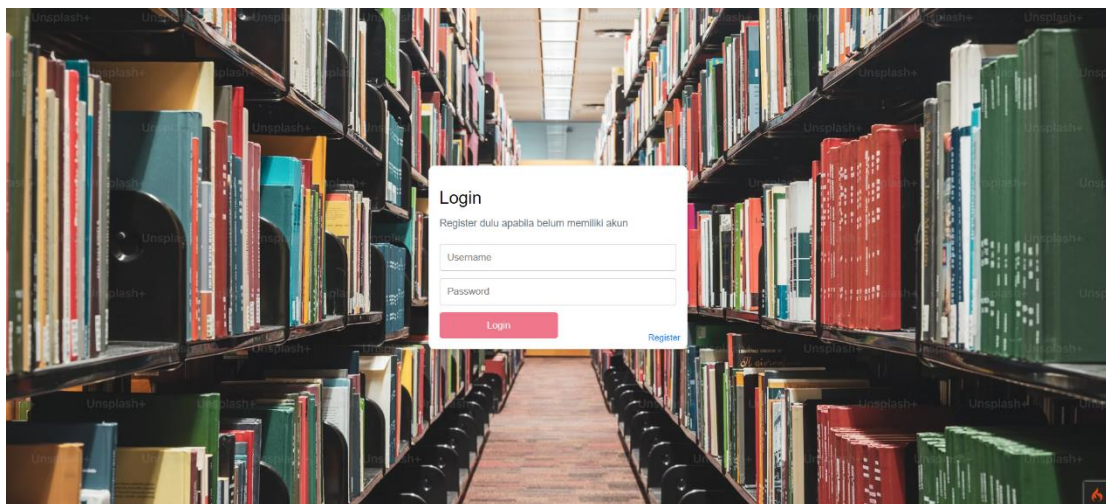
B. Output Program



The screenshot shows a 'Register Form' overlay on a background of open books. The form has a white background and a pink border. It contains the following fields and buttons:

- 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** (Pink button)

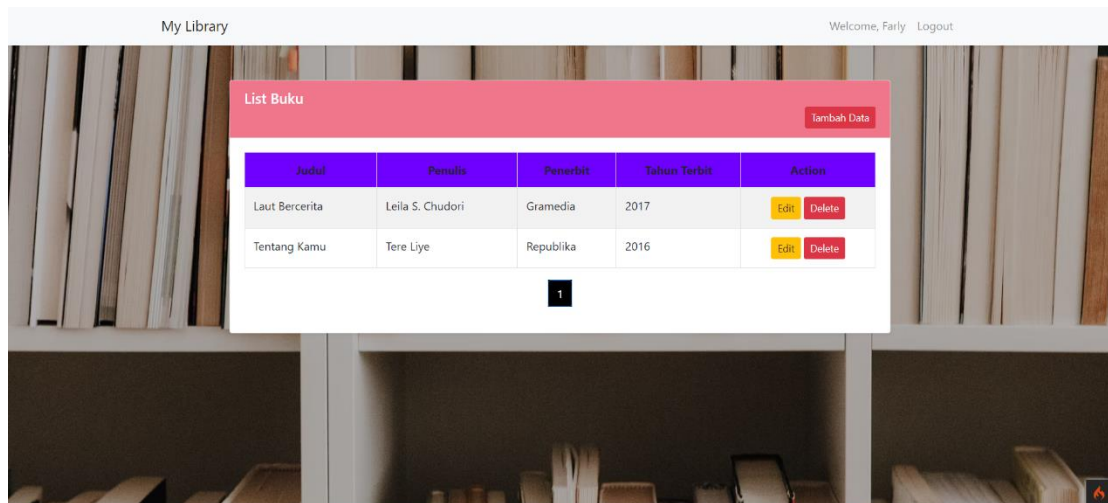
Gambar 2. Screenshot Hasil Jawaban Soal tampilan register



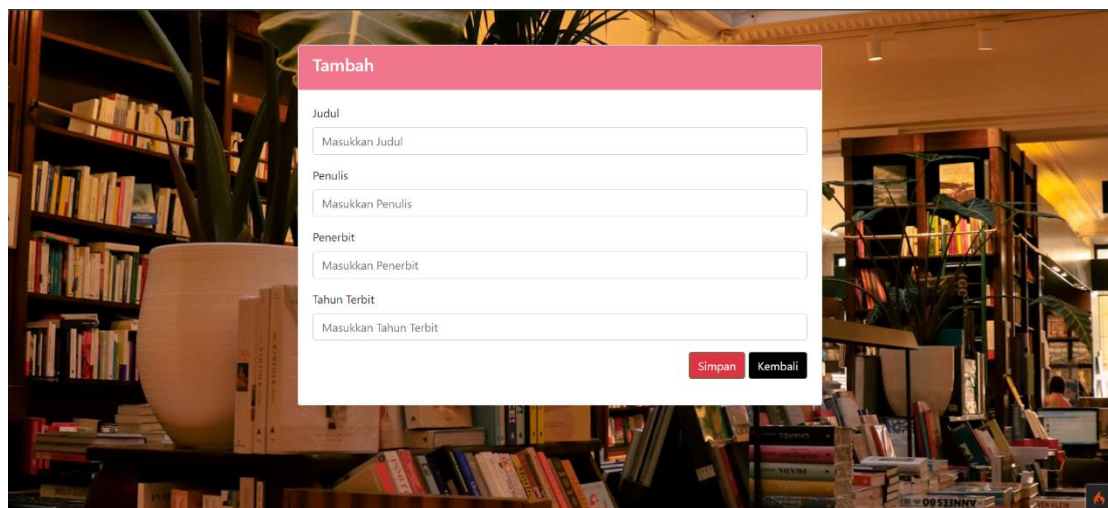
The screenshot shows a 'Login' form overlay on a background of bookshelves. The form has a white background and a pink border. It contains the following fields and buttons:

- Login** (Title)
- Register dulu apabila belum memiliki akun (Text)
- Username** (Label) with a text input field.
- Password** (Label) with a text input field.
- Login** (Pink button)
- Register** (Blue link)

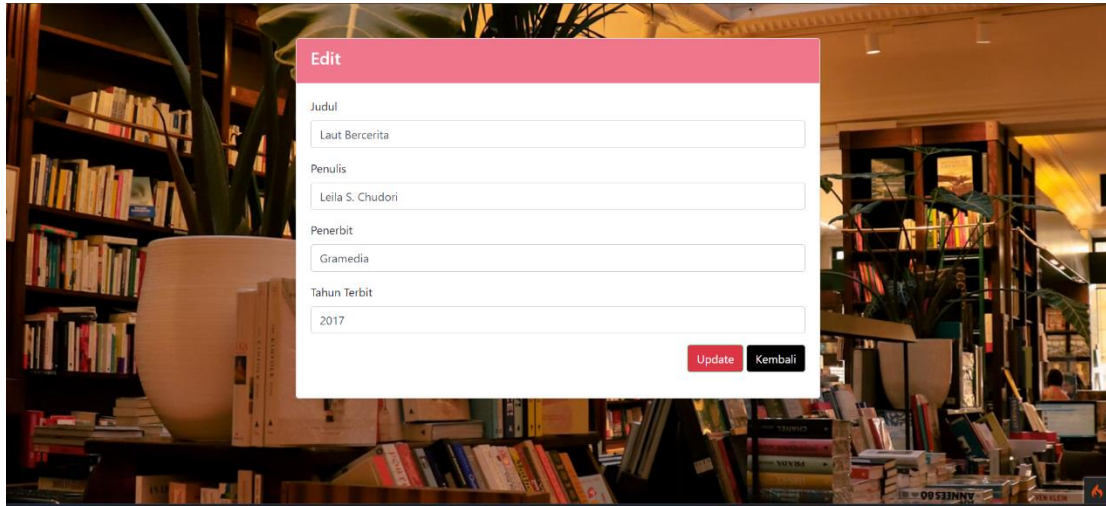
Gambar 3. Screenshot Hasil Jawaban Soal 1 tampilan login



Gambar 4. Screenshot Hasil Jawaban Soal 1 tampilan home



Gambar 5. Screenshot Hasil Jawaban Soal 1 tampilan tambah buku



Gambar 6. Screenshot Hasil Jawaban Soal 1 tampilan edi atau mengupdatet buku

C. Pembahasan

Untuk 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.

Untuk Buku.php di controller

Kode tersebut ditulis menggunakan framework CodeIgniter untuk menangani manajemen buku (Buku controller). Mari kita bahas fungsinya secara umum:

1. Konstruktor dan Dependensi

- Class Buku mewarisi class Controller dari CodeIgniter.
- Konstruktor (__construct) menginstansiasi model BukuModel untuk akses database dan library form dan url untuk keperluan formulir.

2. Method Index (Kemungkinan untuk Halaman Utama)

- Method index mirip dengan login, mengambil data buku dengan function paginate dan mengembalikan view vw_buku.
- Dugaan sementara, method ini digunakan untuk menampilkan halaman utama setelah login.

3. Method Create (Membuat Buku Baru)

- Method create melakukan pengecekan apakah user sudah login. Jika belum, diarahkan ke halaman login.
- Jika sudah login, maka akan menampilkan view vw_add untuk form pembuatan buku baru. Judul view diberi label "Create new post".

4. Method Store (Menyimpan Data Buku Baru)

- Method store menerima data submit form melalui method POST.
- Data yang diambil meliputi judul, penulis, penerbit, dan tahun terbit.
- Terdapat validasi data menggunakan method validateData (fungsi validasi kemungkinan dibuat terpisah) berdasarkan aturan validasi yang terdapat pada model \$this->model->validationRules.
- Jika validasi gagal, diarahkan kembali ke method create.
- Jika validasi berhasil, data yang sudah tervalidasi disimpan ke database menggunakan function save pada model.
- Setelah berhasil disimpan,flashdata session untuk menampilkan pesan sukses dibuat dan halaman dialihkan ke halaman utama (base_url('home')).

5. Method Edit (Menampilkan Form Edit Buku)

- Method edit menerima id buku sebagai parameter.
- Method ini melakukan pengecekan apakah user sudah login. Jika belum, diarahkan ke halaman login.
- Kemudian, method find pada model digunakan untuk mengambil detail buku berdasarkan id.
- Jika buku tidak ditemukan, flashdata session untuk menampilkan pesan error dibuat dan halaman dialihkan kembali (redirect()->back()).
- Jika buku ditemukan, data buku tersebut beserta judul "Edit Post" akan dikirim ke view vw_edit.

6. Method Update (Proses Update Data Buku)

- Method update menerima id buku sebagai parameter.
- Method ini melakukan pengecekan apakah user sudah login. Jika belum, diarahkan ke halaman login.
- Mirip dengan method edit, method find digunakan untuk mengambil detail buku berdasarkan id.
- Jika buku tidak ditemukan, flashdata session untuk menampilkan pesan error dibuat dan halaman dialihkan kembali (redirect()->back()).
- Jika buku ditemukan, data submit form melalui method POST akan divalidasi menggunakan method validateData dan aturan validasi dari model.
- Jika validasi gagal, diarahkan kembali ke method create.
- Jika validasi berhasil, data yang sudah tervalidasi akan digunakan untuk update data buku di database menggunakan function update pada model.
- Terdapat pengecekan apakah update berhasil. Jika berhasil, flashdata session untuk menampilkan pesan sukses dibuat dan halaman dialihkan ke halaman utama (base_url('home')). Jika gagal, flashdata session untuk menampilkan pesan error dibuat dan halaman dialihkan kembali (redirect()->back()).

7. Method Delete (Menghapus Buku)

- Method delete menerima id buku sebagai parameter.
- Method ini melakukan pengecekan apakah user sudah login. Jika belum, diarahkan ke halaman utama.

- Method delete pada model digunakan untuk menghapus data buku berdasarkan id.
- Terdapat pengecekan apakah delete berhasil. Jika berhasil, flashdata session untuk menampilkan pesan sukses dibuat dan halaman dialihkan ke halaman utama (base_url('home')). Jika gagal, flashdata session untuk menampilkan pesan error dibuat dan halaman dialihkan ke halaman utama (base_url('home')).

Untuk 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.

Untuk Login.php

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

1. Method Index (Form Login)

- Method index kemungkinan digunakan untuk menampilkan halaman login.
- Method ini mengembalikan view vw_login.

2. Method Process (Autentikasi Login)

- Method process menangani proses login setelah pengguna submit form login.
- Method ini menggunakan model UserModel untuk berinteraksi dengan database.
- Data username dan password diambil dari request (\$this->request->getVar('username') dan \$this->request->getVar('password')).

- Kemudian method where dan first digunakan untuk mencari data user di database berdasarkan username.
- Jika user ditemukan (\$dataUser tidak kosong):
 - Password yang diinputkan dibandingkan dengan password yang tersimpan di database menggunakan function password_verify.
 - Fungsi ini direkomendasikan untuk verifikasi password karena menggunakan teknik hashing yang lebih aman.
 - Jika password cocok:
 - Data user (username) dan flag logged_in disimpan ke session.
 - Pengguna dialihkan ke halaman utama (base_url('home')).
 - Jika password salah:
 - Flashdata session untuk menampilkan pesan error "Username & Password Salah" dibuat.
 - Pengguna dialihkan kembali ke halaman login (redirect()->back()).
- Jika user tidak ditemukan (\$dataUser kosong):
 - Flashdata session untuk menampilkan pesan error "Username & Password Salah" dibuat.
 - Pengguna dialihkan kembali ke halaman login (redirect()->back()).

3. Method Logout

- Method logout digunakan untuk proses keluar (logout) pengguna.
- Method ini akan menghapus semua data session menggunakan session()->destroy().
- Setelah itu, pengguna akan dialihkan kembali ke halaman login (redirect()->to('/login')).

Untuk 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')`).

Untuk 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.

Untuk 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.

Untuk `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.

Untuk 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 menunjukan kemungkinan Summernote digunakan sebagai text editor WYSIWYG.

Untuk `vw_login.php`

Kode tersebut menampilkan formulir untuk login ke 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 login.
- Bagian `<body>` berisi konten utama yaitu formulir login.

2. Form Login

- Formulir login menggunakan class form-signin dan dibungkus dengan tag <main>.
- Formulir menggunakan method POST dan action ke URL <?= base_url(); ?>/login/process untuk memproses data login.
- Formulir dilengkapi dengan token CSRF untuk keamanan.
- Elemen formulir meliputi:
 - Judul "Login".
 - Pesan "Register dulu apabila belum memiliki akun".
 - Username input field.
 - Password input field.
 - Tombol "Login" untuk submit formulir.
- Link "Register" di bagian bawah formulir untuk menuju halaman registrasi.

3. Pesan Error

- Kode ini mengecek keberadaanflashdata session dengan key "error" atau session "requires_login".
- Jika ada, maka akan menampilkan pesan error di dalam alert berwarna kuning.
 - Pesan khusus ditampilkan jika session "requires_login" bernilai true ("Login terlebih dahulu!").
 - Pesan dariflashdata session "error" ditampilkan jika ada.
- Tombol "x" untuk menutup alert.

4. Styling CSS

- Kode CSS mengatur tampilan keseluruhan halaman login, meliputi:
 - Gaya formulir login dengan class form-signin.
 - Gaya elemen formulir seperti input text, button, dan alert.
 - Penggunaan background image untuk seluruh halaman.

Untuk 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 keberadaan flashdata 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.

Untuk 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 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]`).

Untuk 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.

Untuk 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.

Untuk 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.

Untuk 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.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/farlyhaydyhdjalil/Praktikum-PEMROGRAMANWEBII/tree/e4654830b243167b55eeee4883c41280dcfc52c8/Modul%206/PRAK601>