

**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 1 tampilan register	41
Gambar 3. Screenshot Hasil Jawaban Soal 1 tampilan login.....	42
Gambar 4. Screenshot Hasil Jawaban Soal 1 tampilan home	42
Gambar 5. Screenshot Hasil Jawaban Soal 1 tampilan tambah buku	43
Gambar 6. Screenshot Hasil Jawaban Soal 1 tampilan edi atau mengupdate buku	43

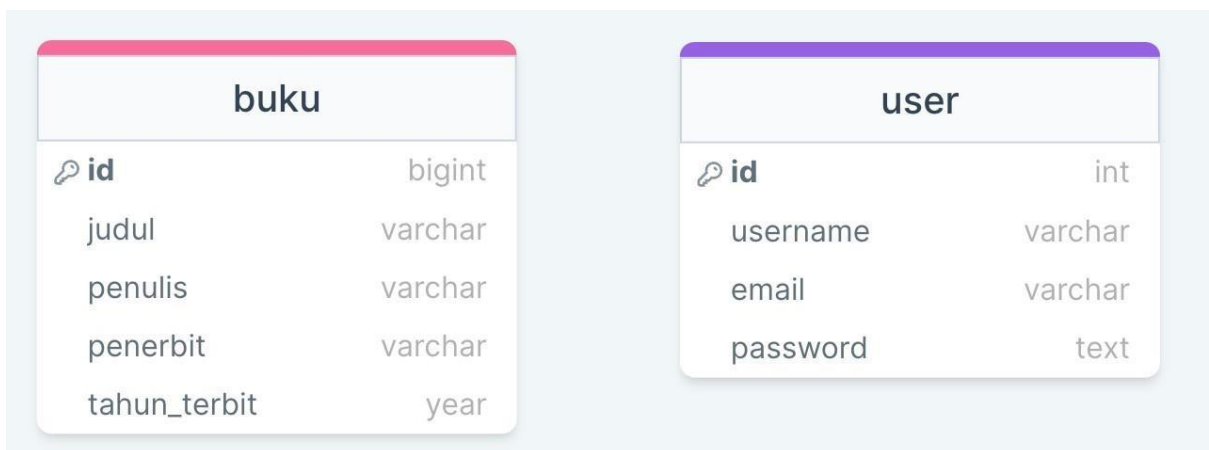
DAFTAR TABEL

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.....	30
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.....	37
Tabel 16. Source Code Soal 1 perpustakaan.sql	41

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->paginate(10),
26              'pager' => $this->model->pager,
27              'title' => 'POST LIST'
28          ];
29
30          return view('vw_login', $data);
31      }
32
33      public function index()
34      {
35          if (!session()->get('logged_in')) {
36              return redirect()->to(base_url('/login'));
37          }
38          $data = [
39              'posts' => $this->model->paginate(10),
40              'pager' => $this->model->pager,
41              'title' => 'POST LIST'
42          ];
43
44          return view('vw_buku', $data);
45      }
46
47      public function create()
48      {
49          // Check if user is authenticated
50          if (!session()->get('logged_in')) {

```



```

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

```

76     return redirect()->to(base_url('home'));
77 }
78
79 public function edit($id)
80 {
81     // Check if user is authenticated
82     if (!session()->get('logged_in')) {
83         return redirect()-
>to(base_url('/login'));
84     }
85     $post = $this->model->find($id);

```

```

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

```

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

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

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

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
diisi',
23                     'min_length' => '{field} Minimal
4 Karakter',
24                     'max_length' => '{field}
Maksimal 20 Karakter',
25                     'is_unique' => 'Username sudah
digunakan sebelumnya'
26                 ]
27             ],
28             'password' => [
29                 'rules' =>
30                 'required|min_length[0]|max_length[50]',
31                 'errors' => [
32                     'required' => '{field} Harus
diisi',
33                     'min_length' => '{field} Minimal
0 Karakter',
34                     'max_length' => '{field}
Maksimal 50 Karakter',
35                 ]
36             ],
37             'password_conf' => [
38                 'rules' => 'matches[password]',
39                 'errors' => [
40                     'matches' => 'Konfirmasi
Password tidak sesuai dengan password',
41                 ]
42             ],
43             'email' => [

```

	'rules' =>
43	'required min_length[4] max_length[100] valid_email'
44	,
	'errors' => [
	'required' => '{field} Harus
45	diisi',
	'min_length' => '{field} Minimal
46	4 Karakter',
	'max_length' => '{field}
47	Maksimal 100 Karakter',
	'valid_email' => 'Email tidak
48	valid'
49]
50],
51))) {
	session()->setFlashdata('error', \$this->
52	validator->listErrors());
53	return redirect()->back()->withInput();
54	}
55	
56	\$users = new UsersModel();
57	
58	// Hash the password before storing it
	\$hashedPassword = password_hash(\$this->
59	request->getVar('password'), PASSWORD_DEFAULT);
60	
61	\$users->insert([
	'username' => \$this->request->
62	getVar('username'),
63	'password' => \$hashedPassword,
	'email' => \$this->request->
64	getVar('email')
65]);
66	
67	return redirect()->to('/login');
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
13	\$request, \$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
18	login page
19	return redirect()->to('/login');
20	}
21	}
22	public function after(RequestInterface
23	\$request, ResponseInterface \$response,
24	\$arguments = null)
25	{
26	// Do nothing after the request
27	}
28	}

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:
7	url('https://images.unsplash.com/photo-
8	1688320006653-
9	72c493042675?q=80&w=1887&auto=format&fit=crop&ixlib=
10	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"
20style="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-
25danger">
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
36for="judul">Judul</label>
37                            <input type="text"
38class="form-control" id="judul" name="judul"
39placeholder="Masukkan Judul">
40                        </div>
41                        <div class="form-group">
42                            <label
43for="penulis">Penulis</label>
44                            <input type="text"
45class="form-control" id="penulis" name="penulis"
46placeholder="Masukkan Penulis">

```

```

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

```

	<code>\$(document).ready(function() {</code>
64	<code> \$('#post_content').summernote({</code>
65	<code> tabsize: 2,</code>
66	<code> height: 500</code>
67	<code> });</code>
68	<code>});</code>
69	<code></script></code>
70	<code><?= \$this->endSection() ?></code>
71	
72	

Tabel 7. Source Code Soal 1 vw_add.php

vw_buku.php

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

```

19 target="#navbarNav" aria-controls="navbarNav" aria-
    expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-
20 icon"></span>
21 </button>
    <div class="collapse navbar-collapse"
22 id="navbarNav">
23 <ul class="navbar-nav ml-auto">
    <?php if (session()-
24 >get('logged_in')) : ?>
25 <li class="nav-item">
    <a class="nav-link"
    href="#">Welcome, <?= session()->get('username');
26 ?></a>
27 </li>
28 <li class="nav-item">
    <a class="nav-link"
29 href="<?= base_url('login/logout'); ?>">Logout</a>
30 </li>
31 <?php else : ?>
32 <li class="nav-item">
    <a class="nav-link"
33 href="<?= base_url('login'); ?>">Login</a>
34 </li>
35 <li class="nav-item">
    <a class="nav-link"
36 href="<?= base_url('register'); ?>">Register</a>
37 </li>
38 <?php endif; ?>
39 </ul>
40 </div>
41 </div>
42 </nav>
43
44 <div class="container mt-5">
45 <div class="row justify-content-center">
46 <div class="col-md-10">
47 <div class="card shadow-sm">
    <div class="card-header text-white"
48 style="background-color: #f0768b;">
49 <h5 class="mb-0">List Buku</h5>
    <a href="<?=
    base_url('buku/create'); ?>" class="btn btn-sm

```

```

float-right" style="background-color: #dc3545;
50 color: white; ">Tambah Data</a>
51         </div>
52         <div class="card-body">
53             <?php if (session()-
>getFlashdata('success')) : ?>
54                 <div class="alert alert-
success">
55                     <?= session()-
>getFlashdata('success'); ?>
56                 </div>
57             <?php endif; ?>
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
77	(\$posts as \$row) : ?>
78	<tr>
	<td><?=
79	\$row['judul']; ?></td>
	<td><?=
80	\$row['penulis']; ?></td>
	<td><?=
81	\$row['penerbit']; ?></td>
	<td><?=
82	\$row['tahun_terbit']; ?></td>
	<td
83	class="text-center">
	<form
	onsubmit="return confirm('Apakah Anda Yakin ?');"
	action="<?= base_url('buku/delete/' . \$row['id']);
84	?>" method="POST">
	<input
	type="hidden" name="{csrf_token}"
85	value="{csrf_hash}">
	<input
86	type="hidden" name="_method" value="DELETE">
	<a
	href="<?= base_url('buku/edit/' . \$row['id']); ?>"
87	class="btn btn-warning btn-sm">Edit
	<button
	type="submit" class="btn btn-danger btn-
88	sm">Delete</button>
89	</form>
90	</td>
91	</tr>
92	<?php endforeach;
93	?>
94	<?php else : ?>
	<tr>
95	<td colspan="5">
96	class="text-center">No post found.</td>
97	</tr>
98	<?php endif; ?>
99	</tbody>
100	</table>

101	<div class="d-flex justify-
102	content-center">
103	<?= \$pager->links(); ?>
104	</div>
105	</div>
106	</div>
107	</div>
108	</div>
109	</div>
110	
111	<?= \$this->endSection() ?>
112	
113	<?= \$this->section('extra-js') ?>
114	<script>
	\$(document).ready(function() {
115	\$('.pagination li').addClass('page-
	item').css('background-color', '#000000');
116	\$('.pagination li a').addClass('page-
117	link').css('background-color', '#000000');
118	});
	</script>
	<?= \$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"
19 style="background-color: #f0768b;">
20                     <h4>Edit</h4>
21                 </div>
22                 <div class="card-body">
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                     <?= validation_list_errors() ?>
32                     <?= form_open('buku/update/' .
33 $post['id']); ?>
34                     <input type="hidden"
35 name="__method" value="PUT">
36                     <div class="form-group">
37                         <label
38 for="judul">Judul</label>
39                         <input type="text"
40 class="form-control" id="judul" name="judul"
41 placeholder="Masukkan Judul" value="<?=
$post['judul'] ?>">
42                     </div>
43                     <div class="form-group">
44                         <label
45 for="penulis">Penulis</label>
46                         <input type="text"
47 class="form-control" id="penulis" name="penulis"
48 placeholder="Masukkan Penulis" value="<?=
49 $post['penulis'] ?>">
50                     </div>

```



```

42         <div class="form-group">
43             <label
for="penerbit">Penerbit</label>
44             <input type="text"
class="form-control" id="penerbit" name="penerbit"
placeholder="Masukkan Penerbit" value="<?=
$post['penerbit'] ?>">
45         </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.jsdelivrivr.net/npm/summernote@0.8.15
/dist/summernote-bs4.min.css" rel="stylesheet">
67 <script
src="https://cdn.jsdelivrivr.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</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=M3wxMjA3fDB8MHxwaG90byl1YWdlfHx8fGVufDB8f
	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         display: none;
77         padding: 10px;
78         margin-bottom: 10px;
79         border-radius: 4px;
80         color: #856404;
81         background-color: #fff3cd;
82         border: 1px solid #ffeeba;
83         position: relative;
84     }
85
86     .form-signin .alert .btn-close {
87         position: absolute;
88         top: 10px;
89         right: 10px;
90         background: none;
91         border: none;
92         font-size: 16px;
93         cursor: pointer;
94     }
95
96     .form-signin .text-muted {
97         color: #6c757d;
98         margin-top: 20px;
99     }
100
101     .btn-register {
102         position: absolute;
103         bottom: 10px;
104         right: 10px;
105         font-size: 12px;
106         color: #007bff;
107         cursor: pointer;
108     }
109
```

```

110         .btn-register:hover {
111             text-decoration: underline;
112         }
113     </style>
114 </head>
115
116 <body>
117
118     <main class="form-signin">
119         <div class="alert" id="initial-alert">
120             Login terlebih dahulu!
121             <button type="button" class="btn-close"
122             aria-label="Close"
123             onclick="this.parentElement.style.display='none';">&
124             times;</button>
125         </div>
126         <form method="post" action="<?= base_url();
127         ?>/login/process">
128             <?= csrf_field(); ?>
129             <h1 class="h3 mb-3 fw-normal">Login</h1>
130             <p>Register dulu apabila belum memiliki
131             akun</p>
132             <input type="text" name="username"
133             id="username" placeholder="Username" required
134             autofocus>
135             <input type="password" name="password"
136             id="password" placeholder="Password" required>
137             <button type="submit">Login</button>
138             </form>
139             <span class="btn-register"
140             onclick="window.location.href='<?php echo
141             base_url('register'); ?>'>Register</span>
142         </main>
143
144     <script>
145         // Show the initial alert when the page
146         loads
147         window.onload = function() {
148             document.getElementById('initial-
149             alert').style.display = 'block';
150         }

```

141	<code></script></code>
142	
143	<code></body></code>
144	
145	<code></html></code>

Tabel 10. Source Code Soal 1 vw_login.php

vw_register.php

1	<code><!doctype html></code>
2	<code><html lang="en" class="h-100"></code>
3	
4	<code><head></code>
5	<code> <meta charset="utf-8"></code>
6	<code> <meta name="viewport" content="width=device-</code>
	<code>width, initial-scale=1"></code>
7	<code> <title>Register</title></code>
8	
9	<code> <!-- Custom styles for this template --></code>
10	<code> <style></code>
11	<code> body,</code>
12	<code> html {</code>
13	<code> margin: 0;</code>
14	<code> padding: 0;</code>
15	<code> font-family: Arial, sans-serif;</code>
16	<code> height: 100%;</code>
17	<code> display: flex;</code>
18	<code> flex-direction: column;</code>
19	<code> background-image:</code>
	<code>url('https://images.unsplash.com/photo-</code>
	<code>1457369804613-</code>
	<code>52c61a468e7d?q=80&w=2070&auto=format&fit=crop&ixlib</code>
	<code>=rb-</code>
	<code>4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8</code>
	<code>fHx8fA%3D%3D');</code>
20	<code> background-repeat: no-repeat;</code>
21	<code> background-attachment: fixed;</code>
22	<code> background-position: center;</code>
23	<code> background-size: cover;</code>
24	<code> background-color: #f5f5f5;</code>
25	<code> padding: 20px; /* Add padding to body</code>
	<code>*/</code>

```
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,
0.1);
35         margin-top: 40px;
36     }
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

```


108	<form method="post" action="<?=
109	base_url(); ?>/register/process">
110	<?= csrf_field(); ?>
111	<div class="mb-3">
112	<label for="username"
113	class="form-label">Username</label>
114	<input type="text" class="form-
115	control" id="username" name="username" required>
116	</div>
117	<div class="mb-3">
118	<label for="password"
119	class="form-label">Password</label>
120	<input type="password"
121	class="form-control" id="password" name="password"
122	required>
123	</div>
124	<div class="mb-3">
125	<label for="password_conf"
126	class="form-label">Confirm Password</label>
127	<input type="password"
128	class="form-control" id="password_conf"
129	name="password_conf" required>
130	</div>
131	<div class="mb-3">
132	<label for="email" class="form-
133	label">Email</label>
134	<input type="email"
	class="form-control" id="email" name="email"
	required>
	</div>
	<div class="mb-3">
	<button type="submit"
	class="btn">Register</button>
	</div>
	</form>
	<hr />
	</div>
	</main>
	</body>
	</html>

Tabel 11. Source Code Soal 1 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' ,
    '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.',
	'integer' => 'Tahun terbit harus
41	berupa angka.',
	'greater_than' => 'Tahun terbit
42	harus lebih besar dari 1800.',
	'less_than' => 'Tahun terbit harus
43	lebih kecil dari 2024.'
]
44],
45]);
46	
47	protected \$skipValidation = false;
48	
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 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
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 }

```

17	}
----	---

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
  COLLATE=utf8mb4_general_ci;
65
66 --
67 -- Dumping data untuk tabel `migrations`
68 --
69
70 INSERT INTO `migrations` (`id`, `version`, `class`,
71   `group`, `namespace`, `time`, `batch`) VALUES

```

```

(1, '2024-06-03-103310',
68 'App\\Database\\Migrations\\Users', 'default',
  'App', 1718081782, 1),
(2, '2024-06-03-122804',
69 'App\\Database\\Migrations\\Buku', 'default',
  'App', 1718081782, 1),
(4, '2024-06-11-173406',
70 'App\\Database\\Migrations\\Users', 'default',
71 'App', 1718127395, 3);

72 -- -----
73 -----
74
75 --
76 -- Struktur dari tabel `users`
77 --
78
79 CREATE TABLE `users` (
80   `id` int(10) UNSIGNED NOT NULL,
81   `username` varchar(255) NOT NULL,
82   `email` varchar(255) NOT NULL,
83   `password` text NOT NULL,
84   `created_at` datetime DEFAULT NULL,
85   `updated_at` datetime DEFAULT NULL
86 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
87 COLLATE=utf8mb4_general_ci;
88 --
89 -- Dumping data untuk tabel `users`
90 --
91
92 INSERT INTO `users` (`id`, `username`, `email`,
  `password`, `created_at`, `updated_at`) VALUES
  (4, 'Farly', 'farlyhaydyl23@gmail.com',
93 '$2y$10$xYhsftGqi132/1Y8Qrypge2VGyR2bS0TEBFpXTa4sXU
  KI4GwnEoh2', NULL, NULL);
94
95 --
96 -- Indexes for dumped tables
97 --
98
99 --
100 -- Indeks untuk tabel `buku`

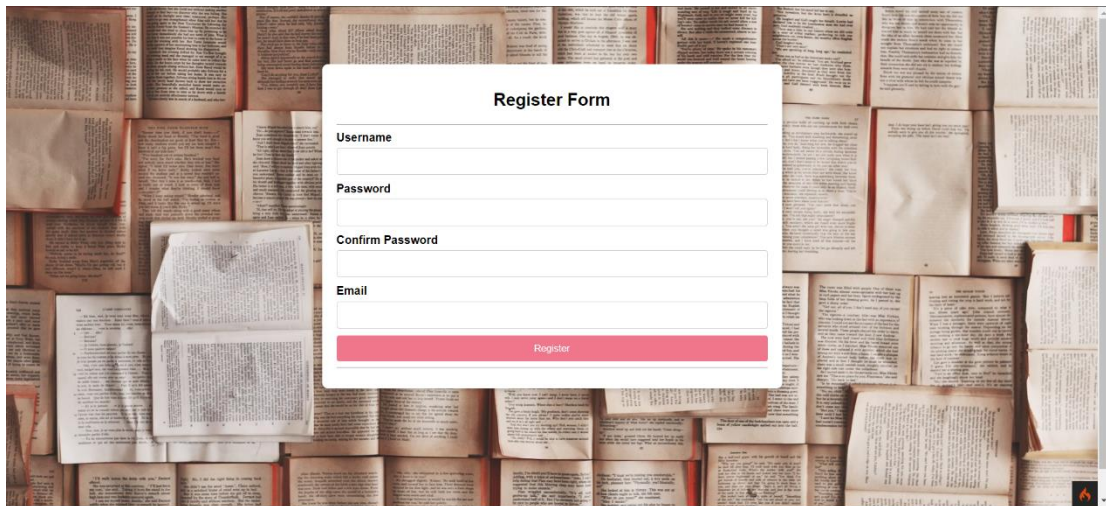
```

```
101 --
102 ALTER TABLE `buku`
103     ADD PRIMARY KEY (`id`);
104
105 --
106 -- Indeks untuk tabel `migrations`
107 --
108 ALTER TABLE `migrations`
109     ADD PRIMARY KEY (`id`);
110
111 --
112 -- Indeks untuk tabel `users`
113 --
114 ALTER TABLE `users`
115     ADD PRIMARY KEY (`id`),
116     ADD UNIQUE KEY `email` (`email`);
117
118 --
119 -- AUTO_INCREMENT untuk tabel yang dibuang
120 --
121
122 --
123 -- AUTO_INCREMENT untuk tabel `buku`
124 --
125     ALTER TABLE `buku`
126     MODIFY `id` bigint(20) UNSIGNED NOT NULL
127     AUTO_INCREMENT, AUTO_INCREMENT=19;
128 --
129 -- AUTO_INCREMENT untuk tabel `migrations`
130 --
131     ALTER TABLE `migrations`
132     MODIFY `id` bigint(20) UNSIGNED NOT NULL
133     AUTO_INCREMENT, AUTO_INCREMENT=5;
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;
```


140	<pre> /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */; /*!40101 SET 141 CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */; /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */; </pre>
-----	--

Tabel 16. Source Code Soal 1 perpustakaan.sql

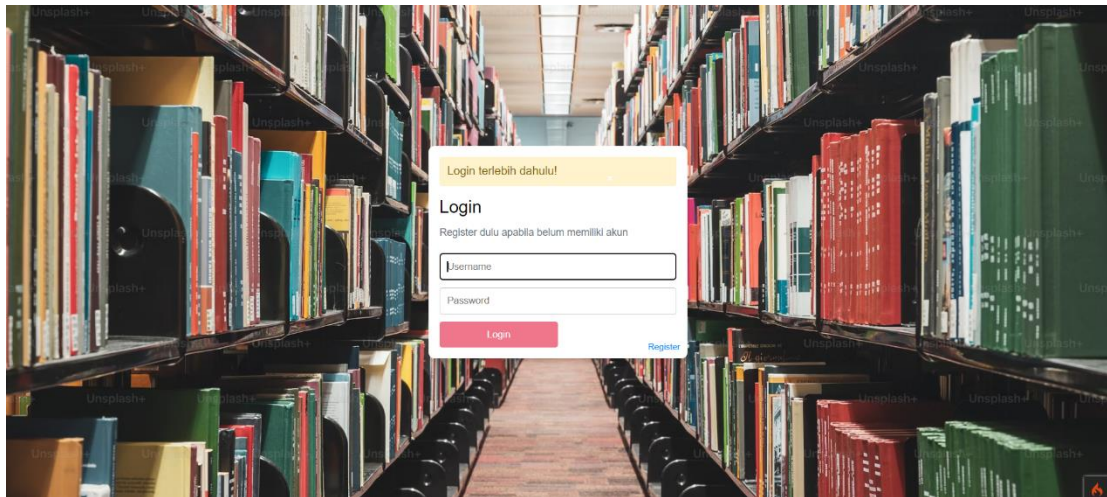
B. Output Program



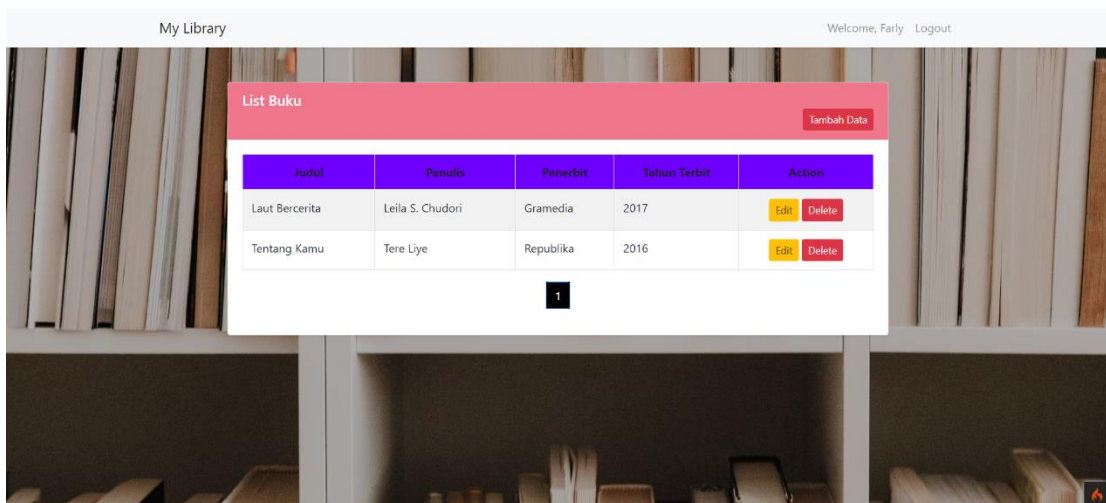
The image shows a 'Register Form' overlaying a background of numerous old, open books. The form is white with a thin black border and contains the following fields and elements:

- Register Form**: The title of the form, centered at the top.
- Username**: A text input field.
- Password**: A text input field.
- Confirm Password**: A text input field.
- Email**: A text input field.
- Register**: A pink button with the text 'Register' in white, located at the bottom of the form.

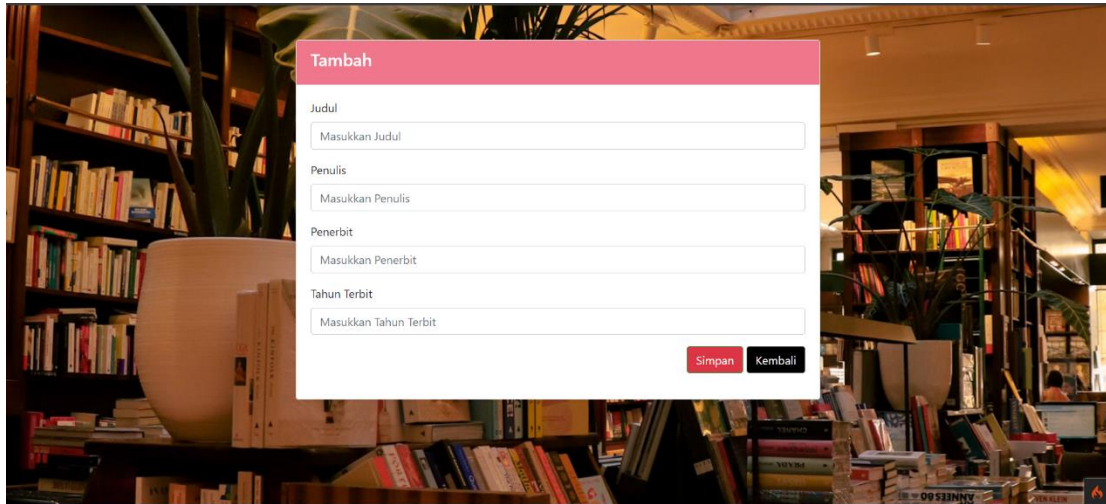
Gambar 2. Screenshot Hasil Jawaban Soal 1 tampilan register



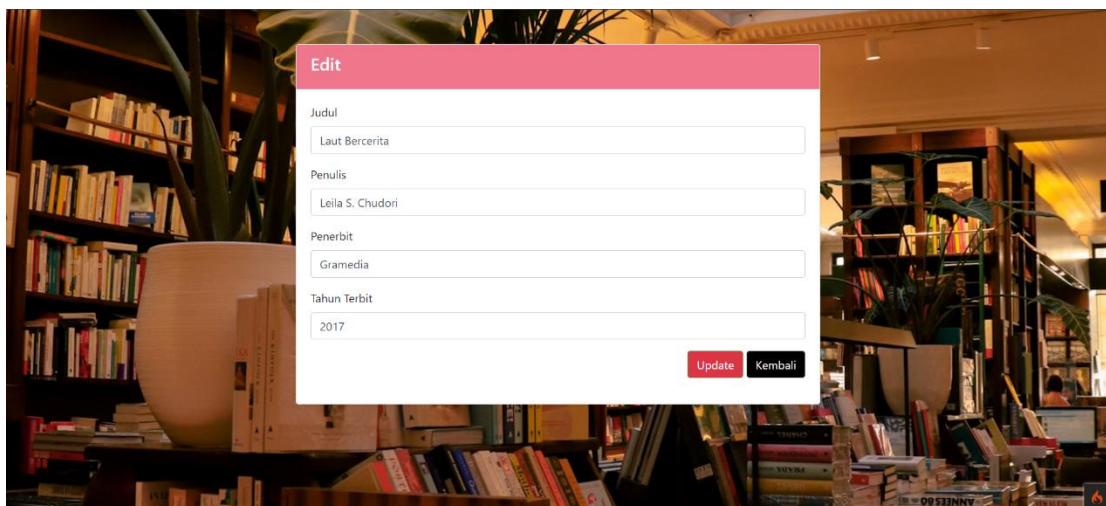
Gambar 3. Screenshot Hasil Jawaban Soal 1 tampilan login



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

1. Struktur Class

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

2. Method-method

- **login():** Method ini menampilkan halaman login. Sepertinya ada kesalahan pemanggilan model BukuModel untuk menampilkan daftar post (mungkin seharusnya menggunakan model lain).
- **index():** Method ini menampilkan halaman daftar buku. Method ini melakukan pengecekan apakah user sudah login (menggunakan session), jika belum akan di redirect ke halaman login. Kemudian method ini mengambil data buku dengan pagination (10 data per halaman) dan mengirimkannya ke view vw_buku bersama dengan judul halaman.

- **create():** Method ini menampilkan halaman create buku baru. Method ini juga melakukan pengecekan apakah user sudah login, jika belum akan di redirect ke halaman login. Terakhir, method ini mengirimkan judul halaman ke view vw_add.
- **store():** Method ini menangani proses penyimpanan data buku baru. Method ini mengambil data buku yang di submit dari form dan melakukan validasi data. Jika validasi gagal, user akan dikembalikan ke halaman create(). Jika validasi sukses, data buku akan disimpan ke database menggunakan model BukuModel. Setelah berhasil disimpan, method ini menampilkan flashdata sukses dan redirect ke halaman home.
- **edit(\$id):** Method ini menampilkan halaman edit buku. Method ini mengambil data buku berdasarkan id dari database dan melakukan pengecekan apakah data buku ditemukan. Jika tidak ditemukan, method ini menampilkan flashdata error dan redirect ke halaman sebelumnya. Jika data ditemukan, method ini mengirimkan data buku dan judul halaman ke view vw_edit.
- **update(\$id):** Method ini menangani proses update data buku. Method ini mengambil data buku yang di submit dari form dan melakukan validasi data. Jika validasi gagal, user akan dikembalikan ke halaman edit. Jika validasi sukses, data buku akan di update di database menggunakan model BukuModel. Method ini kemudian mengecek apakah update berhasil dan menampilkan flashdata sesuai hasil update (berhasil/gagal) dan redirect ke halaman home.
- **delete(\$id):** Method ini menangani proses delete buku. Method ini melakukan pengecekan apakah user sudah login dan id buku tersedia. Jika tidak valid, method ini akan redirect ke halaman home. Selanjutnya, method ini menghapus data buku berdasarkan id dari database menggunakan model BukuModel. Terakhir, method ini mengecek apakah delete berhasil dan menampilkan flashdata sesuai hasil delete (berhasil/gagal) dan redirect ke halaman home.

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 `UsersModel` 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 menunjukkan kemungkinan Summernote digunakan sebagai text editor WYSIWYG.

Untuk 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(); ?>.</li - 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 muncul saat halaman dimuat.
 - Tombol x untuk menutup alert menggunakan script Javascript.

4. 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').

5. Script Javascript

- Script Javascript di bagian akhir digunakan untuk menampilkan alert awal saat halaman web dimuat.
 - Fungsi window.onload dijalankan saat halaman selesai loading.
 - Script menampilkan elemen dengan id initial-alert.

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 $updatedAtField = '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 UserModel.php

Kode tersebut mendefinisikan model UserModel 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 UserModel extends Model mendefinisikan class UserModel 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/297035367c88ea5ca87df46c393bd2ddf0485d82/Modul%206/PRAK601>