

**LAPORAN PRAKTIKUM
PEMROGRAMAN II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Siti Ainur Rahmawati NIM. 2210817220029

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2024**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN 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 : Siti Ainur Rahmawati
NIM : 2210817220029

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	6
B. Output Program.....	28
C. Pembahasan.....	30
Tautan Git	39

DAFTAR GAMBAR

Gambar 1. Screenshot Output Soal 1 Home.php.....	28
Gambar 2. Screenshot Output Soal 1 Register.php	28
Gambar 3. Screenshot Output Soal 1 Login.php	29
Gambar 4. Screenshot Output Soal 1 Dashboard.php	29
Gambar 5. Screenshot Output Soal 1 FormBuku.php	29

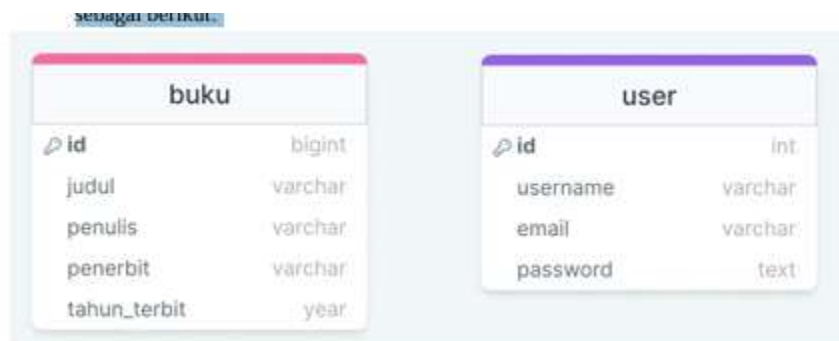
DAFTAR TABEL

Tabel 1. Source Code Soal 1Route.php.....	6
Tabel 2. Source Code Soal 1 Dashboard.php	7
Tabel 3. Source Code Soal 1 FormBukuController.php.....	8
Tabel 4. Source Code Soal 1 Home.php.....	10
Tabel 5. Source Code Soal 1 LoginController.php	11
Tabel 6. Source Code Soal 1 LogoutController.php	13
Tabel 7. Source Code Soal 1 RegisterController.php.....	13
Tabel 8. Source Code Soal 1 AuthFilter.php.....	14
Tabel 9. Source Code Soal 1 BookModel.php	15
Tabel 10. Source Code Soal 1 UserModel.php	16
Tabel 11. Source Code Soal 1 main.php	18
Tabel 12. Source Code Soal 1 1dashboard.php.....	20
Tabel 13. Source Code Soal 1 FormBuku.php	21
Tabel 14. Source Code Soal 1 home.php.....	22
Tabel 15. Source Code Soal 1 login.php	23
Tabel 16. Source Code Soal 1 register.php	24
Tabel 17. Source Code Soal 1 User.php.....	25

SOAL 1

Buatlah sebuah website yang memiliki fitur login dan logout. Setelah login pengguna website dapat melakukan CRUD yang terhubung langsung dengan database. Beberapa syarat yang harus ada pada website:

- Jika seorang pengguna belum login dan mengakses halaman yang mengharuskan login, maka tampilkan sebuah pesan peringatan di halaman login, peringatan berisi “Login terlebih dahulu!”.
- Berikan validasi pada bagian ‘Create’ atau ‘tambah data’ di form, sebagai berikut:
 - Judul : Harus diisi dan berupa string
 - penulis : Harus diisi dan berupa string
 - penerbit : Harus diisi dan berupa string
 - tahun_terbit : Harus diisi dan berupa angka, angka harus lebih besar dari 1800 dan lebih kecil dari 2024
- (opsional) buatlah custom pesan validasi menggunakan bahasa indonesia
- Buat file migration pada codeigniter dengan Desain konseptual database sebagai berikut:



A. Source Code

Tabel 1. Source Code Soal 1Route.php

1	<?php
2	\$routes->setDefaultController('Home::index');
3	\$routes->group('login', function (\$routes) {
4	\$routes->get('/', 'LoginController::index');
5	\$routes->post('/', 'LoginController::login');
6	\$routes->post('/', 'LoginController::logout');

```

7  });
8  $routes->group('dashboard', function ($routes) {
9      $routes->get('/', 'Dashboard::index', ['filter' =>
10         'auth']);
11     $routes->get('delete/(:num)',
12         'Dashboard::delete/$1', ['filter' => 'auth']);
13     $routes->get('edit/(:num)',
14         'FormBukuController::index/$1', ['filter' => 'auth']);
15 });
16 $routes->get('/', 'RegisterController::index');
17 $routes->group('register', function($routes){
18     $routes->get('/', 'RegisterController::register');
19     $routes->post('/', 'RegisterController::store');
20 });
21 $routes->group('logout', function ($routes) {
22     $routes->get('/', 'LogoutController::index',
23         ['filter' => 'auth']);
24 });
25 $routes->group('FormBuku', function($routes){
26     $routes->get('edit/(:num)',
27         'FormBukuController::index/$1', ['filter' => 'auth']);
28     $routes->get('/', 'FormBukuController::index',
29         ['filter' => 'auth']);
30     $routes->post('store', 'FormBukuController::store');
31 });

```

Tabel 2. Source Code Soal 1 Dashboard.php

```

1  <?php
2  namespace App\Controllers;
3  use App\Models\BookModel;
4  use App\Controllers\BaseController;
5
6  class Dashboard extends BaseController
7  {
8      public function index()
9      {
10         if (! $this->isLoggedIn()) {
11             return redirect()->to('/');
12         }
13         $model = new BookModel();

```

13	\$data = [
14	'title' => 'Dashboard',
15	'buku' => \$model->findAll()
16];
17	return view('dashboard', \$data);
18	}
19	private function isLoggedIn() : bool
20	{
21	if (session()->get('logged_in')) {
22	return true;
23	}
24	return false;
25	}
26	public function delete(\$id)
27	{
28	\$model = new BookModel();
29	\$buku = \$model->find(\$id);
30	if (\$buku === null) {
31	throw new
	\CodeIgniter\Exceptions\PageNotFoundException('Book with
	ID ' . \$id . ' not found');
32	}
33	\$model->delete(\$id);
34	return redirect()->to('/dashboard');
35	}
36	}

Tabel 3. Source Code Soal 1 FormBukuController.php

1	<?php
2	namespace App\Controllers;
3	use App\Models\BookModel;
4	use App\Controllers\BaseController;
5	class FormBukuController extends BaseController


```

6  {
7      protected $model;

8      public function __construct()
9      {
10         $this->model = new BookModel();
11         helper(['form', 'url', 'session']);
12     }
13     public function index($id_buku = null)
14     {
15         $model = new BookModel();
16         $data = [
17             'title' => 'Formulir Buku',
18             'buku'  => null
19         ];

20         if ($id_buku) {
21             $buku = $model->find($id_buku);

22             if ($buku === null) {
23                 throw new
24                     \CodeIgniter\Exceptions\PageNotFoundException('Book with
25                     ID ' . $id_buku . ' not found');
26             }

27             $data = [
28                 'title' => 'Edit Buku dengan ID ' .
29                 $id_buku,
30                 'buku' => $buku
31             ];
32         }

33         return view('FormBuku', $data);
34     }
35     public function store()
36     {
37         $id = $this->request->getPost('id_buku');
38         $buku = [
39             'judul_buku' => $this->request-
40             >getPost('judul'),

```

37	<pre> 'penulis' => \$this->request- >getPost('penulis'), </pre>
38	<pre> 'penerbit' => \$this->request- >getPost('penerbit'), </pre>
39	<pre> 'tahun_terbit' => \$this->request- >getPost('tahun') </pre>
40	<pre>]; </pre>
41	<pre> if (\$id) { </pre>
42	<pre> \$this->model->update(\$id, \$buku); </pre>
43	<pre> } else { </pre>
44	<pre> \$this->model->insert(\$buku); </pre>
45	<pre> } </pre>
46	<pre> if (\$this->model->save(\$buku) === false) { </pre>
47	<pre> return redirect()->back()->with('errors', \$this->model->errors())->withInput(); </pre>
48	<pre> } </pre>
49	<pre> session()->setFlashdata('success', 'Buku Telah Berhasil Ditambahkan'); </pre>
50	<pre> return redirect()->to('/dashboard'); </pre>
51	<pre> } </pre>
52	<pre> } </pre>

Tabel 4. Source Code Soal 1 Home.php

1	<?php
2	namespace App\Controllers;
3	use CodeIgniter\Controller;
4	class Home extends Controller
5	{
6	public function index()
7	{
8	return view('home');
9	}
10	}

Tabel 5. Source Code Soal 1 LoginController.php

1	<?php
2	namespace App\Controllers;
3	use App\Controllers\BaseController;
4	use App\Models\UserModel;
5	class LoginController extends BaseController
6	{
7	protected \$model;
8	public function __construct()
9	{
10	\$this->model = new UserModel();
11	\$this->helpers = ['form', 'url'];
12	}
13	public function index()
14	{
15	if (\$this->isLoggedIn()) {
16	return redirect()->to(base_url('dashboard'));
17	}
18	\$data = [
19	'title' => 'Login'
20];
21	return view('login', \$data);
22	}
23	public function login()
24	{
25	\$data = \$this->request->getPost(['email', 'password']);
26	if (! \$this->validateData(\$data, [
27	'email' => 'required valid_email',
28	'password' => 'required min_length[8]'
29])) {
30	return \$this->index();
31	}
32	\$email = \$this->request->getPost('email');

```
33     $password = $this->request->getPost('password');
34     $credentials = ['email' => $email];
35     $user = $this->model->where($credentials)
        ->first();

36     if (! $user) {
37         session()->setFlashdata('error', 'Email atau
password anda salah.');
```

38 return redirect()->back();
39 }

```
40     $passwordCheck = password_verify($password,
$user['password']);

41     if (! $passwordCheck) {
42         session()->setFlashdata('error', 'Email atau
password anda salah.');
```

43 return redirect()->back();
44 }

```
45     $userData = [
46         'username' => $user['username'],
47         'email' => $user['email'],
48         'logged_in' => TRUE
49     ];

50     session()->set($userData);
51     return redirect()->to(base_url('/dashboard'));
52 }
```

```
53 private function isLoggedIn(): bool
54 {
55     if (session()->get('logged_in')) {
56         return true;
57     }

58     return false;
59 }
60 }
```

Tabel 6. Source Code Soal 1 LogoutController.php

1	<?php
2	namespace App\Controllers;
3	use App\Controllers\BaseController;
4	class LogoutController extends BaseController
5	{
6	public function index()
7	{
8	\$session = session();
9	\$session->destroy();
10	return redirect()->to('/login');
11	}
12	}

Tabel 7. Source Code Soal 1 RegisterController.php

1	<?php namespace App\Controllers;
2	use App\Models\UserModel;
3	class RegisterController extends BaseController
4	{
5	protected \$model;
6	public function __construct()
7	{
8	\$this->model = new UserModel();
9	helper(['form', 'url', 'session']);
10	}
11	public function index(): string
12	{
13	\$data = [
14	'title' => 'Beranda'
15];
16	return view('home', \$data);
17	}
18	public function Register(): string
19	{
20	\$data = [
21	'title' => 'Register'
22];

23	return view('register', \$data);
24	}
25	public function store(){
26	\$data = [
27	'username' => \$this->request-
	>getPost('username'),
28	'email' => \$this->request-
	>getPost('email'),
29	'password' => \$this->request-
	>getPost('password'),
30	'password_confirm' => \$this->request-
	>getPost('password_confirm')
31];
32	if (\$this->model->insert(\$data) === false) {
33	return redirect()->back()->with('error',
	\$this->model->errors())->withInput();
34	}
35	session()->setFlashdata('success', 'Registrasi
	berhasil! Silakan login.');
36	return redirect()->to('/register');
37	}
38	}

Tabel 8. Source Code Soal 1 AuthFilter.php

1	<?php
2	namespace App\Filters;
3	use CodeIgniter\Filters\FilterInterface;
4	use CodeIgniter\HTTP\RequestInterface;
5	use CodeIgniter\HTTP\ResponseInterface;
6	class AuthFilter implements FilterInterface{
7	public function before(RequestInterface \$request,
	\$arguments = null)
8	{
9	if (!session()->get('logged_in')) {
10	session()->setFlashdata('error', 'Login
	terlebih dahulu!');
11	return redirect()->to(base_url('login'));
12	}

13	return null;
14	}
15	public function after(RequestInterface \$request,
16	ResponseInterface \$response, \$arguments = null)
17	{
18	}
19	}

Tabel 9. Source Code Soal 1 BookModel.php

1	<?php
2	namespace App\Models;
3	use CodeIgniter\Database\Migration;
4	use CodeIgniter\Model;
5	class BookModel extends Model{
6	protected \$table = 'buku';
7	protected \$primaryKey = 'id_buku';
8	protected \$useAutoIncrement = true;
9	protected \$returnType = 'array';
10	protected \$useSoftDeletes = false;
11	protected \$protectFields = true;
12	protected \$allowedFields = ['judul_buku', 'penulis', 'penerbit', 'tahun_terbit'];
13	protected bool \$allowEmptyInserts = false;
14	protected bool \$updateOnlyChanged = true;
15	protected \$skipValidation = false;
16	protected \$cleanValidationRules = true;
17	protected \$validationRules = [
18	'judul_buku' => 'required string',
19	'penulis' => 'required string',
20	'penerbit' => 'required string',
21	'tahun_terbit' =>
22	'required numeric greater_than[1800] less_than[2024]'
];

```

23     protected $validationMessages = [
24         'judul_buku' => [
25             'required' => 'Judul buku harus diisi.',
26             'string' => 'Judul buku harus berupa string.'
27         ],
28         'penulis' => [
29             'required' => 'Penulis harus diisi.',
30             'string' => 'Penulis harus berupa string.'
31         ],
32         'penerbit' => [
33             'required' => 'Penerbit harus diisi.',
34             'string' => 'Penerbit harus berupa string.'
35         ],
36         'tahun_terbit' => [
37             'required' => 'Tahun terbit harus diisi.',
38             'numeric' => 'Tahun terbit harus berupa
angka.',
39             'greater_than' => 'Tahun terbit harus lebih
besar dari 1800.',
40             'less_than' => 'Tahun terbit harus kurang dari
2024.'
41         ]
42     ];
43 }

```

Tabel 10. Source Code Soal 1 UserModel.php

```

1  <?php
2  namespace App\Models;
3  use CodeIgniter\Model;
4  class UserModel extends Model{
5      protected $table = 'user';
6      protected $allowedFields = ['username', 'email',
'password'];
7      protected $validationRules = [
8          'username' => 'required',
9          'email' =>
'required|valid_email|is_unique[user.email]',
10         'password' => 'required|min_length[8]',
11         'password_confirm' => 'required|matches[password]'

```



```

12 ];
13     protected $validationMessages = [
14         'email' => [
15             'is_unique' => 'Maaf, Email tersebut telah
16 terdaftar. Tolong gunakan email lainnya.'
17         ],
18         'password' => [
19             'min_length' => 'Password harus memiliki
20 panjang minimal 8 karakter.'
21         ],
22         'password_confirm' => [
23             'matches' => 'Password tidak sesuai'
24         ]
25     ];
26
27     protected $skipValidation = false;
28     protected $beforeInsert = ['hashPassword'];
29     protected function hashPassword(array $data)
30     {
31         if (! isset($data['data']['password'])) {
32             return $data;
33         }
34         $data['data']['password'] =
35         password_hash($data['data']['password'],
36             PASSWORD_DEFAULT);
37         return $data;
38     }
39
40     public function checkLogin($email, $password)
41     {
42         $user = $this->where('email', $email)->first();
43         if (!$user) {
44             return false;
45         }
46         if (!password_verify($password,
47             $user['password'])) {
48             return false;
49         }
50         return $user;
51     }
52 }

```

Tabel 11. Source Code Soal 1 main.php

1	<!DOCTYPE html>
2	<html lang="en">
3	<head>
4	<meta charset="UTF-8">
5	<meta name="viewport" content="width=device-width, initial-scale=1.0">
6	<meta http-equiv="X-UA-Compatible" content="ie=edge">
7	<meta name="<?= csrf_token() ?>" content="<?= csrf_hash() ?>">
8	<title><?= \$title; ?></title>
9	<link rel="stylesheet"
10	href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGs03+HhXv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
11	<link rel="stylesheet" href="<?= base_url('css/style.css') ?>">
12	<style>
13	body {
14	padding-top: 56px;
15	}
16	</style>
17	</head>
18	<body>
19	<!-- Navbar -->
20	<nav class="navbar navbar-expand-lg navbar-light bg-light fixed-top">
21	<a class="navbar-brand" href="<?= base_url('/') ?>">Sistem Informasi Perpustakaan
22	<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
23	
24	</button>
25	<div class="collapse navbar-collapse" id="navbarNav">
26	<ul class="navbar-nav ml-auto">

27	<code><?php if (session()->get('logged_in')) :</code>
	<code>?></code>
28	<code> <li class="nav-item"></code>
29	<code> <a class="nav-link" href="<?=</code>
	<code>base_url('/dashboard') ?>">Dashboard</code>
30	<code> </code>
31	<code> <li class="nav-item"></code>
32	<code> <a class="nav-link" href="<?=</code>
	<code>base_url('/logout') ?>">Logout</code>
33	<code> </code>
34	<code> <?php else : ?></code>
35	<code> <li class="nav-item"></code>
36	<code> <a class="nav-link" href="<?=</code>
	<code>base_url('/login') ?>">Login</code>
37	<code> </code>
38	<code> <li class="nav-item"></code>
39	<code> <a class="nav-link" href="<?=</code>
	<code>base_url('/register') ?>">Register</code>
40	<code> </code>
41	<code> <?php endif; ?></code>
42	<code> </code>
43	<code> </div></code>
44	<code></nav></code>
45	<code><!-- Main Content --></code>
46	<code><div class="container mt-4"></code>
47	<code> <?= \$this->renderSection('content') ?></code>
48	<code></div></code>
49	<code><!-- Footer --></code>
50	<code><footer class="text-center mt-5"></code>
51	<code> <p><small>Perpustakaan@<a</code>
	<code>href="https://example.com/">contoh.com</small></</code>
	<code>p></code>
52	<code></footer></code>
53	<code><script src="https://code.jquery.com/jquery-</code>
	<code>3.4.1.slim.min.js" integrity="sha384-</code>
	<code>J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfo</code>
	<code>RSJoZ+n" crossorigin="anonymous"></script></code>

54	<code><script</code> src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
55	<code><script</code> src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
56	<code></body></code>
57	<code></html></code>

Tabel 12. Source Code Soal 1 1dashboard.php

1	<code><?= \$this->extend('layouts/main') ?></code>
2	<code><?= \$this->section('content') ?></code>
3	<code><a href="<?php echo base_url('FormBuku'); ?>" class="btn btn-outline-primary my-2">Tambah Buku</code>
4	<code><table border="1"></code>
5	<code> <tr></code>
6	<code> <th>ID</th></code>
7	<code> <th>Judul</th></code>
8	<code> <th>Penerbit</th></code>
9	<code> <th>Penulis</th></code>
10	<code> <th>Tahun Terbit</th></code>
11	<code> <th>Aksi</th></code>
12	<code> </tr></code>
13	<code> <?php if (!empty(\$buku) && is_array(\$buku)): ?></code>
14	<code> <?php foreach (\$buku as \$book): ?></code>
15	<code> <tr></code>
16	<code> <td><?= esc(\$book['id_buku']) ?></td></code>
17	<code> <td><?= esc(\$book['judul_buku'])</code>
	<code>?></td></code>
18	<code> <td><?= esc(\$book['penulis'])</code>
	<code>?></td></code>
19	<code> <td><?= esc(\$book['penerbit'])</code>
	<code>?></td></code>
20	<code> <td><?=</code>
	<code>esc(\$book['tahun_terbit']) ?></td></code>

21	<td>
22	<a href="<?= base_url('/dashboard/edit/' .
23	\$book['id_buku']) ?>" class="btn btn-primary">Edit
24	<a href="<?= base_url('/dashboard/delete/'
25	. \$book['id_buku']) ?>" class="btn btn-primary">Hapus
26	</td>
27	</tr>
28	<?php endforeach; ?>
29	<?php else : ?>
30	<tr>
31	<td colspan="6">Tidak ada buku yang
32	terdaftar</td>
33	</tr>
	<?php endif; ?>
	</table>
	<?= \$this->endSection() ?>

Tabel 13. Source Code Soal 1 FormBuku.php

1	<?= \$this->extend('layouts/main') ?>
2	<?= \$this->section('content') ?>
3	<h1>Form Buku</h1>
4	<form method="post" action="<?=
5	base_url('/FormBuku/store') ?>">
6	<?php if (session()->getFlashdata('errors')) : ?>
7	<div class="alert alert-danger">
8	<?php foreach (session()-
9	>getFlashdata('errors') as \$error) : ?>
10	<p><?= \$error ?></p>
11	<?php endforeach; ?>
12	</div>
13	<?php endif; ?>
14	<input type="hidden" name="id_buku" value="<?=
15	isset(\$buku['id_buku']) ?
	htmlspecialchars(\$buku['id_buku']) : '' ?>">
	<label for="judul">Judul</label>
	<input type="text" name="judul" id="judul" max="500"
	required value="<?= isset(\$buku['judul_buku']) ?
	htmlspecialchars(\$buku['judul_buku']) : '' ?>">
	<label for="penulis">Penulis</label>

16	<code><input type="text" name="penulis" id="penulis" max="500" required value="<?= isset(\$buku['penulis']) ? htmlspecialchars(\$buku['penulis']) : ' ' ?>">
</code>
17	<code><label for="penerbit">Penerbit</label></code>
18	<code><input type="text" name="penerbit" id="penerbit" max="250" required value="<?= isset(\$buku['penerbit']) ? htmlspecialchars(\$buku['penerbit']) : ' ' ?>">
</code>
19	<code><label for="tahun">Tahun Terbit</label></code>
20	<code><input type="number" name="tahun" id="tahun" required value="<?= isset(\$buku['tahun_terbit']) ? htmlspecialchars(\$buku['tahun_terbit']) : ' ' ?>">
</code>
21	<code><button type="submit" name="submit">Simpan</button></code>
22	<code></form></code>
23	<code><?= \$this->endSection() ?></code>

Tabel 14. Source Code Soal 1 home.php

1	<code><?= \$this->extend('layouts/main') ?></code>
2	<code><?= \$this->section('content') ?></code>
3	<code><section class="jumbotron text-center" style="height:100vh"></code>
4	<code><h1 class="mt-5">PERPUSTAKAAN ATLANTIS</h1></code>
5	<code><p class="lead text-muted"></code>
6	<code>Wujudkan Kota Impian penuh Keajaiban dari Seluruh Dunia dan Dimensi di Kota Atlantis</code>
	<code>Pusat Portal Tata Surya</code>
7	<code></p></code>
8	<code><a href="<?php echo base_url('login'); ?>" class="btn btn-outline-primary my-2">Login</code>
9	<code><a href="<?php echo base_url('register'); ?>" class="btn btn-success my-2">Register
</code>
10	<code></code>
11	<code></section></code>
12	<code><?= \$this->endSection() ?></code>

Tabel 15. Source Code Soal 1 login.php

1	<?= \$this->extend('layouts/main') ?>
2	<?= \$this->section('content') ?>
3	<div class="container mt-5">
4	<div class="row">
5	<div class="col-md-4 offset-md-4">
6	<div class="card">
7	<div class="card-body">
8	<h4 class="text-center" style="font-weight: bold;">LOGIN</h4>
9	<hr>
10	<?php if (session()->getFlashdata('error')) : ?>
11	<div class="alert alert-danger">
12	<?php echo session()->getFlashdata('error'); ?>
13	</div>
14	<?php endif; ?>
15	<?= form_open('login'); ?>
16	<div class="form-group">
17	<label for="email">Email</label>
18	<input type="email" name="email" class="form-control" required>
19	</div>
20	<div class="form-group">
21	<label
22	for="password">Password</label>
23	<input type="password" name="password" class="form-control" required>
24	</div>
25	<div class="form-group">
26	<button class="btn btn-primary">Login</button>
27	</div>
28	<?= form_close(); ?>
29	</div>

30	<code><div class="text-center mt-2"></code>
31	<code> Belum punya akun? <a href="<?php echo</code></code>
32	<code>base_url('register'); ?>">Silakan daftar.</code>
33	<code></div></code>
34	<code></div></code>
35	<code></div></code>
36	<code><?= \$this->endSection() ?></code>

Tabel 16. Source Code Soal 1 register.php

1	<code><?= \$this->extend('layouts/main.php') ?></code>
2	<code><?= \$this->section('content') ?></code>
3	<code><div class="row justify-content-center"></code>
4	<code> <div class="col-md-6"></code>
5	<code> <h4 class="text-center" style="font-weight:</code>
6	<code>bold;">REGISTER</h4></code>
7	<code> <hr></code>
8	<code> <?php if (session()->getFlashdata('error')) : ?></code>
9	<code> <div class="alert alert-danger"></code>
10	<code> <?php foreach (session()-</code>
11	<code>>getFlashdata('error') as \$error) : ?></code>
12	<code> <p><?= esc(\$error) ?></p></code>
13	<code> <?php endforeach ?></code>
14	<code> </div></code>
15	<code> <?php endif; ?></code>
16	<code> <?= validation_list_errors() ?></code>
17	<code> <?= form_open('register'); ?></code>
18	<code> <div class="form-group"></code>
19	<code> <label for="username">Nama Pengguna</label></code>
20	<code> <input type="text" name="username"</code>
21	<code>class="form-control" required></code>
22	<code> </div></code>
23	<code> <div class="form-group"></code>
24	<code> <label for="email">Email</label></code>
25	<code> <input type="email" name="email" class="form-</code>
26	<code>control" required></code>
27	<code> </div></code>

24	<div class="form-group">
25	<label for="password">Password</label>
26	<input type="password" name="password"
	class="form-control" required>
27	</div>
28	<div class="form-group">
29	<label for="password_confirm">Konfirmasi
	Password</label>
30	<input type="password"
	name="password_confirm" class="form-control"
	id="password_confirm" required>
31	</div>
32	<div class="form-group text-center">
33	<button class="btn btn-
	primary">Register</button>
34	</div>
35	<?= form_close(); ?>
36	</div>
37	</div>
38	<?php if (session()->getFlashdata('success')) : ?>
39	<script>
40	alert('<?= session()->getFlashdata('success') ?>');
41	window.location.href = '<?= base_url('/login') ?>';
42	</script>
43	<?php endif; ?>
44	<?= \$this->endSection() ?>

Tabel 17. Source Code Soal 1 User.php

1	<?php
2	namespace App\Database\Migrations;
3	use CodeIgniter\Database\Migration;
4	class User extends Migration
5	{
6	public function up()
7	{
8	\$this->forge->addField([
9	'id_user' => [

```

10         'type' => 'INT',
11         'unsigned' => TRUE,
12         'auto_increment' => TRUE
13     ],
14     'username' => [
15         'type' => 'VARCHAR',
16         'constraint' => 25,
17         'null' => FALSE,
18     ],
19     'email' => [
20         'type' => 'VARCHAR',
21         'constraint' => 30,
22         'null' => FALSE,
23         'unique' => TRUE,
24     ],
25     'password' => [
26         'type' => 'TEXT',
27         'constraint' => 255,
28         'null' => FALSE,
29     ]
30 ];

31 $this->forge->addKey('id_user', TRUE);
32 $this->forge->createTable('user', true);
33 }
34 public function down()
35 {
36     $this->forge->dropTable('user');
37 }
38 }
39 ?>

```

```

1  <?php
2  namespace App\Database\Migrations;
3  use CodeIgniter\Database\Migration;

4  class Buku extends Migration
5  {
6      public function up()
7      {

```

```
8         $this->forge->addField([
9             'id_buku' => [
10                 'type' => 'INT',
11                 'unsigned' => TRUE,
12                 'auto_increment' => TRUE,
13                 'unique' => TRUE,
14             ],
15             'judul_buku' => [
16                 'type' => 'VARCHAR',
17                 'constraint' => 500,
18                 'null' => FALSE,
19             ],
20             'penulis' => [
21                 'type' => 'VARCHAR',
22                 'constraint' => 500,
23                 'null' => FALSE,
24             ],
25             'penerbit' => [
26                 'type' => 'VARCHAR',
27                 'constraint' => 250,
28                 'null' => FALSE,
29             ],
30             'tahun_terbit' => [
31                 'type' => 'INT',
32                 'constraint' => 11,
33                 'null' => FALSE,
34             ]
35         ]);

36         $this->forge->addKey('id_buku', TRUE);
37         $this->forge->createTable('buku', true);
38     }
39     public function down()
40     {
41         $this->forge->dropTable('buku');
42     }
43 }
```

B. Output Program



Gambar 1. Screenshot Output Soal 1 Home.php

The screenshot shows the "REGISTER" page of the library system. The header is the same as the home page. The main content area has a title "REGISTER" and a registration form with fields for "Nama Pengguna", "Email", "Password", and "Konfirmasi Password", followed by a "Register" button. The footer includes the email "Perpustakaan@sekolah.com" and a small flame icon.

Gambar 2. Screenshot Output Soal 1 Register.php

Gambar 3. Screenshot Output Soal 1 Login.php

ID	Author	Publisher	Periodic	Volume	Tahun Terbit	Aksi
1	Ayem	Shinta	Tamara	2000		Edit Hapus
5	Kucing	Shinta	Tamara	2000		Edit Hapus
6	HEHE	Shinta	Tamara	2001		Edit Hapus

Gambar 4. Screenshot Output Soal 1 Dashboard.php

Gambar 5. Screenshot Output Soal 1 FormBuku.php

C. Pembahasan

Route.php

Pada baris [2] Mengatur `Home::index` sebagai controller default yang akan dipanggil saat mengakses root URL (`http://localhost/`).

Pada baris [3] – [7] Mengatur route untuk halaman login

Pada baris [8] – [12] Mengatur route untuk halaman dashboard

Pada baris [13] Mengatur route root (`http://localhost/`) untuk menampilkan halaman registrasi dengan `RegisterController::index`.

Pada baris [14] – [17] Mengatur route untuk halaman register

Pada baris [18] – [20] Mengatur route untuk halaman logout

Pada baris [21] – [25] Mengatur route untuk halaman FormBuku

Dashboard.php

Pada baris [1] Namespace Menunjukkan bahwa controller ini berada dalam namespace `App\Controllers`.

Pada baris [2] – [4] Use Statements Mengimpor `BookModel` untuk mengakses data buku dan `BaseController` sebagai basis controller.

Pada baris [5] – [36] Mendefinisikan class `Dashboard` yang mewarisi `BaseController`.

Pada baris [8] – [12] Memanggil Constructor Kelas Induk `parent::__construct()`; digunakan untuk memastikan constructor dari kelas induk (`BaseController`) juga dipanggil. Ini penting jika ada inisialisasi di kelas induk yang harus dilakukan.

Pada baris [13] – [18] Mendefinisikan Fungsi yang bertujuan untuk menampilkan halaman dashboard kepada pengguna yang sudah login.

Pada baris [19] – [25] Fungsi `isLoggedIn()` dalam kelas `Dashboard` adalah fungsi privat yang digunakan untuk memeriksa apakah pengguna saat ini sudah login atau belum. Fungsi ini mengembalikan nilai boolean (`true` atau `false`).

Pada baris [26] – [35] Fungsi `delete` dalam kelas `Dashboard` digunakan untuk menghapus data buku dari database berdasarkan ID buku yang diberikan.

FormBukuController.php

Pada baris [2] Namespace Menunjukkan bahwa controller ini berada dalam namespace `App\Controllers`.

Pada baris [3] – [4] Use Statements Mengimpor `BookModel` untuk mengakses data buku dan `BaseController` sebagai basis controller.

Pada baris [5] – [52] Mendefinisikan class `FormBukuController` yang mewarisi `BaseController`.

Pada baris [8] – [12] Memanggil Constructor Kelas Induk `parent::__construct()`; digunakan untuk memastikan constructor dari kelas induk (`BaseController`) juga dipanggil. Ini penting jika ada inisialisasi di kelas induk yang harus dilakukan.

Pada baris [13] – [31] Fungsi `index` dalam `FormBukuController` digunakan untuk menampilkan formulir buku. Formulir ini bisa digunakan untuk menambahkan buku baru atau mengedit buku yang sudah ada, tergantung pada apakah ID buku diberikan atau tidak. Pada baris [32] – [52] Fungsi `store` digunakan untuk menangani pengiriman data formulir untuk menambahkan atau mengedit buku

Home.php

Pada baris [2] Namespace Menunjukkan bahwa controller ini berada dalam namespace `App\Controllers`.

Pada baris [3] Use Statements Mengimpor `BookModel` untuk mengakses data buku dan `BaseController` sebagai basis controller.

Pada baris [4] – [10] Mendefinisikan class `Home` yang mewarisi `Controller`.

Pada baris [6] – [9] `index()` adalah metode yang akan dijalankan ketika pengguna mengakses URL yang mengarah ke controller `Home` tanpa metode spesifik.

LoginController.php

Pada baris [2] Namespace Menunjukkan bahwa controller ini berada dalam namespace `App\Controllers`.

Pada baris [3] – [4] Use Statements Mengimpor `BookModel` untuk mengakses data buku dan `BaseController` sebagai basis controller.

Pada baris [5] – [60] Mendefinisikan class `LoginController` yang mewarisi `Controller`.

Pada baris [8] – [12] Mendefinisikan konstruktor untuk sebuah kelas

Pada baris [13] – [22] Mendefinisakan fungsi `index()` yang digunakan untuk mengecek apakah pengguna sudah login. Jika iya, mengarahkan pengguna ke dashboard, menyiapkan data untuk view, dalam hal ini hanya judul halaman, dan menampilkan view login kepada pengguna.

Pada baris [23] – [52] Mendefinisikan Fungsi `login()` yang bertanggung jawab untuk menangani proses login pengguna.

Pada baris [53] – [59] Mendefinisikan Fungsi `isLoggedIn()` yang bertujuan untuk memeriksa apakah pengguna saat ini telah masuk ke dalam sistem.

LogoutController.php

Pada baris [2] Namespace Menunjukkan bahwa controller ini berada dalam namespace `App\Controllers`.

Pada baris [3] Use Statements Mengimpor `BookModel` untuk mengakses data buku dan `BaseController` sebagai basis controller.

Pada baris [4] – [12] Mendefinisikan class `LogoutController` yang mewarisi `BaseController`.

Pada baris [6] – [11] Mendefinisikan fungsi `index()` yang bertujuan untuk mengakhiri sesi pengguna dan mengarahkan pengguna kembali ke halaman login.

RegisterController.php

Pada baris [1] Namespace Menunjukkan bahwa controller ini berada dalam namespace `App\Controllers`.

Pada baris [2] Use Statements Mengimpor `UserModel` untuk mengakses data buku dan `BaseController` sebagai basis controller.

Pada baris [3] – [38] Mendefinisikan class `RegisterController` yang mewarisi `Controller`.

Pada baris [6] – [10] Mendefinisikan konstruktor untuk sebuah kelas

Pada baris [11] – [17] Mendefinisikan Fungsi `index()` yang Anda bagikan bertujuan untuk mengakhiri sesi pengguna dan mengarahkan pengguna kembali ke halaman home

Pada baris [18] – [24] Mendefinisikan Fungsi `Register()` ini sangat sederhana dan bertujuan untuk menampilkan halaman registrasi kepada pengguna dengan judul yang sesuai

Pada baris [25] – [37] Mendefinisikan Fungsi `store()` yang Anda bagikan bertujuan untuk menangani proses registrasi pengguna.

AuthFilter.php

Pada baris [2] Namespace Menunjukkan bahwa controller ini berada dalam namespace `App\Controllers`.

Pada baris [3] – [5] Mengimpor antarmuka `RequestInterface`, `ResponseInterface`, dan `FilterInterface` dari CodeIgniter yang diperlukan untuk membuat filter

Pada baris [6] – [18] Mendeklarasikan kelas `AuthFilter` yang mengimplementasikan `FilterInterface`

Pada baris [7] – [14] Mendefinisikan Method `before` akan dijalankan sebelum request diproses oleh controller.

Pada baris [15] – [17] Mendefinisikan Method `after` akan dijalankan setelah request diproses oleh controller.

BookModel.php

Pada baris [2] Namespace Menunjukkan bahwa controller ini berada dalam namespace `App\Models`.

Pada baris [3] – [4] Mengimpor antarmuka `RequestInterface`, `ResponseInterface`, dan `FilterInterface` dari CodeIgniter yang diperlukan untuk membuat filter

Pada baris [5] – [43] Mendefinisikan kelas `BookModel` yang meng-extend `Model` dari CodeIgniter, yang menyediakan fungsionalitas ORM (Object-Relational Mapping) untuk berinteraksi dengan database

Pada baris [6] `$table`: Nama tabel dalam database

Pada baris [7] `$primaryKey`: Primary key dari tabel

Pada baris [8] `$useAutoIncrement`: Mengatur apakah primary key menggunakan auto increment.

Pada baris [9] `$returnType`: Jenis data yang dikembalikan oleh model

Pada baris [10] `$useSoftDeletes`: Mengatur apakah menggunakan soft delete atau tidak.

Pada baris [11] `$protectFields`: Mengatur apakah field harus dilindungi dari mass assignment

Pada baris [12] `$allowedFields`: Field yang diizinkan untuk diisi melalui mass assignment

Pada baris [13] `$allowEmptyInserts`: Mengatur apakah mengizinkan insert dengan data kosong

Pada baris [14] `$updateOnlyChanged`: Mengatur apakah hanya mengupdate field yang berubah saja.

Pada baris [15] `$skipValidation`: Mengatur apakah skip validasi atau tidak

Pada baris [16] `$cleanValidationRules`: Mengatur apakah membersihkan aturan validasi atau tidak

Pada baris [17] – [22] `$validationRules`: Aturan validasi untuk field-field dalam tabel

Pada baris [23] – [42] `$validationMessages`: Pesan yang ditampilkan jika validasi gagal.

UserModel.php

Pada baris [2] Namespace Menunjukkan bahwa controller ini berada dalam namespace `App\Models`.

Pada baris [3] Mengimpor antarmuka `RequestInterface`, `ResponseInterface`, dan `FilterInterface` dari CodeIgniter yang diperlukan untuk membuat filter

Pada baris [4] – [46] Mendefinisikan kelas `BookModel` yang meng-extend `Model` dari CodeIgniter, yang menyediakan fungsionalitas ORM (Object-Relational Mapping) untuk berinteraksi dengan database

Pada baris [5] `$table`: Nama tabel dalam database.

Pada baris [6] `$allowedFields`: Field yang diizinkan untuk diisi melalui mass assignment.

Pada baris [7] – [12] `$validationRules`: Aturan validasi untuk field-field dalam tabel

Pada baris [13] – [23] `$validationMessages`: Pesan yang ditampilkan jika validasi gagal

Pada baris [24] `$skipValidation`: Mengatur apakah skip validasi atau tidak

Pada baris [25] `$beforeInsert`: Callback yang akan dipanggil sebelum data disimpan ke dalam database.

Pada baris [26] – [33] Mendefinisikan `hashPassword`: Fungsi untuk meng-hash password sebelum disimpan ke dalam database.

Pada baris [34] – [44] `checkLogin`: Fungsi untuk memeriksa kredensial login

Main.php

Pada baris [1] – [17] Mendefinisikan Tag HTML dan Head

Pada baris [19] – [44] Mendefinisikan Navbar yang Menggunakan Bootstrap untuk membuat navbar yang responsif dan fixed-top.

Pada baris [45] – [48] Mendefinisikan Main Content yang diisi dengan konten dinamis menggunakan `renderSection` dari CodeIgniter

Pada baris [49] – [52] Mendefinisikan Footer yang Menampilkan informasi footer dengan teks yang dipusatkan

Pada baris [53] Library jQuery JavaScript yang digunakan oleh Bootstrap untuk fungsionalitas interaktif.

Pada baris [54] Library Popper.js untuk penanganan popper yang digunakan oleh Bootstrap untuk tooltip dan popover.

Pada baris [55] Menambahkan fungsionalitas interaktif Bootstrap seperti modal, dropdown, dll

dashboard.php

Pada baris [1] Bagian ini menunjukkan bahwa tampilan ini adalah bagian dari layout utama yang disebut `layouts/main`.

Pada baris [2] Bagian ini mendefinisikan awal dari `content` section yang akan diisi oleh konten yang spesifik untuk halaman ini.

Pada baris [3] Menampilkan tombol untuk menambah buku baru.

Pada baris [4] – [32] Membuat tabel HTML dengan border untuk menampilkan daftar buku.

Pada baris [33] Menandai akhir dari `content` section. Semua konten di antara `section('content')` dan `endSection()` akan dimasukkan ke dalam `content` section dari layout utama (`layouts/main`).

FormBuku.php

Pada baris [1] Bagian ini menunjukkan bahwa tampilan ini adalah bagian dari layout utama yang disebut `layouts/main`.

Pada baris [2] Bagian ini mendefinisikan awal dari `content` section yang akan diisi oleh konten yang spesifik untuk halaman ini.

Pada baris [3] Menampilkan judul halaman yaitu "Form Buku"

Pada baris [4] – [22] Membuat form HTML dengan metode POST yang mengirimkan data ke URL `base_url('/FormBuku/store')` saat form disubmit

Pada baris [23] Menandai akhir dari `content` section. Semua konten di antara `section('content')` dan `endSection()` akan dimasukkan ke dalam `content` section dari layout utama (`layouts/main`).

home.php

Pada baris [1] Bagian ini menunjukkan bahwa tampilan ini adalah bagian dari layout utama yang disebut `layouts/main`.

Pada baris [2] Bagian ini mendefinisikan awal dari `content` section yang akan diisi oleh konten yang spesifik untuk halaman ini.

Pada baris [3] – [11] Bagian ini membuat sebuah section dengan kelas `jumbotron` yang merupakan komponen dari Bootstrap untuk menampilkan konten utama dalam sebuah halaman.

Pada baris [12] Menandai akhir dari `content` section. Semua konten di antara `section('content')` dan `endSection()` akan dimasukkan ke dalam `content` section dari layout utama (`layouts/main`).

login.php

Pada baris [1] Bagian ini menunjukkan bahwa tampilan ini adalah bagian dari layout utama yang disebut `layouts/main`.

Pada baris [2] Bagian ini mendefinisikan awal dari `content` section yang akan diisi oleh konten yang spesifik untuk halaman ini.

Pada baris [10] – [14] Mengecek apakah ada pesan error dalam session flashdata dan menampilkannya jika ada. `alert alert-danger` digunakan untuk memberikan gaya tampilan alert berwarna merah dari Bootstrap.

Pada baris [15] - [28] Membuat form dengan method `POST` yang akan mengarah ke URL `login`. `form-group` digunakan untuk mengelompokkan elemen form. `form-control` memberikan gaya pada input sesuai dengan Bootstrap.

Pada baris [30] – [32] Menampilkan teks dengan link ke halaman register untuk pengguna yang belum memiliki akun.

Pada baris [36] Menandai akhir dari `content` section. Semua konten di antara `section('content')` dan `endSection()` akan dimasukkan ke dalam `content` section dari layout utama (`layouts/main`).

Register.php

Pada baris [1] Bagian ini menunjukkan bahwa tampilan ini adalah bagian dari layout utama yang disebut `layouts/main`.

Pada baris [2] Bagian ini mendefinisikan awal dari `content` section yang akan diisi oleh konten yang spesifik untuk halaman ini.

Pada baris [7] – [13] Mengecek apakah ada pesan error dalam session flashdata dan menampilkannya jika ada. `alert alert-danger` digunakan untuk memberikan gaya tampilan alert berwarna merah dari Bootstrap

Pada baris [14] Menampilkan daftar error validasi yang dihasilkan oleh form validation.

Pada baris [15] – [35] Membuat form dengan method `POST` yang akan mengarah ke URL `register`. `form-group` digunakan untuk mengelompokkan elemen form.

Pada baris [38] – [43] Mengecek apakah ada pesan sukses dalam session flashdata dan menampilkannya jika ada. Jika ada pesan sukses, akan muncul alert dengan pesan tersebut, kemudian halaman akan dialihkan ke URL `login`

Pada baris [44] Menandai akhir dari `content` section. Semua konten di antara `section('content')` dan `endSection()` akan dimasukkan ke dalam `content` section dari layout utama (`layouts/main.php`).

User.php

Pada baris [2] – [3] Menggunakan namespace `App\Database\Migrations` dan mengimpor kelas `Migration` dari CodeIgniter.

Pada baris [4] – [36] Mendefinisikan kelas `User` yang meng-extend kelas `Migration`.

Pada baris [6] – [31] Mendefinisikan method `up` yang digunakan untuk membuat tabel `user`.

Pada baris [29] Menambahkan primary key pada kolom `id_user`

Pada baris [30] Membuat tabel `user` dengan mengatur parameter `true` untuk memastikan bahwa tabel tidak akan diganti jika sudah ada

Pada baris [32] – [36] Method `down` digunakan untuk menghapus tabel `user` jika rollback migrasi dilakukan.

Buku.php

Pada baris [2] – [3] Menggunakan namespace `App\Database\Migrations` dan mengimpor kelas `Migration` dari CodeIgniter.

Pada baris [4] – [36] Mendefinisikan kelas `Buku` yang meng-extend kelas `Migration`.

Pada baris [6] – [31] Mendefinisikan method `up` yang digunakan untuk membuat tabel `buku`.

Pada baris [29] Menambahkan primary key pada kolom `id_buku`

Pada baris [30] Membuat tabel `buku` dengan mengatur parameter `true` untuk memastikan bahwa tabel tidak akan diganti jika sudah ada

Pada baris [32] – [36] Method `down` digunakan untuk menghapus tabel `buku` jika rollback migrasi dilakukan.

Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/kemerekehe/Web-II.vsc/tree/main/ci-news>