

**LAPORAN PRAKTIKUM  
PEMROGRAMAN WEB II  
MODUL 6**



**CRUD DAN LOGIN PADA CODEIGNITER**

**Oleh:**

**Aisya Anindia Septha    NIM. 2210817220007**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS LAMBUNG MANGKURAT  
JUNI 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 : Aisya Anindia Septha  
NIM : 2210817220007

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.....	31
C.    Pembahasan .....	34
Tautan Git .....	60

## DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1 Tampilan Home .....	31
Gambar 2. Screenshot Hasil Jawaban Soal 1 Tampilan Login.....	32
Gambar 3. Screenshot Hasil Jawaban Soal 1 Tampilan Register .....	32
Gambar 4. Screenshot Hasil Jawaban Soal 1 Tampilan Dashboard.....	33
Gambar 5. Screenshot Hasil Jawaban Soal 1 Tampilan Create.....	33
Gambar 6. Screenshot Hasil Jawaban Soal 1 Tampilan Edit .....	34
Gambar 7. Screenshot Hasil Jawaban Soal 1 mengakses Dashboard Tanpa Login .....	34

## DAFTAR TABEL

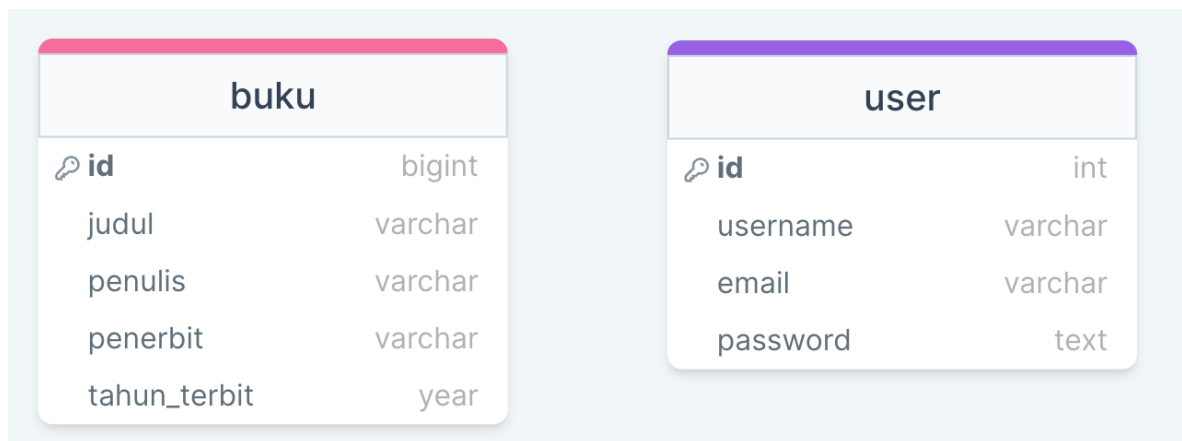
Table 1. Source Code Soal1 Routes.php .....	7
Table 2. Source Code Soal 1 Buku.php.....	7
Table 3. Source Code Soal 1 Login.php.....	13
Table 4. Source Code Soal 1 Home.php.....	15
Table 5. Source Code Soal 1 CreateBukuTable.php .....	16
Table 6. Source Code Soal 1 CreateUserTable.php .....	17
Table 7. Source Code Soal 1 AuthFilters.php .....	18
Table 8. Source Code Soal 1 Buku.Model.php .....	19
Table 9. Source Code Soal 1 UserModel.php .....	19
Table 10. Source Code Soal 1 create.php.....	20
Table 11. Source Code Soal 1 edit.php .....	22
Table 12. Source Code Soal 1 home.php.....	23
Table 13. Source Code Soal 1 index.php.....	26
Table 14. Source Code Soal 1 login.php .....	28
Table 15. Source Code Soal 1 register.php .....	29

## 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:



Simpan dengan nama projek di Github: PRAK601.

## A. Source Code

Table 1. Source Code Soal1 Routes.php

1	<?php
2	
3	use CodeIgniter\Router\RouteCollectionInterface;
4	
5	\$routes->get('/', 'Home::index');
6	\$routes->get('/login', 'Login::loginForm');
7	\$routes->post('/login', 'Login::login');
8	\$routes->get('/register', 'Login::registerForm');
9	\$routes->post('/register', 'Login::register');
10	\$routes->get('/logout', 'Login::logout');
11	\$routes->group('', ['filter' => 'auth'], function
	(\$routes) {
12	\$routes->get('/dashboard', 'Buku::index');
13	\$routes->get('/buku', 'Buku::index');
14	\$routes->get('/buku/create', 'Buku::create');
15	\$routes->post('/buku/store', 'Buku::store');
16	\$routes->get('/buku/edit/(:num)', 'Buku::edit/\$1');
17	\$routes->post('/buku/update/(:num)',
	'Buku::update/\$1');
18	\$routes->get('/buku/delete/(:num)',
	'Buku::delete/\$1');
19	});

Table 2. Source Code Soal 1 Buku.php

1	<?php
2	namespace App\Controllers;
3	use App\Controllers\BaseController;
4	use App\Models\BukuModel;
5	use
6	CodeIgniter\Validation\Exceptions\ValidationException;

```

7 class Buku extends BaseController
8 {
9     protected $bukuModel;
10
11     public function __construct()
12     {
13         $this->bukuModel = new BukuModel();
14     }
15     public function index()
16     {
17         $data['buku'] = $this->bukuModel->findAll();
18         return view('index', $data);
19     }
20     public function create()
21     {
22         return view('create');
23     }
24     public function store()
25     {
26         $validation = \Config\Services::validation();
27         $validation->setRules([
28             'judul' => [
29                 'label' => 'Judul',
30                 'rules' => 'required|string',
31                 'errors' => [
32                     'required' => '{field} harus
diisi.',
33                     'string' => '{field} harus berupa
string.'
34                 ]
35             ],

```



36	'penulis' => [
37	'label' => 'Penulis',
38	'rules' => 'required string',
39	'errors' => [
40	'required' => '{field} harus
41	diisi.',
42	'string' => '{field} harus berupa
43	string.'
44	]
45	],
46	'penerbit' => [
47	'label' => 'Penerbit',
48	'rules' => 'required string',
49	'errors' => [
50	'required' => '{field} harus
51	diisi.',
52	'string' => '{field} harus berupa
53	string.'
54	]
55	],
56	'tahun_terbit' => [
57	'label' => 'Tahun Terbit',
58	'rules' =>
	'required numeric greater_than[1800] less_than[2024]',
	'errors' => [
	'required' => '{field} harus
	diisi.',
	'numeric' => '{field} harus berupa
	angka.',
	'greater_than' => '{field} harus
	lebih besar dari 1800.',

59	<pre> 'less_than' =&gt; '{field} harus kurang dari 2024.'</pre>
60	<pre> ]</pre>
61	<pre> ]</pre>
62	<pre> ]);</pre>
63	<pre> if (!\$validation-&gt;withRequest(\$this-&gt;request)- &gt;run()) {</pre>
64	<pre> return view('create', ['validation' =&gt; \$validation]);</pre>
65	<pre> }</pre>
66	<pre> \$tahunTerbit = \$this-&gt;request- &gt;getPost('tahun_terbit');</pre>
67	<pre> if (\$tahunTerbit &gt; date('Y')) {</pre>
68	<pre>     \$validation-&gt;setError('tahun_terbit', 'Tahun Terbit harus lebih kecil dari tahun saat ini.');</pre>
69	<pre>     throw new ValidationException(\$validation);</pre>
70	<pre> }</pre>
71	<pre> \$data = [</pre>
72	<pre>     'judul' =&gt; \$this-&gt;request- &gt;getPost('judul'),</pre>
73	<pre>     'penulis' =&gt; \$this-&gt;request- &gt;getPost('penulis'),</pre>
74	<pre>     'penerbit' =&gt; \$this-&gt;request- &gt;getPost('penerbit'),</pre>
75	<pre>     'tahun_terbit' =&gt; \$tahunTerbit,</pre>
76	<pre> ];</pre>
77	<pre> \$this-&gt;bukuModel-&gt;save(\$data);</pre>
78	<pre> return redirect()-&gt;to('/dashboard');</pre>
79	<pre> }</pre>
80	<pre> public function edit(\$id)</pre>
81	<pre> {</pre>

82	\$data['buku'] = \$this->bukuModel->find(\$id);
83	if (!\$data['buku']) {
84	throw new
	\CodeIgniter\Exceptions\PageNotFoundException('Buku
	dengan ID ' . \$id . ' tidak ditemukan.');
85	}
86	return view('edit', \$data);
87	}
88	public function update(\$id)
89	{
90	\$validation = \Config\Services::validation();
91	\$validation->setRules([
92	'judul' => [
93	'label' => 'Judul',
94	'rules' => 'required string',
95	'errors' => [
96	'required' => '{field} harus
	diisi.',
97	'string' => '{field} harus berupa
	string.'
98	]
99	],
100	'penulis' => [
101	'label' => 'Penulis',
102	'rules' => 'required string',
103	'errors' => [
104	'required' => '{field} harus
	diisi.',
105	'string' => '{field} harus berupa
	string.'
106	]

```

107         ],
108         'penerbit' => [
109             'label' => 'Penerbit',
110             'rules' => 'required|string',
111             'errors' => [
112                 'required' => '{field} harus
113                 diisi.',
114                 'string' => '{field} harus berupa
115                 string.'
116             ]
117         ],
118         'tahun_terbit' => [
119             'label' => 'Tahun Terbit',
120             'rules' =>
121                 'required|numeric|greater_than[1800]|less_than[2024]',
122             'errors' => [
123                 'required' => '{field} harus
124                 diisi.',
125                 'numeric' => '{field} harus berupa
126                 angka.',
127                 'greater_than' => '{field} harus
128                 lebih besar dari 1800.',
129                 'less_than' => '{field} harus kurang
130                 dari 2024.'
131             ]
132         ]
133     ];
134     if (!$validation->withRequest($this->request)-
135         >run()) {
136         return view('edit', ['validation' =>
137             $validation, 'buku' => $this->bukuModel->find($id)]);
138     }

```

129	}
130	\$tahunTerbit = \$this->request->getPost('tahun_terbit');
131	if (\$tahunTerbit > date('Y')) {
132	\$validation->setError('tahun_terbit', 'Tahun Terbit harus lebih kecil dari tahun saat ini.');
	throw new ValidationException(\$validation);
133	}
134	\$data = [
135	'judul' => \$this->request->getPost('judul'),
136	'penulis' => \$this->request->getPost('penulis'),
137	'penerbit' => \$this->request->getPost('penerbit'),
138	'tahun_terbit' => \$tahunTerbit,
139	];
140	\$this->bukuModel->update(\$id, \$data);
141	return redirect()->to('/dashboard');
142	}
143	public function delete(\$id)
144	{
145	\$this->bukuModel->delete(\$id);
146	return redirect()->to('/dashboard');
147	}
148	}

Table 3. Source Code Soal 1 Login.php

1	<?php
2	namespace App\Controllers;
3	use App\Controllers\BaseController;

```
4 use App\Models\UserModel;
5 class Login extends BaseController
6 {
7     public function loginForm()
8     {
9         helper(['form']);
10        return view('login');
11    }
12    public function login()
13    {
14        $session = session();
15        $userModel = new UserModel();
16        $identifier = $this->request->getVar('identifier');
17        $password = $this->request->getVar('password');
18        $data = $userModel->getUserByEmailOrUsername($identifier);
19        if ($data) {
20            if (password_verify($password, $data['password'])) {
21                $session->set([
22                    'user_id' => $data['id'],
23                    'username' => $data['username'],
24                    'logged_in' => true
25                ]);
26                return redirect()->to('/dashboard');
27            } else {
28                $session->setFlashdata('msg', 'Password
29                salah');
30                return redirect()->to('/login');
31            }
32        }
33    }
34 }
```

31	} else {
32	\$session->setFlashdata('msg', 'Username atau Email salah');
33	return redirect()->to('/login');
34	}
35	}
36	public function registerForm()
37	{
38	return view('register');
39	}
40	public function register()
41	{}
42	public function logout()
43	{
44	\$session = session();
45	\$session->destroy();
46	return redirect()->to('/');
47	}
48	}

Table 4. Source Code Soal 1 Home.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Controllers\BaseController;
6	use App\Models\BukuModel;
7	
8	class Home extends BaseController
9	{
10	public function index()

11	{
12	\$bukuModel = new BukuModel();
13	\$data['buku'] = \$bukuModel->findAll();
14	return view('home', \$data);
15	}
16	}

Table 5. Source Code Soal 1 CreateBukuTable.php

1	<?php
2	namespace App\Database\Migrations;
3	use CodeIgniter\Database\Migration;
4	class CreateBukuTable extends Migration
5	{
6	public function up()
7	{
8	\$this->forge->addField([
9	'id' => [
10	'type' => 'INT',
11	'constraint' => 5,
12	'unsigned' => true,
13	'auto_increment' => true,
14	],
15	'judul' => [
16	'type' => 'VARCHAR',
17	'constraint' => '255',
18	],
19	'penulis' => [
20	'type' => 'VARCHAR',
21	'constraint' => '255',
22	],
23	'penerbit' => [



24	'type'          => 'VARCHAR',
25	'constraint' => '255',
26	],
27	'tahun_terbit' => [
28	'type'          => 'YEAR',
29	],
30	]);
31	\$this->forge->addKey('id', true);
32	\$this->forge->createTable('buku');
33	}
34	public function down()
35	{
36	\$this->forge->dropTable('buku');
37	}
38	}

Table 6. Source Code Soal 1 CreateUserTable.php

1	<?php
2	namespace App\Database\Migrations;
3	use CodeIgniter\Database\Migration;
4	class CreateUserTable extends Migration
5	{
6	public function up()
7	{
8	\$this->forge->addField([
9	'id' => [
10	'type' => 'INT',
11	'unsigned' => true,
12	'auto_increment' => true,
13	],
14	'username' => [
	'type' => 'VARCHAR',

15	'constraint' => '100',
16	],
17	'email' => [
18	'type' => 'VARCHAR',
19	'constraint' => '255',
20	'unique' => true,
21	],
22	'password' => [
23	'type' => 'VARCHAR',
24	'constraint' => '255',
25	],
26	]);
27	\$this->forge->addKey('id', true);
28	\$this->forge->createTable('users');
29	}
30	public function down()
31	{
32	\$this->forge->dropTable('users');
33	}
34	}
35	

Table 7. Source Code Soal 1 AuthFilters.php

1	<?php
2	namespace App\Filters;
3	use CodeIgniter\Filters\FilterInterface;
4	use CodeIgniter\HTTP\RequestInterface;
5	use CodeIgniter\HTTP\ResponseInterface;
6	class AuthFilters implements FilterInterface
7	{
8	public function before(RequestInterface \$request,
	\$args = null)

9	{
10	\$session = session();
11	if (! \$session->get('logged_in')) {
12	return redirect()->to('/login')->with('msg',
	'Mohon login terlebih dahulu!');
13	}
14	}
15	public function after(RequestInterface \$request,
	ResponseInterface \$response, \$arguments = null)
16	{}
17	}

Table 8. Source Code Soal 1 Buku.Model.php

1	<?php
2	namespace App\Models;
3	use CodeIgniter\Model;
4	class BukuModel extends Model
5	{
6	protected \$table = 'buku';
7	protected \$primaryKey = 'id';
8	protected \$allowedFields = ['judul', 'penulis',
	'penerbit', 'tahun_terbit'];
9	}

Table 9. Source Code Soal 1 UserModel.php

1	<?php
2	namespace App\Models;
3	use CodeIgniter\Model;
4	class UserModel extends Model
5	{
6	protected \$table = 'users';
7	protected \$primaryKey = 'id';

8	protected \$allowedFields = ['username', 'email', 'password'];
9	public function
	getUserByEmailOrUsername(\$identifier)
10	{
11	return \$this->where('email', \$identifier)
12	->orWhere('username', \$identifier)
13	->first();
14	}
15	}

Table 10. Source Code Soal 1 create.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	<title>Create Buku</title>
7	<link rel="stylesheet" href="http://localhost/Modul6/app/Views/styles.css">
8	</head>
9	<body class="create">
10	<div class="container register">
11	<h1 class="register">Create Buku</h1>
12	<?php if(isset(\$validation)): ?>
13	<div class="error register">
14	<?= \$validation->listErrors() ?>
15	</div>
16	<?php endif; ?>
17	<form class="register" action="<?= base_url('/buku/store') ?>" method="post">

```
18         <?= csrf_field() ?>
19         <div class="form-group">
20             <label for="judul">Judul:</label>
21             <input      type="text"      name="judul"
id="judul" class="input register" value="<?= old('judul')
?>">
22         </div>
23         <div class="form-group">
24             <label for="penulis">Penulis:</label>
25             <input      type="text"      name="penulis"
id="penulis" class="input register" value="<?=
old('penulis') ?>">
26         </div>
27         <div class="form-group">
28             <label for="penerbit">Penerbit:</label>
29             <input      type="text"      name="penerbit"
id="penerbit" class="input register" value="<?=
old('penerbit') ?>">
30         </div>
31         <div class="form-group">
32             <label      for="tahun_terbit">Tahun
Terbit:</label>
33             <input type="number" name="tahun_terbit"
id="tahun_terbit" class="input register" value="<?=
old('tahun_terbit') ?>">
34         </div>
35         <button      type="submit"      class="btn
register">Submit</button>
36     </form>
37 </div>
38 </body>
```



24	<label for="penulis">Penulis:</label>
25	<input type="text" name="penulis" id="penulis" class="input register" value="<?=\$buku['penulis'] ?>">
26	</div>
27	<div class="form-group">
28	<label for="penerbit">Penerbit:</label>
29	<input type="text" name="penerbit" id="penerbit" class="input register" value="<?=\$buku['penerbit'] ?>">
30	</div>
31	<div class="form-group">
32	<label for="tahun_terbit">Tahun Terbit:</label>
33	<input type="text" name="tahun_terbit" id="tahun_terbit" class="input register" value="<?=\$buku['tahun_terbit'] ?>">
34	</div>
35	<button type="submit" class="btn register">Update</button>
36	</form>
37	<p class="login-link register"><a href="<?=\$base_url('/dashboard') ?>">Kembali</a></p>
38	</div>
39	</body>
40	</html>

Table 12. Source Code Soal 1 home.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	<title>Nindy Gallery</title>
7	<link rel="stylesheet" href="http://localhost/Modul6/app/Views/styles.css">
8	</head>
9	<body class="login">
10	<div class="header">
11	<h2>Welcome to Nindy Gallery</h2>
12	<div class="buttons">
13	<a href="/login" class="btn btn-login">Login</a>
14	<a href="/register" class="btn btn-register">Register</a>
15	</div>
16	</div>
17	<div class="container">
18	<div class="item">
19	
20	<div class="title">HTML 5 Manual Book</div>
21	</div>
22	<div class="item">
23	
24	<div class="title">Harry Potter and The Order of Phoenix</div>
25	</div>
26	<div class="item">



27	<pre> &lt;img src="http://localhost/Modul6/app/Views/gambar/3.JPG" alt="Book Image"&gt; </pre>
28	<pre> &lt;div class="title"&gt;Think and Grow Rich&lt;/div&gt; </pre>
29	<pre> &lt;/div&gt; </pre>
30	<pre> &lt;div class="item"&gt; </pre>
31	<pre> &lt;img src="http://localhost/Modul6/app/Views/gambar/4.JPG" alt="Book Image"&gt; </pre>
32	<pre> &lt;div class="title"&gt;Strategic Leadership&lt;/div&gt; </pre>
33	<pre> &lt;/div&gt; </pre>
34	<pre> &lt;div class="item"&gt; </pre>
35	<pre> &lt;img src="http://localhost/Modul6/app/Views/gambar/5.JPG" alt="Book Image"&gt; </pre>
36	<pre> &lt;div class="title"&gt;Republik Naga (The Dragon Republic)&lt;/div&gt; </pre>
37	<pre> &lt;/div&gt; </pre>
38	<pre> &lt;div class="item"&gt; </pre>
39	<pre> &lt;img src="http://localhost/Modul6/app/Views/gambar/6.JPG" alt="Book Image"&gt; </pre>
40	<pre> &lt;div class="title"&gt;Kecerdasan Emosional&lt;/div&gt; </pre>
41	<pre> &lt;/div&gt; </pre>
42	<pre> &lt;div class="item"&gt; </pre>
43	<pre> &lt;img src="http://localhost/Modul6/app/Views/gambar/7.JPG" alt="Book Image"&gt; </pre>

44	<code>&lt;div class="title"&gt;Pesanan dari Luar Angkasa&lt;/div&gt;</code>
45	<code>&lt;/div&gt;</code>
46	<code>&lt;div class="item"&gt;</code>
47	<code>&lt;img</code>
	<code>src="http://localhost/Modul6/app/Views/gambar/8.JPG"</code>
	<code>alt="Book Image"&gt;</code>
48	<code>&lt;div class="title"&gt;Home Sweet Loan&lt;/div&gt;</code>
49	<code>&lt;/div&gt;</code>
50	<code>&lt;/div&gt;</code>
51	<code>&lt;/body&gt;</code>
52	<code>&lt;/html&gt;</code>

Table 13. Source Code Soal 1 index.php

1	<code>&lt;!DOCTYPE html&gt;</code>
2	<code>&lt;html lang="en"&gt;</code>
3	<code>&lt;head&gt;</code>
4	<code>&lt;meta charset="UTF-8"&gt;</code>
5	<code>&lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;</code>
6	<code>&lt;title&gt;Dashboard&lt;/title&gt;</code>
7	<code>&lt;link rel="stylesheet"</code>
	<code>href="http://localhost/Modul6/app/Views/styles.css"&gt;</code>
8	<code>&lt;/head&gt;</code>
9	<code>&lt;body class="index"&gt;</code>
10	<code>&lt;h1 class="index"&gt;Daftar Buku&lt;/h1&gt;</code>
11	<code>&lt;?php if(session()-&gt;getFlashdata('success')): ?&gt;</code>
12	<code>&lt;p class="success index"&gt;&lt;?= session()-&gt;getFlashdata('success') ?&gt;&lt;/p&gt;</code>
13	<code>&lt;?php endif; ?&gt;</code>
14	<code>&lt;table class="index"&gt;</code>
15	<code>&lt;thead&gt;</code>

```

16         <tr class="index">
17             <th class="index">Judul</th>
18             <th class="index">Penulis</th>
19             <th class="index">Penerbit</th>
20             <th class="index">Tahun Terbit</th>
21             <th class="index">Aksi</th>
22         </tr>
23     </thead>
24     <tbody class="index">
25         <?php if (!empty($buku)): ?>
26             <?php foreach ($buku as $item): ?>
27                 <tr class="index">
28                     <td class="index"><?= $item['judul']
29                     <td class="index"><?=
30                     <td class="index"><?=
31                     <td class="index"><?=
32                     <td class="action-links index">
33                         <a class="index" href="<?=
34                         <a class="index" href="<?=
35                     </td>
36                 </tr>
37             <?php endforeach; ?>
38         <?php else: ?>
39             <tr class="index">

```

40	<code>&lt;td class="index" colspan="5"&gt;Tidak ada data buku yang tersedia.&lt;/td&gt;</code>
41	<code>&lt;/tr&gt;</code>
42	<code>&lt;?php endif; ?&gt;</code>
43	<code>&lt;/tbody&gt;</code>
44	<code>&lt;/table&gt;</code>
45	<code>&lt;a href="&lt;?= base_url('/buku/create') ?&gt;" class="add- link index"&gt;Tambah Buku Baru&lt;/a&gt;</code>
46	<code>&lt;p&gt;&lt;a href="&lt;?= base_url('/logout') ?&gt;" class="logout-button index"&gt;Logout&lt;/a&gt;&lt;/p&gt;</code>
47	<code>&lt;/body&gt;</code>
48	<code>&lt;/html&gt;</code>

Table 14. Source Code Soal 1 login.php

1	<code>&lt;!DOCTYPE html&gt;</code>
2	<code>&lt;html lang="en"&gt;</code>
3	<code>&lt;head&gt;</code>
4	<code>&lt;meta charset="UTF-8"&gt;</code>
5	<code>&lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;</code>
6	<code>&lt;title&gt;Login&lt;/title&gt;</code>
7	<code>&lt;link rel="stylesheet" href="http://localhost/Modul6/app/Views/styles.css"&gt;</code>
8	<code>&lt;/head&gt;</code>
9	<code>&lt;body class="login"&gt;</code>
10	<code>&lt;div class="container login"&gt;</code>
11	<code>&lt;h1 class="login"&gt;Login&lt;/h1&gt;</code>
12	<code>&lt;?php if(session()-&gt;getFlashdata('msg')): ?&gt;</code>
13	<code>&lt;div class="error login"&gt;&lt;?= session()- &gt;getFlashdata('msg') ?&gt;&lt;/div&gt;</code>
14	<code>&lt;?php endif; ?&gt;</code>

15	<code>&lt;form action="/login" method="post"</code>
	<code>class="login"&gt;</code>
16	<code>&lt;input type="text" name="identifier"</code>
	<code>placeholder="Username or Email" required class="login"&gt;</code>
17	<code>&lt;input type="password" name="password"</code>
	<code>placeholder="Password" required class="login"&gt;</code>
18	<code>&lt;button type="submit" class="btn</code>
	<code>login"&gt;Login&lt;/button&gt;</code>
19	<code>&lt;/form&gt;</code>
20	<code>&lt;div class="register-link login"&gt;</code>
21	<code>&lt;p&gt;Don't have an account? &lt;a</code>
	<code>href="/register"&gt;Register here&lt;/a&gt;&lt;/p&gt;</code>
22	<code>&lt;/div&gt;</code>
23	<code>&lt;/div&gt;</code>
24	<code>&lt;/body&gt;</code>
25	<code>&lt;/html&gt;</code>

Table 15. Source Code Soal 1 register.php

1	<code>&lt;!DOCTYPE html&gt;</code>
2	<code>&lt;html lang="en"&gt;</code>
3	<code>&lt;head&gt;</code>
4	<code>&lt;meta charset="UTF-8"&gt;</code>
5	<code>&lt;title&gt;Register&lt;/title&gt;</code>
6	<code>&lt;link rel="stylesheet"</code>
	<code>href="http://localhost/Modul6/app/Views/styles.css"&gt;</code>
7	<code>&lt;/head&gt;</code>
8	<code>&lt;body class="register"&gt;</code>
9	<code>&lt;div class="container register"&gt;</code>
10	<code>&lt;h1 class="register"&gt;Register&lt;/h1&gt;</code>
11	<code>&lt;form class="register" action="/register"</code>
	<code>method="post"&gt;</code>

```

12         <?php         if         (session()-
>getFlashdata('errors')): ?>
13             <div class="alert alert-danger">
14                 <ul>
15                     <?php         foreach         (session()-
>getFlashdata('errors') as $error): ?>
16                         <li><?= esc($error) ?></li>
17                     <?php endforeach ?>
18                 </ul>
19             </div>
20         <?php endif ?>
21         <?php         if         (session()-
>getFlashdata('success')): ?>
22             <div class="alert alert-success">
23                 <p><?=         session()-
>getFlashdata('success') ?></p>
24             </div>
25         <?php endif ?>
26         <div class="form-group">
27             <label for="username">Username:</label>
28             <input         type="text"         name="username"
id="username" class="register" value="<?= old('username')
?>" required>
29         </div>
30         <div class="form-group">
31             <label for="email">Email:</label>
32             <input         type="email"         name="email"
id="email" class="register" value="<?= old('email') ?>"
required>
33         </div>
34         <div class="form-group">

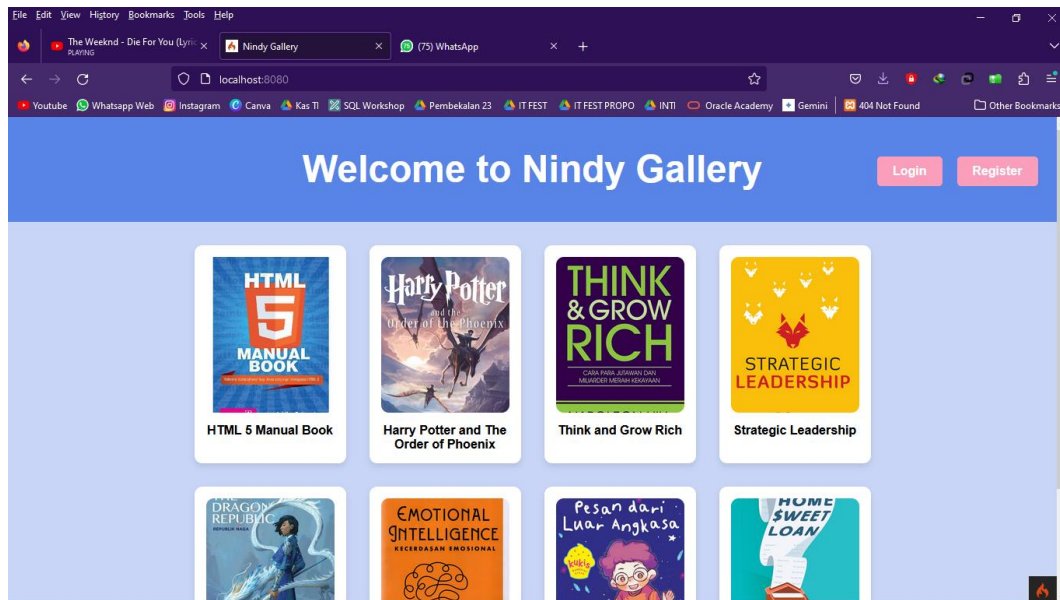
```

```

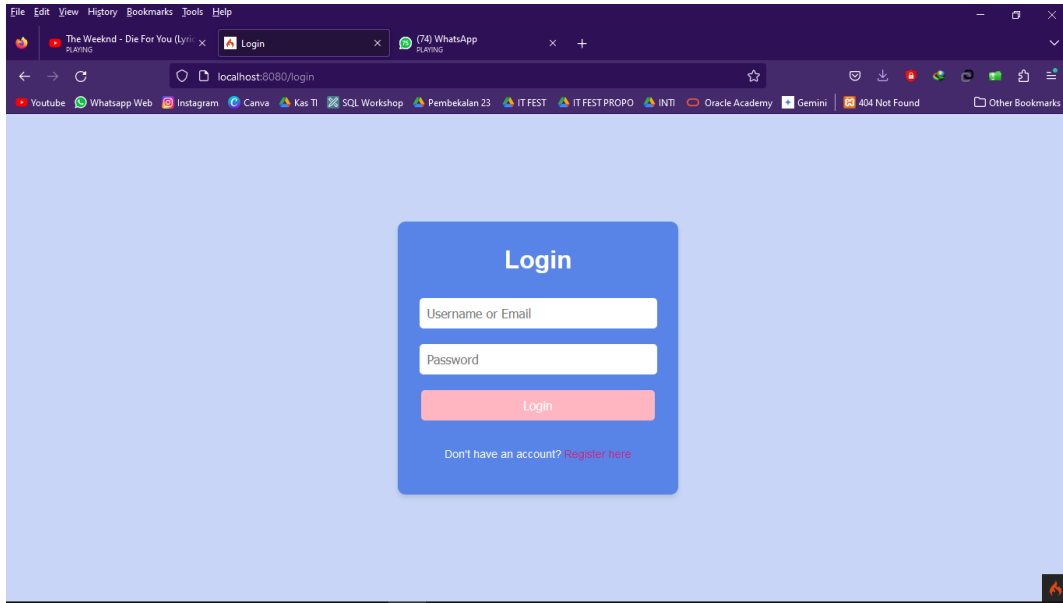
35         <label for="password">Password:</label>
36         <input type="password" name="password"
id="password" class="register" required>
37     </div>
38     <button type="submit" class="btn
register">Register</button>
39 </form>
40 <div class="login-link register">
41     Already have an account? <a href="/login"
class="register">Login here</a>
42 </div>
43 </div>
44 </body>
45 </html>

```

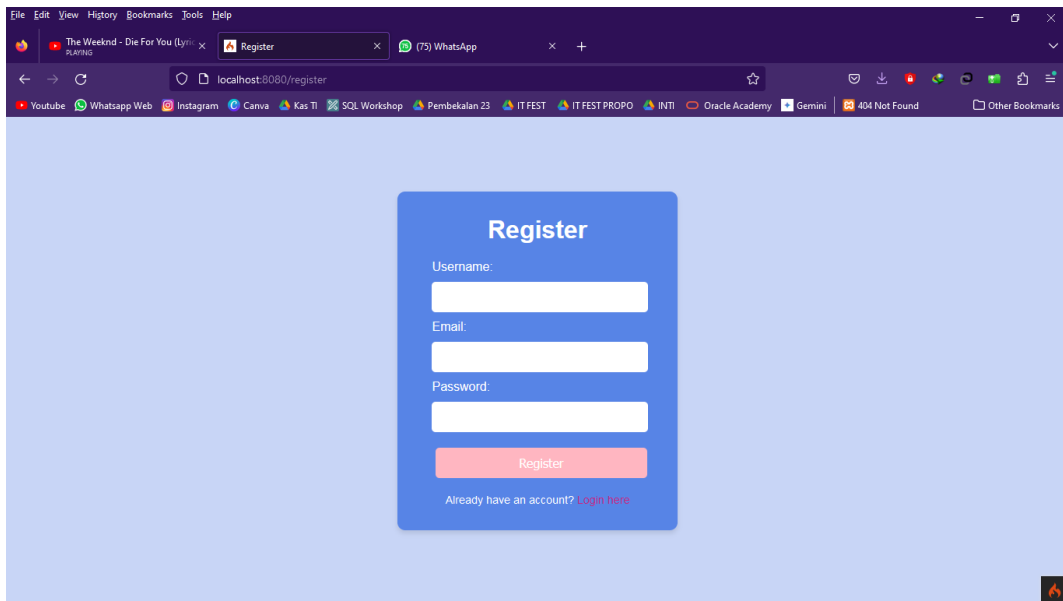
## B. Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1 Tampilan Home

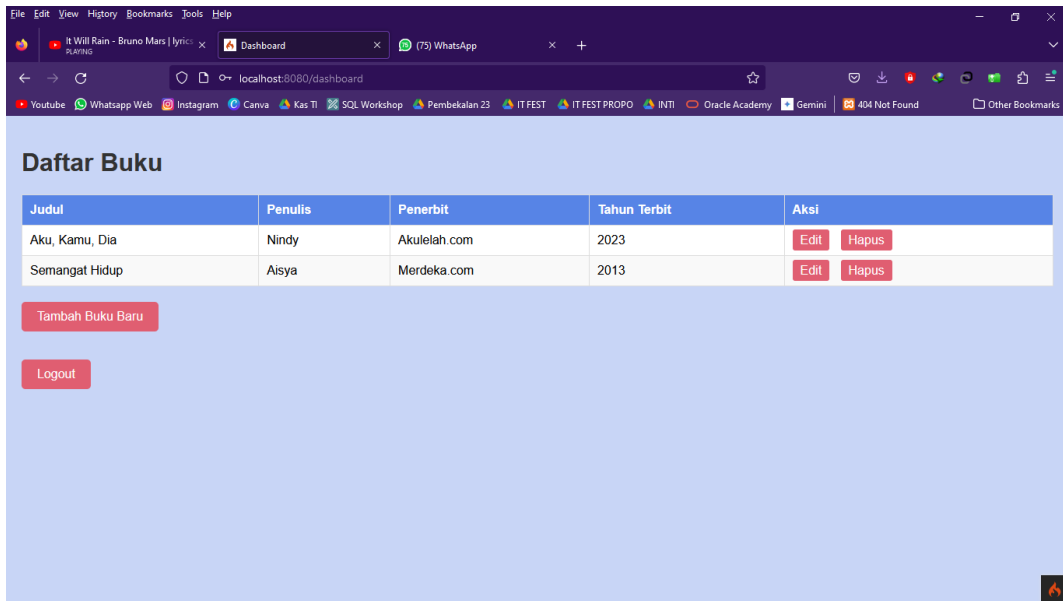


*Gambar 2. Screenshot Hasil Jawaban Soal 1 Tampilan Login*

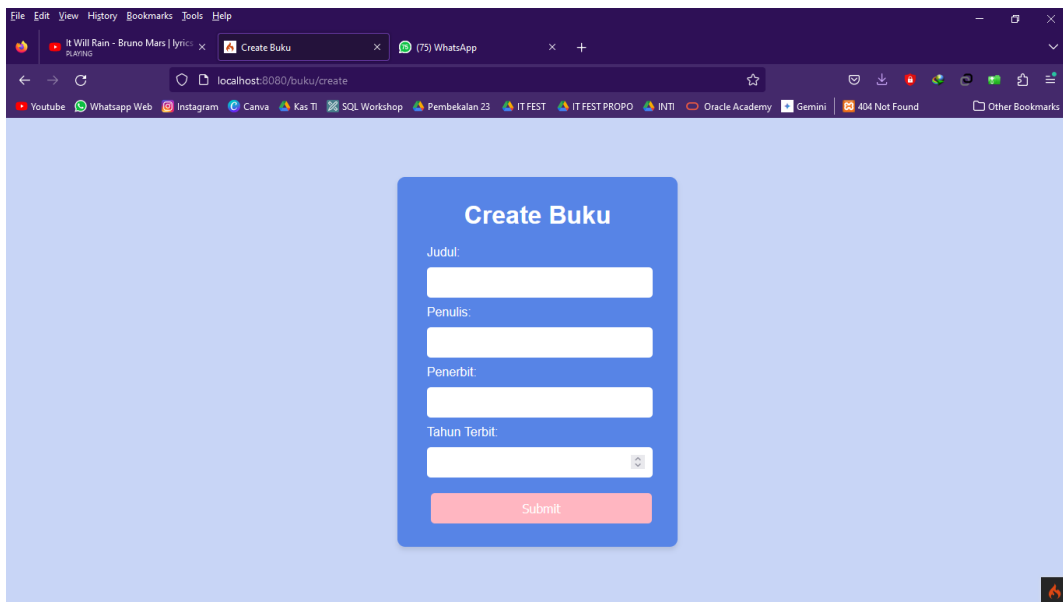


*Gambar 3. Screenshot Hasil Jawaban Soal 1 Tampilan Register*

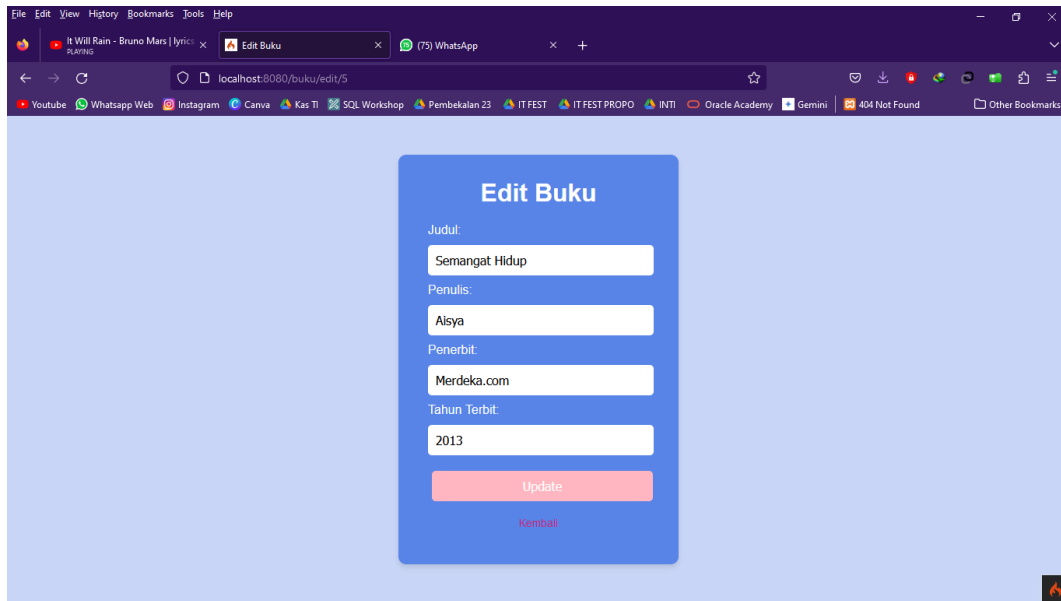




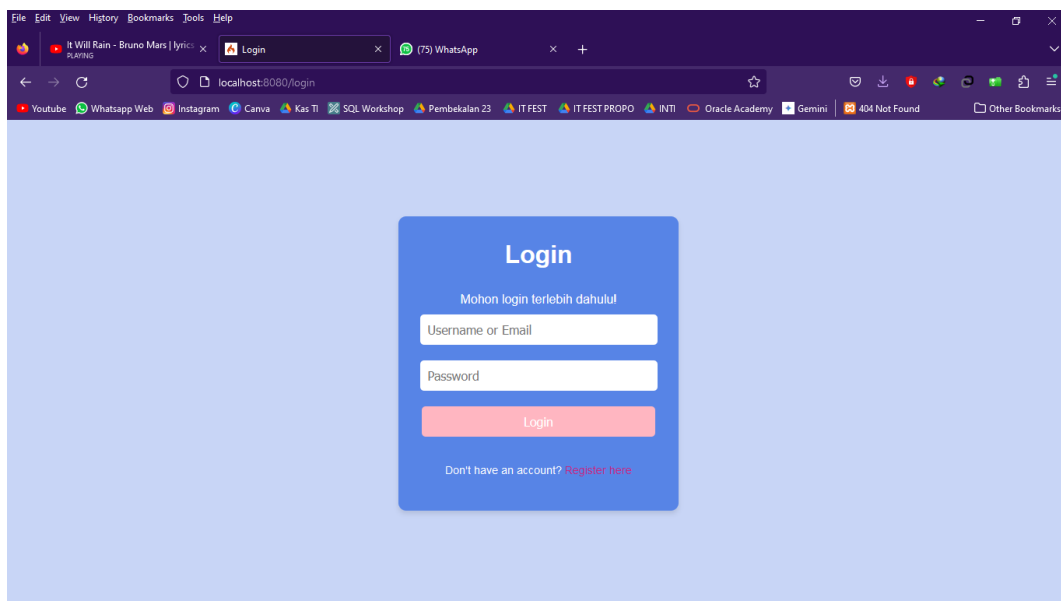
Gambar 4. Screenshot Hasil Jawaban Soal 1 Tampilan Dashboard



Gambar 5. Screenshot Hasil Jawaban Soal 1 Tampilan Create



*Gambar 6. Screenshot Hasil Jawaban Soal 1 Tampilan Edit*



*Gambar 7. Screenshot Hasil Jawaban Soal 1 mengakses Dashboard Tanpa Login*

## C. Pembahasan

### Routes.php

Pada baris [3], syntax ``use CodeIgniter\Router\RouteCollectionInterface;`` digunakan untuk menggunakan (import) antarmuka `RouteCollectionInterface` dari namespace `CodeIgniter\Router`.

Pada baris [5], syntax ``$routes->get('/', 'Home::index');`` digunakan untuk mendefinisikan rute GET pada URL root ( '/') yang akan memanggil metode ``index`` pada kontroler ``Home``.

Pada baris [6], syntax ``$routes->get('/login', 'Login::loginForm');`` digunakan untuk mendefinisikan rute GET pada URL ``/login`` yang akan memanggil metode ``loginForm`` pada kontroler ``Login``.

Pada baris [7], syntax ``$routes->post('/login', 'Login::login');`` digunakan untuk mendefinisikan rute POST pada URL ``/login`` yang akan memanggil metode ``login`` pada kontroler ``Login``.

Pada baris [8], syntax ``$routes->get('/register', 'Login::registerForm');`` digunakan untuk mendefinisikan rute GET pada URL ``/register`` yang akan memanggil metode ``registerForm`` pada kontroler ``Login``.

Pada baris [9], syntax ``$routes->post('/register', 'Login::register');`` digunakan untuk mendefinisikan rute POST pada URL ``/register`` yang akan memanggil metode ``register`` pada kontroler ``Login``.

Pada baris [10], syntax ``$routes->get('/logout', 'Login::logout');`` digunakan untuk mendefinisikan rute GET pada URL ``/logout`` yang akan memanggil metode ``logout`` pada kontroler ``Login``.

Pada baris [11], syntax ``$routes->group(", ['filter' => 'auth'], function ($routes) {`` digunakan untuk membuat grup rute yang akan menggunakan filter ``auth``. Semua rute di dalam grup ini akan diperiksa oleh filter ``auth`` untuk memastikan pengguna telah terautentikasi.

Pada baris [12], syntax ``$routes->get('/dashboard', 'Buku::index');`` digunakan untuk mendefinisikan rute GET pada URL ``/dashboard`` yang akan memanggil metode ``index`` pada kontroler ``Buku``.

Pada baris [13], syntax ``$routes->get('/buku', 'Buku::index');`` digunakan untuk mendefinisikan rute GET pada URL ``/buku`` yang akan memanggil metode ``index`` pada kontroler ``Buku``.

Pada baris [14], syntax ``$routes->get('/buku/create', 'Buku::create');`` digunakan untuk mendefinisikan rute GET pada URL ``/buku/create`` yang akan memanggil metode ``create`` pada kontroler ``Buku``.

Pada baris [15], syntax ``$routes->post('/buku/store', 'Buku::store');`` digunakan untuk mendefinisikan rute POST pada URL `'/buku/store'` yang akan memanggil metode ``store`` pada kontroler ``Buku``.

Pada baris [16], syntax ``$routes->get('/buku/edit/(:num)', 'Buku::edit/$1');`` digunakan untuk mendefinisikan rute GET pada URL `'/buku/edit/(:num)'` yang akan memanggil metode ``edit`` pada kontroler ``Buku``, dengan parameter numerik yang akan diteruskan ke metode tersebut sebagai ``$1``.

Pada baris [17], syntax ``$routes->post('/buku/update/(:num)', 'Buku::update/$1');`` digunakan untuk mendefinisikan rute POST pada URL `'/buku/update/(:num)'` yang akan memanggil metode ``update`` pada kontroler ``Buku``, dengan parameter numerik yang akan diteruskan ke metode tersebut sebagai ``$1``.

Pada baris [18], syntax ``$routes->get('/buku/delete/(:num)', 'Buku::delete/$1');`` digunakan untuk mendefinisikan rute GET pada URL `'/buku/delete/(:num)'` yang akan memanggil metode ``delete`` pada kontroler ``Buku``, dengan parameter numerik yang akan diteruskan ke metode tersebut sebagai ``$1``.

Pada baris [19], syntax ``});`` digunakan untuk menutup grup rute yang telah didefinisikan sebelumnya.

## **Buku.php**

Pada baris [2], syntax ``namespace App\Controllers;`` digunakan untuk mendefinisikan namespace dari kelas ini sebagai ``App\Controllers``.

Pada baris [3], syntax ``use App\Controllers\BaseController;`` digunakan untuk mengimpor kelas ``BaseController`` dari namespace ``App\Controllers``.

Pada baris [4], syntax ``use App\Models\BukuModel;`` digunakan untuk mengimpor kelas ``BukuModel`` dari namespace ``App\Models``.

Pada baris [5], syntax ``use CodeIgniter\Validation\Exceptions\ValidationException;`` digunakan untuk mengimpor kelas ``ValidationException`` dari namespace ``CodeIgniter\Validation\Exceptions``.

Pada baris [6], syntax ``class Buku extends BaseController`` digunakan untuk mendefinisikan kelas ``Buku`` yang merupakan turunan dari ``BaseController``.

Pada baris [8], syntax ``protected $bukuModel;`` digunakan untuk mendeklarasikan properti ``bukuModel`` yang bersifat `protected`.

Pada baris [10], syntax ``public function __construct()`` digunakan untuk mendefinisikan konstruktor kelas ``Buku``.

Pada baris [12], syntax ``$this->bukuModel = new BukuModel();`` digunakan untuk menginisialisasi properti ``bukuModel`` dengan instansi dari kelas ``BukuModel``.

Pada baris [14], syntax ``public function index()`` digunakan untuk mendefinisikan metode ``index``.

Pada baris [16], syntax ``$data['buku'] = $this->bukuModel->findAll();`` digunakan untuk mengambil semua data buku dari model dan menyimpannya dalam array ``$data`` dengan kunci `'buku'`.

Pada baris [17], syntax ``return view('index', $data);`` digunakan untuk merender view `'index'` dengan data buku.

Pada baris [19], syntax ``public function create()`` digunakan untuk mendefinisikan metode ``create``.

Pada baris [21], syntax ``return view('create');`` digunakan untuk merender view `'create'`.

Pada baris [23], syntax ``public function store()`` digunakan untuk mendefinisikan metode ``store``.

Pada baris [25], syntax ``$validation = \Config\Services::validation();`` digunakan untuk mendapatkan layanan validasi.

Pada baris [26], syntax ``$validation->setRules([...]);`` digunakan untuk menetapkan aturan validasi untuk input form.

Pada baris [62], syntax ``if (!$validation->withRequest($this->request)->run()) {`` digunakan untuk menjalankan validasi dan memeriksa apakah input valid.

Pada baris [63], syntax ``return view('create', ['validation' => $validation]);`` digunakan untuk merender kembali view `'create'` dengan pesan validasi jika validasi gagal.

Pada baris [65], syntax ``$tahunTerbit = $this->request->getPost('tahun_terbit');`` digunakan untuk mengambil nilai input `'tahun_terbit'` dari form.

Pada baris [66], syntax ``if ($tahunTerbit > date('Y')) {`` digunakan untuk memeriksa apakah `'tahun_terbit'` lebih besar dari tahun saat ini.

Pada baris [67], syntax ``$validation->setError('tahun_terbit', 'Tahun Terbit harus lebih kecil dari tahun saat ini.');` digunakan untuk menetapkan pesan kesalahan jika 'tahun\_terbit' lebih besar dari tahun saat ini.

Pada baris [68], syntax ``throw new ValidationException($validation);`` digunakan untuk melempar pengecualian validasi jika 'tahun\_terbit' tidak valid.

Pada baris [70-74], syntax ``$data = [...];`` digunakan untuk membuat array data dari input form.

Pada baris [76], syntax ``$this->bukuModel->save($data);`` digunakan untuk menyimpan data buku baru ke database.

Pada baris [77], syntax ``return redirect()->to('/dashboard');`` digunakan untuk mengarahkan pengguna ke halaman '/dashboard'.

Pada baris [79], syntax ``public function edit($id)`` digunakan untuk mendefinisikan metode ``edit`` dengan parameter ``$id``.

Pada baris [81], syntax ``$data['buku'] = $this->bukuModel->find($id);`` digunakan untuk mengambil data buku berdasarkan ID dan menyimpannya dalam array ``$data`` dengan kunci 'buku'.

Pada baris [82], syntax ``if (!$data['buku']) {`` digunakan untuk memeriksa apakah buku dengan ID tersebut ditemukan.

Pada baris [83], syntax ``throw new \CodeIgniter\Exceptions\PageNotFoundException('Buku dengan ID ' . $id . ' tidak ditemukan.');` digunakan untuk melempar pengecualian jika buku tidak ditemukan.

Pada baris [85], syntax ``return view('edit', $data);`` digunakan untuk merender view 'edit' dengan data buku.

Pada baris [87], syntax ``public function update($id)`` digunakan untuk mendefinisikan metode ``update`` dengan parameter ``$id``.

Pada baris [89], syntax ``$validation = \Config\Services::validation();`` digunakan untuk mendapatkan layanan validasi.

Pada baris [90], syntax ``$validation->setRules([...]);`` digunakan untuk menetapkan aturan validasi untuk input form.

Pada baris [126], syntax ``if (!$validation->withRequest($this->request)->run()) {`` digunakan untuk menjalankan validasi dan memeriksa apakah input valid.

Pada baris [127], syntax ``return view('edit', ['validation' => $validation, 'buku' => $this->bukuModel->find($id)]);`` digunakan untuk merender kembali view 'edit' dengan pesan validasi jika validasi gagal.

Pada baris [129], syntax ``$tahunTerbit = $this->request->getPost('tahun_terbit');`` digunakan untuk mengambil nilai input 'tahun\_terbit' dari form.

Pada baris [130], syntax ``if ($tahunTerbit > date('Y')) {`` digunakan untuk memeriksa apakah 'tahun\_terbit' lebih besar dari tahun saat ini.

Pada baris [131], syntax ``$validation->setError('tahun_terbit', 'Tahun Terbit harus lebih kecil dari tahun saat ini.');` digunakan untuk menetapkan pesan kesalahan jika 'tahun\_terbit' lebih besar dari tahun saat ini.

Pada baris [132], syntax ``throw new ValidationException($validation);`` digunakan untuk melempar pengecualian validasi jika 'tahun\_terbit' tidak valid.

Pada baris [134-138], syntax ``$data = [...];`` digunakan untuk membuat array data dari input form.

Pada baris [140], syntax ``$this->bukuModel->update($id, $data);`` digunakan untuk memperbarui data buku di database berdasarkan ID.

Pada baris [141], syntax ``return redirect()->to('/dashboard');`` digunakan untuk mengarahkan pengguna ke halaman '/dashboard'.

Pada baris [143], syntax ``public function delete($id)`` digunakan untuk mendefinisikan metode ``delete`` dengan parameter ``$id``.

Pada baris [145], syntax ``$this->bukuModel->delete($id);`` digunakan untuk menghapus data buku dari database berdasarkan ID.

Pada baris [146], syntax ``return redirect()->to('/dashboard');`` digunakan untuk mengarahkan pengguna ke halaman '/dashboard'.

## **Login.php**

Pada baris [2], syntax ``namespace App\Controllers;`` digunakan untuk mendefinisikan namespace dari kelas ini sebagai ``App\Controllers``.

Pada baris [3], syntax ``use App\Controllers\BaseController;`` digunakan untuk mengimpor kelas ``BaseController`` dari namespace ``App\Controllers``.

Pada baris [4], syntax ``use App\Models\UserModel;`` digunakan untuk mengimpor kelas ``UserModel`` dari namespace ``App\Models``.

Pada baris [5], syntax ``class Login extends BaseController`` digunakan untuk mendefinisikan kelas ``Login`` yang merupakan turunan dari ``BaseController``.

Pada baris [7], syntax ``public function loginForm()`` digunakan untuk mendefinisikan metode ``loginForm``.

Pada baris [9], syntax ``helper(['form']);`` digunakan untuk memuat helper form.

Pada baris [10], syntax ``return view('login');`` digunakan untuk merender view 'login'.

Pada baris [12], syntax ``public function login()`` digunakan untuk mendefinisikan metode ``login``.

Pada baris [14], syntax ``$session = session();`` digunakan untuk mendapatkan instance dari session.

Pada baris [15], syntax ``$userModel = new UserModel();`` digunakan untuk menginisialisasi objek ``UserModel``.

Pada baris [16], syntax ``$identifier = $this->request->getVar('identifier');`` digunakan untuk mengambil nilai input 'identifier' dari form.

Pada baris [17], syntax ``$password = $this->request->getVar('password');`` digunakan untuk mengambil nilai input 'password' dari form.

Pada baris [18], syntax ``$data = $userModel->getUserByEmailOrUsername($identifier);`` digunakan untuk mengambil data pengguna berdasarkan email atau username menggunakan metode ``getUserByEmailOrUsername``.

Pada baris [19], syntax ``if ($data) {`` digunakan untuk memeriksa apakah data pengguna ditemukan.

Pada baris [20], syntax ``if (password_verify($password, $data['password'])) {`` digunakan untuk memverifikasi apakah password yang dimasukkan cocok dengan password yang disimpan di database.

Pada baris [21], syntax ``$session->set(...);`` digunakan untuk menyimpan data pengguna ke dalam session jika password valid.

Pada baris [26], syntax ``return redirect()->to('/dashboard');`` digunakan untuk mengarahkan pengguna ke halaman '/dashboard' jika login berhasil.



Pada baris [28], syntax ``$session->setFlashdata('msg', 'Password salah');`` digunakan untuk menyimpan pesan kesalahan ke dalam session jika password salah.

Pada baris [29], syntax ``return redirect()->to('/login');`` digunakan untuk mengarahkan pengguna kembali ke halaman login jika password salah.

Pada baris [32], syntax ``$session->setFlashdata('msg', 'Username atau Email salah');`` digunakan untuk menyimpan pesan kesalahan ke dalam session jika username atau email salah.

Pada baris [33], syntax ``return redirect()->to('/login');`` digunakan untuk mengarahkan pengguna kembali ke halaman login jika username atau email salah.

Pada baris [36], syntax ``public function registerForm()`` digunakan untuk mendefinisikan metode ``registerForm``.

Pada baris [38], syntax ``return view('register');`` digunakan untuk merender view 'register'.

Pada baris [40], syntax ``public function register()`` digunakan untuk mendefinisikan metode ``register``. Metode ini kosong dan belum diimplementasikan.

Pada baris [42], syntax ``public function logout()`` digunakan untuk mendefinisikan metode ``logout``.

Pada baris [44], syntax ``$session = session();`` digunakan untuk mendapatkan instance dari session.

Pada baris [45], syntax ``$session->destroy();`` digunakan untuk menghapus semua data session.

Pada baris [46], syntax ``return redirect()->to('/')`` digunakan untuk mengarahkan pengguna ke halaman root ('/') setelah logout.

## **Home.php**

Pada baris [3], syntax ``namespace App\Controllers;`` digunakan untuk mendefinisikan namespace dari kelas ini sebagai ``App\Controllers``.

Pada baris [5], syntax ``use App\Controllers\BaseController;`` digunakan untuk mengimpor kelas ``BaseController`` dari namespace ``App\Controllers``.

Pada baris [6], syntax ``use App\Models\BukuModel;`` digunakan untuk mengimpor kelas ``BukuModel`` dari namespace ``App\Models``.

Pada baris [8], syntax ``class Home extends BaseController`` digunakan untuk mendefinisikan kelas ``Home`` yang merupakan turunan dari ``BaseController``.

Pada baris [10], syntax ``public function index()`` digunakan untuk mendefinisikan metode ``index``.

Pada baris [12], syntax ``$bukuModel = new BukuModel();`` digunakan untuk menginisialisasi objek ``BukuModel``.

Pada baris [13], syntax ``$data['buku'] = $bukuModel->findAll();`` digunakan untuk mengambil semua data buku dari model dan menyimpannya dalam array ``$data`` dengan kunci 'buku'.

Pada baris [14], syntax ``return view('home', $data);`` digunakan untuk merender view 'home' dengan data buku.

### **CreateBukuTable.php**

Pada baris [2], syntax ``namespace App\Database\Migrations;`` digunakan untuk mendefinisikan namespace dari kelas ini sebagai ``App\Database\Migrations``.

Pada baris [3], syntax ``use CodeIgniter\Database\Migration;`` digunakan untuk mengimpor kelas ``Migration`` dari namespace ``CodeIgniter\Database``.

Pada baris [4], syntax ``class CreateBukuTable extends Migration`` digunakan untuk mendefinisikan kelas ``CreateBukuTable`` yang merupakan turunan dari ``Migration``.

Pada baris [6], syntax ``public function up()`` digunakan untuk mendefinisikan metode ``up``. Metode ini digunakan untuk membuat tabel dalam database.

Pada baris [8], syntax ``$this->forge->addField([`` digunakan untuk memulai penambahan field (kolom) pada tabel yang akan dibuat.

Pada baris [31], syntax ``$this->forge->addKey('id', true);`` digunakan untuk menetapkan kolom ``id`` sebagai primary key.

Pada baris [32], syntax ``$this->forge->createTable('buku');`` digunakan untuk membuat tabel ``buku`` di database dengan kolom yang telah didefinisikan.

Pada baris [34], syntax ``public function down()`` digunakan untuk mendefinisikan metode ``down``. Metode ini digunakan untuk menghapus tabel dari database.

Pada baris [36], syntax ``$this->forge->dropTable('buku');`` digunakan untuk menghapus tabel ``buku`` dari database.

### **CreateUserTable.php**

Pada baris [2], syntax ``namespace App\Database\Migrations;`` digunakan untuk mendefinisikan namespace dari kelas ini sebagai ``App\Database\Migrations``.

Pada baris [3], syntax ``use CodeIgniter\Database\Migration;`` digunakan untuk mengimpor kelas ``Migration`` dari namespace ``CodeIgniter\Database``.

Pada baris [4], syntax ``class CreateUsersTable extends Migration`` digunakan untuk mendefinisikan kelas ``CreateUsersTable`` yang merupakan turunan dari ``Migration``.

Pada baris [6], syntax ``public function up()`` digunakan untuk mendefinisikan metode ``up``. Metode ini digunakan untuk membuat tabel dalam database.

Pada baris [8], syntax ``$this->forge->addField([`` digunakan untuk memulai penambahan field (kolom) pada tabel yang akan dibuat.

Pada baris [28], syntax ``$this->forge->addKey('id', true);`` digunakan untuk menetapkan kolom ``id`` sebagai primary key.

Pada baris [29], syntax ``$this->forge->createTable('users');`` digunakan untuk membuat tabel ``users`` di database dengan kolom yang telah didefinisikan.

Pada baris [31], syntax ``public function down()`` digunakan untuk mendefinisikan metode ``down``. Metode ini digunakan untuk menghapus tabel dari database.

Pada baris [33], syntax ``$this->forge->dropTable('users');`` digunakan untuk menghapus tabel ``users`` dari database.

### **AuthFilters.php**

Pada baris [2], syntax ``namespace App\Filters;`` digunakan untuk mendefinisikan namespace dari kelas ini sebagai ``App\Filters``.

Pada baris [3], syntax ``use CodeIgniter\Filters\FilterInterface;`` digunakan untuk mengimpor interface ``FilterInterface`` dari namespace ``CodeIgniter\Filters``.

Pada baris [4], syntax ``use CodeIgniter\HTTP\RequestInterface;`` digunakan untuk mengimpor interface ``RequestInterface`` dari namespace ``CodeIgniter\HTTP``.

Pada baris [5], syntax ``use CodeIgniter\HTTP\ResponseInterface;`` digunakan untuk mengimpor interface ``ResponseInterface`` dari namespace ``CodeIgniter\HTTP``.

Pada baris [6], syntax ``class AuthFilters implements FilterInterface`` digunakan untuk mendefinisikan kelas ``AuthFilters`` yang mengimplementasikan ``FilterInterface``.

Pada baris [8], syntax ``public function before(RequestInterface $request, $arguments = null)`` digunakan untuk mendefinisikan metode ``before``. Metode ini akan dieksekusi sebelum request diproses.

Pada baris [10], syntax ``$session = session();`` digunakan untuk mendapatkan instance dari session.

Pada baris [11], syntax ``if (!$session->get('logged_in')) {`` digunakan untuk memeriksa apakah sesi ``logged_in`` tidak ada atau bernilai false.

Pada baris [12], syntax ``return redirect()->to('/login')->with('msg', 'Mohon login terlebih dahulu!');`` digunakan untuk mengarahkan pengguna ke halaman login dengan pesan 'Mohon login terlebih dahulu!' jika sesi ``logged_in`` tidak ada atau bernilai false.

Pada baris [15], syntax ``public function after(RequestInterface $request, ResponseInterface $response, $arguments = null)`` digunakan untuk mendefinisikan metode ``after``. Metode ini akan dieksekusi setelah request diproses.

Pada baris [16], syntax ``{}`` digunakan untuk mendefinisikan tubuh kosong dari metode ``after``.

### **BukuModel.php**

Pada baris [2], syntax ``namespace App\Models;`` digunakan untuk mendefinisikan namespace dari kelas ini sebagai ``App\Models``.

Pada baris [3], syntax ``use CodeIgniter\Model;`` digunakan untuk mengimpor kelas ``Model`` dari namespace ``CodeIgniter``.

Pada baris [4], syntax ``class BukuModel extends Model`` digunakan untuk mendefinisikan kelas ``BukuModel`` yang merupakan turunan dari ``Model``.

Pada baris [6], syntax ``protected $table = 'buku';`` digunakan untuk mendefinisikan nama tabel yang akan digunakan oleh model ini, yaitu ``buku``.

Pada baris [7], syntax ``protected $primaryKey = 'id';`` digunakan untuk mendefinisikan primary key dari tabel ``buku``, yaitu ``id``.

Pada baris [8], syntax ``protected $allowedFields = ['judul', 'penulis', 'penerbit', 'tahun_terbit'];`` digunakan untuk mendefinisikan field-field yang diperbolehkan untuk diisi atau diubah melalui model ini, yaitu ``judul``, ``penulis``, ``penerbit``, dan ``tahun_terbit``.

## **UserModel.php**

Pada baris [2], syntax ``namespace App\Models;`` digunakan untuk mendefinisikan namespace dari kelas ini sebagai ``App\Models``.

Pada baris [3], syntax ``use CodeIgniter\Model;`` digunakan untuk mengimpor kelas ``Model`` dari namespace ``CodeIgniter``.

Pada baris [4], syntax ``class UserModel extends Model`` digunakan untuk mendefinisikan kelas ``UserModel`` yang merupakan turunan dari ``Model``.

Pada baris [6], syntax ``protected $table = 'users';`` digunakan untuk mendefinisikan nama tabel yang akan digunakan oleh model ini, yaitu ``users``.

Pada baris [7], syntax ``protected $primaryKey = 'id';`` digunakan untuk mendefinisikan primary key dari tabel ``users``, yaitu ``id``.

Pada baris [8], syntax ``protected $allowedFields = ['username', 'email', 'password'];`` digunakan untuk mendefinisikan field-field yang diperbolehkan untuk diisi atau diubah melalui model ini, yaitu ``username``, ``email``, dan ``password``.

Pada baris [9], syntax ``public function getUserByEmailOrUsername($identifier)`` digunakan untuk mendefinisikan metode ``getUserByEmailOrUsername``. Metode ini bertujuan untuk mendapatkan data pengguna berdasarkan email atau username.

Pada baris [11], syntax ``return $this->where('email', $identifier)`` digunakan untuk menambahkan kondisi pencarian di mana kolom ``email`` sesuai dengan nilai ``$identifier``.

Pada baris [12], syntax ``->orWhere('username', $identifier)`` digunakan untuk menambahkan kondisi pencarian di mana kolom ``username`` sesuai dengan nilai ``$identifier``, sebagai alternatif dari kondisi sebelumnya.

Pada baris [13], syntax ``->first();`` digunakan untuk mengambil baris pertama yang sesuai dengan kondisi pencarian dan mengembalikannya.

## **create.php**

Pada baris [1], syntax ``<!DOCTYPE html>`` digunakan untuk mendefinisikan dokumen ini sebagai dokumen HTML5.

Pada baris [2], syntax ``<html lang="en">`` digunakan untuk mendefinisikan elemen root dari dokumen HTML dan menetapkan bahasa dokumen sebagai bahasa Inggris.

Pada baris [3], syntax `<head>` digunakan untuk membuka elemen head yang berisi metadata dan link eksternal untuk dokumen HTML.

Pada baris [4], syntax `<meta charset="UTF-8">` digunakan untuk menetapkan karakter encoding dokumen sebagai UTF-8.

Pada baris [5], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` digunakan untuk menetapkan viewport agar sesuai dengan lebar perangkat dan skala awal 1.0, yang berguna untuk tampilan responsif pada perangkat seluler.

Pada baris [6], syntax `<title>Create Buku</title>` digunakan untuk menetapkan judul dokumen yang akan ditampilkan pada tab browser.

Pada baris [7], syntax `<link rel="stylesheet" href="http://localhost/Modul6/app/Views/styles.css">` digunakan untuk menautkan file CSS eksternal yang berisi aturan gaya untuk dokumen ini.

Pada baris [8], syntax `</head>` digunakan untuk menutup elemen head.

Pada baris [9], syntax `<body class="create">` digunakan untuk membuka elemen body dan menetapkan kelas CSS `create` untuk elemen body.

Pada baris [10], syntax `<div class="container register">` digunakan untuk membuka elemen div dengan kelas CSS `container register`, yang berfungsi sebagai pembungkus utama konten halaman.

Pada baris [11], syntax `<h1 class="register">Create Buku</h1>` digunakan untuk menampilkan judul halaman sebagai heading level 1 dengan kelas CSS `register`.

Pada baris [12], syntax `<?php if(isset($validation)): ?>` digunakan untuk memeriksa apakah variabel `$validation` ada.

Pada baris [13], syntax `<div class="error register">` digunakan untuk membuka elemen div dengan kelas CSS `error register` yang akan menampilkan pesan kesalahan jika `$validation` ada.

Pada baris [14], syntax `<?= $validation->listErrors() ?>` digunakan untuk menampilkan daftar kesalahan validasi.

Pada baris [15], syntax `</div>` digunakan untuk menutup elemen div yang menampilkan pesan kesalahan.

Pada baris [16], syntax `<?php endif; ?>` digunakan untuk menutup blok pernyataan if dari PHP.

Pada baris [17], syntax `<form class="register" action="<?= base_url('/buku/store') ?>" method="post">` digunakan untuk membuka elemen form dengan kelas CSS `register`, menetapkan aksi form ke URL `base_url('/buku/store')`, dan menetapkan metode pengiriman form sebagai POST.

Pada baris [18], syntax `<?= csrf_field() ?>` digunakan untuk menghasilkan field token CSRF (Cross-Site Request Forgery) untuk keamanan form.

Pada baris [19], syntax `<div class="form-group">` digunakan untuk membuka elemen div dengan kelas CSS `form-group`, yang berfungsi sebagai pembungkus untuk setiap grup input form.

Pada baris [20], syntax `<label for="judul">Judul:</label>` digunakan untuk menampilkan label "Judul" yang terkait dengan input field `judul`.

Pada baris [21], syntax `<input type="text" name="judul" id="judul" class="input register" value="<?= old('judul') ?>">` digunakan untuk membuat input field tipe teks dengan nama `judul`, id `judul`, kelas CSS `input register`, dan nilai default dari data lama yang dikirimkan sebelumnya.

Pada baris [22], syntax `</div>` digunakan untuk menutup elemen div pembungkus grup input `judul`.

Pada baris [23], syntax `<div class="form-group">` digunakan untuk membuka elemen div dengan kelas CSS `form-group` untuk input `penulis`.

Pada baris [24], syntax `<label for="penulis">Penulis:</label>` digunakan untuk menampilkan label "Penulis" yang terkait dengan input field `penulis`.

Pada baris [25], syntax `<input type="text" name="penulis" id="penulis" class="input register" value="<?= old('penulis') ?>">` digunakan untuk membuat input field tipe teks dengan nama `penulis`, id `penulis`, kelas CSS `input register`, dan nilai default dari data lama yang dikirimkan sebelumnya.

Pada baris [26], syntax `</div>` digunakan untuk menutup elemen div pembungkus grup input `penulis`.

Pada baris [27], syntax `<div class="form-group">` digunakan untuk membuka elemen div dengan kelas CSS `form-group` untuk input `penerbit`.

Pada baris [28], syntax `<label for="penerbit">Penerbit:</label>` digunakan untuk menampilkan label "Penerbit" yang terkait dengan input field `penerbit`.

Pada baris [29], syntax `<input type="text" name="penerbit" id="penerbit" class="input register" value="<?= old('penerbit') ?>">` digunakan untuk membuat input field tipe teks dengan nama `penerbit`, id `penerbit`, kelas CSS `input register`, dan nilai default dari data lama yang dikirimkan sebelumnya.

Pada baris [30], syntax `</div>` digunakan untuk menutup elemen div pembungkus grup input `penerbit`.

Pada baris [31], syntax `<div class="form-group">` digunakan untuk membuka elemen div dengan kelas CSS `form-group` untuk input `tahun_terbit`.

Pada baris [32], syntax `<label for="tahun_terbit">Tahun Terbit:</label>` digunakan untuk menampilkan label "Tahun Terbit" yang terkait dengan input field `tahun_terbit`.

Pada baris [33], syntax `<input type="number" name="tahun_terbit" id="tahun_terbit" class="input register" value="<?= old('tahun_terbit') ?>">` digunakan untuk membuat input field tipe number dengan nama `tahun_terbit`, id `tahun_terbit`, kelas CSS `input register`, dan nilai default dari data lama yang dikirimkan sebelumnya.

Pada baris [34], syntax `</div>` digunakan untuk menutup elemen div pembungkus grup input `tahun_terbit`.

Pada baris [35], syntax `<button type="submit" class="btn register">Submit</button>` digunakan untuk membuat tombol submit dengan kelas CSS `btn register` yang akan mengirimkan form.

Pada baris [36], syntax `</form>` digunakan untuk menutup elemen form.

Pada baris [37], syntax `</div>` digunakan untuk menutup elemen div pembungkus utama konten halaman.

Pada baris [38], syntax `</body>` digunakan untuk menutup elemen body.

Pada baris [39], syntax `</html>` digunakan untuk menutup elemen root dari dokumen HTML.

## **edit.php**

Pada baris [1], syntax `<!DOCTYPE html>` digunakan untuk mendefinisikan dokumen ini sebagai dokumen HTML5.

Pada baris [2], syntax `<html lang="en">` digunakan untuk mendefinisikan elemen root dari dokumen HTML dan menetapkan bahasa dokumen sebagai bahasa Inggris.



Pada baris [3], syntax `<head>` digunakan untuk membuka elemen head yang berisi metadata dan link eksternal untuk dokumen HTML.

Pada baris [4], syntax `<meta charset="UTF-8">` digunakan untuk menetapkan karakter encoding dokumen sebagai UTF-8.

Pada baris [5], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` digunakan untuk menetapkan viewport agar sesuai dengan lebar perangkat dan skala awal 1.0, yang berguna untuk tampilan responsif pada perangkat seluler.

Pada baris [6], syntax `<title>Edit Buku</title>` digunakan untuk menetapkan judul dokumen yang akan ditampilkan pada tab browser.

Pada baris [7], syntax `<link rel="stylesheet" href="http://localhost/Modul6/app/Views/styles.css">` digunakan untuk menautkan file CSS eksternal yang berisi aturan gaya untuk dokumen ini.

Pada baris [8], syntax `</head>` digunakan untuk menutup elemen head.

Pada baris [9], syntax `<body class="edit">` digunakan untuk membuka elemen body dan menetapkan kelas CSS `edit` untuk elemen body.

Pada baris [10], syntax `<div class="container register">` digunakan untuk membuka elemen div dengan kelas CSS `container register`, yang berfungsi sebagai pembungkus utama konten halaman.

Pada baris [11], syntax `<h1 class="register">Edit Buku</h1>` digunakan untuk menampilkan judul halaman sebagai heading level 1 dengan kelas CSS `register`.

Pada baris [12], syntax `<?php if(isset($validation)): ?>` digunakan untuk memeriksa apakah variabel `$validation` ada.

Pada baris [13], syntax `<div class="error register">` digunakan untuk membuka elemen div dengan kelas CSS `error register` yang akan menampilkan pesan kesalahan jika `$validation` ada.

Pada baris [14], syntax `<?= $validation->listErrors() ?>` digunakan untuk menampilkan daftar kesalahan validasi.

Pada baris [15], syntax `</div>` digunakan untuk menutup elemen div yang menampilkan pesan kesalahan.

Pada baris [16], syntax `<?php endif; ?>` digunakan untuk menutup blok pernyataan if dari PHP.

Pada baris [17], syntax `<form class="register" action="<?= base_url('/buku/update/' . $buku['id']) ?>" method="post">` digunakan untuk membuka elemen form dengan kelas CSS `register`, menetapkan aksi form ke URL `base_url('/buku/update/' . $buku['id'])`, dan menetapkan metode pengiriman form sebagai POST.

Pada baris [18], syntax `<?= csrf_field() ?>` digunakan untuk menghasilkan field token CSRF (Cross-Site Request Forgery) untuk keamanan form.

Pada baris [19], syntax `<div class="form-group">` digunakan untuk membuka elemen div dengan kelas CSS `form-group`, yang berfungsi sebagai pembungkus untuk setiap grup input form.

Pada baris [20], syntax `<label for="judul">Judul:</label>` digunakan untuk menampilkan label "Judul" yang terkait dengan input field `judul`.

Pada baris [21], syntax `<input type="text" name="judul" id="judul" class="input register" value="<?= $buku['judul'] ?>">` digunakan untuk membuat input field tipe teks dengan nama `judul`, id `judul`, kelas CSS `input register`, dan nilai dari `$buku['judul']`.

Pada baris [22], syntax `</div>` digunakan untuk menutup elemen div pembungkus grup input `judul`.

Pada baris [23], syntax `<div class="form-group">` digunakan untuk membuka elemen div dengan kelas CSS `form-group` untuk input `penulis`.

Pada baris [24], syntax `<label for="penulis">Penulis:</label>` digunakan untuk menampilkan label "Penulis" yang terkait dengan input field `penulis`.

Pada baris [25], syntax `<input type="text" name="penulis" id="penulis" class="input register" value="<?= $buku['penulis'] ?>">` digunakan untuk membuat input field tipe teks dengan nama `penulis`, id `penulis`, kelas CSS `input register`, dan nilai dari `$buku['penulis']`.

Pada baris [26], syntax `</div>` digunakan untuk menutup elemen div pembungkus grup input `penulis`.

Pada baris [27], syntax `<div class="form-group">` digunakan untuk membuka elemen div dengan kelas CSS `form-group` untuk input `penerbit`.

Pada baris [28], syntax `<label for="penerbit">Penerbit:</label>` digunakan untuk menampilkan label "Penerbit" yang terkait dengan input field `penerbit`.

Pada baris [29], syntax `<input type="text" name="penerbit" id="penerbit" class="input register" value="<?= $buku['penerbit'] ?>">` digunakan untuk membuat input field tipe teks dengan nama `penerbit`, id `penerbit`, kelas CSS `input register`, dan nilai dari `$buku['penerbit']`.

Pada baris [30], syntax `</div>` digunakan untuk menutup elemen div pembungkus grup input `penerbit`.

Pada baris [31], syntax `<div class="form-group">` digunakan untuk membuka elemen div dengan kelas CSS `form-group` untuk input `tahun_terbit`.

Pada baris [32], syntax `<label for="tahun_terbit">Tahun Terbit:</label>` digunakan untuk menampilkan label "Tahun Terbit" yang terkait dengan input field `tahun_terbit`.

Pada baris [33], syntax `<input type="text" name="tahun_terbit" id="tahun_terbit" class="input register" value="<?= $buku['tahun_terbit'] ?>">` digunakan untuk membuat input field tipe teks dengan nama `tahun_terbit`, id `tahun_terbit`, kelas CSS `input register`, dan nilai dari `$buku['tahun_terbit']`.

Pada baris [34], syntax `</div>` digunakan untuk menutup elemen div pembungkus grup input `tahun_terbit`.

Pada baris [35], syntax `<button type="submit" class="btn register">Update</button>` digunakan untuk membuat tombol submit dengan kelas CSS `btn register` yang akan mengirimkan form.

Pada baris [36], syntax `</form>` digunakan untuk menutup elemen form.

Pada baris [37], syntax `<p class="login-link register"><a href="<?= base_url('/dashboard') ?>">Kembali</a></p>` digunakan untuk menampilkan link "Kembali" dengan kelas CSS `login-link register` yang mengarahkan pengguna ke halaman dashboard.

Pada baris [38], syntax `</div>` digunakan untuk menutup elemen div pembungkus utama konten halaman.

Pada baris [39], syntax `</body>` digunakan untuk menutup elemen body.

Pada baris [40], syntax `</html>` digunakan untuk menutup elemen root dari dokumen HTML.

## home.php

Pada baris [1], syntax `<!DOCTYPE html>` digunakan untuk mendefinisikan jenis dokumen HTML yang digunakan, yaitu dokumen HTML5.

Pada baris [2], syntax `<html lang="en">` digunakan untuk menandakan awal dari elemen HTML dan menetapkan bahasa default dokumen ke bahasa Inggris (English).

Pada baris [3], syntax `<head>` digunakan untuk menandai awal dari bagian kepala dokumen HTML yang berisi informasi meta-data dan tautan ke file eksternal seperti CSS atau JavaScript.

Pada baris [4], syntax `<meta charset="UTF-8">` digunakan untuk menginformasikan browser bahwa karakter encoding yang digunakan dalam dokumen adalah UTF-8, yang mendukung karakter internasional.

Pada baris [5], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` digunakan untuk mendefinisikan tampilan viewport pada perangkat mobile dengan lebar yang sama dengan lebar layar perangkat dan skalanya dimulai dari 1.0.

Pada baris [6], syntax `<title>Nindy Gallery</title>` digunakan untuk menetapkan judul halaman web yang akan ditampilkan di tab browser atau judul jendela.

Pada baris [7], syntax `<link rel="stylesheet" href="http://localhost/Modul6/app/Views/styles.css">` digunakan untuk menghubungkan dokumen dengan file eksternal CSS yang berada di alamat URL tertentu untuk mengatur tata letak dan gaya halaman.

Pada baris [8], syntax `</head>` digunakan untuk menandakan akhir dari bagian kepala dokumen HTML.

Pada baris [9], syntax `<body class="login">` digunakan untuk menetapkan kelas CSS "login" ke elemen `<body>`, yang dapat digunakan untuk mengatur gaya halaman khusus untuk halaman login.

Pada baris [10], syntax `<div class="header">` digunakan untuk memulai div dengan kelas "header" yang umumnya digunakan untuk bagian kepala atau header dari halaman web.

Pada baris [11], syntax `<h2>Welcome to Nindy Gallery</h2>` digunakan untuk menampilkan judul halaman kedua (h2) yang berisi teks "Welcome to Nindy Gallery".

Pada baris [12], syntax `<div class="buttons">` digunakan untuk memulai div dengan kelas "buttons", yang mungkin berisi tombol atau elemen navigasi lainnya.

Pada baris [13], syntax `<a href="/login" class="btn btn-login">Login</a>` digunakan untuk membuat tautan (link) dengan teks "Login" yang mengarah ke "/login" dan memiliki kelas CSS "btn" dan "btn-login".

Pada baris [14], syntax `<a href="/register" class="btn btn-register">Register</a>` digunakan untuk membuat tautan (link) dengan teks "Register" yang mengarah ke "/register" dan memiliki kelas CSS "btn" dan "btn-register".

Pada baris [15], syntax `</div>` digunakan untuk menutup div dengan kelas "buttons".

Pada baris [16], syntax `</div>` digunakan untuk menutup div dengan kelas "header".

Pada baris [17], syntax `<div class="container">` digunakan untuk memulai div dengan kelas "container", yang umumnya digunakan untuk mengelompokkan elemen-elemen dalam tata letak halaman.

Pada baris [18] - [49], setiap syntax `<div class="item">` dan isinya (gambar dan judul buku) digunakan untuk menampilkan setiap buku dalam galeri dengan gambar dan judul yang sesuai.

Pada baris [50], syntax `</div>` digunakan untuk menutup div dengan kelas "container", yang menandakan akhir dari konten utama halaman.

Pada baris [51], syntax `</body>` digunakan untuk menutup elemen `<body>`, menandakan akhir dari konten utama halaman web.

Pada baris [52], syntax `</html>` digunakan untuk menutup elemen `<html>`, menandakan akhir dari dokumen HTML secara keseluruhan.

## **index.php**

Pada baris [1], syntax `<!DOCTYPE html>` digunakan untuk mendefinisikan jenis dokumen HTML yang digunakan, yaitu dokumen HTML5.

Pada baris [2], syntax `<html lang="en">` digunakan untuk menandakan awal dari elemen HTML dan menetapkan bahasa default dokumen ke bahasa Inggris (English).

Pada baris [3], syntax `<head>` digunakan untuk menandai awal dari bagian kepala dokumen HTML yang berisi informasi meta-data dan tautan ke file eksternal seperti CSS atau JavaScript.

Pada baris [4], syntax `<meta charset="UTF-8">` digunakan untuk menginformasikan browser bahwa karakter encoding yang digunakan dalam dokumen adalah UTF-8, yang mendukung karakter internasional.

Pada baris [5], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` digunakan untuk mendefinisikan tampilan viewport pada perangkat mobile dengan lebar yang sama dengan lebar layar perangkat dan skalanya dimulai dari 1.0.

Pada baris [6], syntax `<title>Dashboard</title>` digunakan untuk menetapkan judul halaman web yang akan ditampilkan di tab browser atau judul jendela.

Pada baris [7], syntax `<link rel="stylesheet" href="http://localhost/Modul6/app/Views/styles.css">` digunakan untuk menghubungkan dokumen dengan file eksternal CSS yang berada di alamat URL tertentu untuk mengatur tata letak dan gaya halaman.

Pada baris [8], syntax `</head>` digunakan untuk menandakan akhir dari bagian kepala dokumen HTML.

Pada baris [9], syntax `<body class="index">` digunakan untuk menetapkan kelas CSS "index" ke elemen `<body>`, yang dapat digunakan untuk mengatur gaya halaman khusus untuk halaman indeks (dashboard).

Pada baris [10], syntax `<h1 class="index">Daftar Buku</h1>` digunakan untuk menampilkan judul utama halaman dengan kelas CSS "index".

Pada baris [11] - [13], PHP script digunakan untuk menampilkan pesan sukses menggunakan session Flashdata jika ada, dengan kelas CSS "success" dan "index".

Pada baris [14], syntax `<table class="index">` digunakan untuk membuat tabel dengan kelas CSS "index", yang digunakan untuk menampilkan daftar buku.

Pada baris [15] - [23], syntax `<thead>`, `<tr>`, `<th>` digunakan untuk mendefinisikan baris header tabel dengan kelas CSS "index".

Pada baris [25] - [37], PHP script digunakan untuk menampilkan data buku dari variabel `$buku` dalam tabel menggunakan pengulangan `foreach`, dengan setiap sel diberi kelas CSS "index".

Pada baris [44], syntax `</table>` digunakan untuk menutup elemen tabel.

Pada baris [45], syntax `<a href="<?= base_url('/buku/create') ?>" class="add-link index">Tambah Buku Baru</a>` digunakan untuk membuat tautan untuk menambah buku baru dengan kelas CSS "add-link" dan "index".

Pada baris [46], syntax `<p><a href="<?= base_url('/logout') ?>" class="logout-button index">Logout</a></p>` digunakan untuk membuat tautan logout dengan kelas CSS "logout-button" dan "index".

Pada baris [47], syntax `</body>` digunakan untuk menutup elemen `<body>`, menandakan akhir dari konten utama halaman web.

Pada baris [48], syntax `</html>` digunakan untuk menutup elemen `<html>`, menandakan akhir dari dokumen HTML secara keseluruhan.

### **login.php**

Pada baris [1], syntax `<!DOCTYPE html>` digunakan untuk mendefinisikan jenis dokumen HTML yang digunakan, yaitu dokumen HTML5.

Pada baris [2], syntax `<html lang="en">` digunakan untuk menandakan awal dari elemen HTML dan menetapkan bahasa default dokumen ke bahasa Inggris (English).

Pada baris [3], syntax `<head>` digunakan untuk menandai awal dari bagian kepala dokumen HTML yang berisi informasi meta-data dan tautan ke file eksternal seperti CSS atau JavaScript.

Pada baris [4], syntax `<meta charset="UTF-8">` digunakan untuk menginformasikan browser bahwa karakter encoding yang digunakan dalam dokumen adalah UTF-8, yang mendukung karakter internasional.

Pada baris [5], syntax `<meta name="viewport" content="width=device-width, initial-scale=1.0">` digunakan untuk mendefinisikan tampilan viewport pada perangkat mobile dengan lebar yang sama dengan lebar layar perangkat dan skalanya dimulai dari 1.0.

Pada baris [6], syntax `<title>Login</title>` digunakan untuk menetapkan judul halaman web yang akan ditampilkan di tab browser atau judul jendela.

Pada baris [7], syntax `<link rel="stylesheet" href="http://localhost/Modul6/app/Views/styles.css">` digunakan untuk menghubungkan dokumen dengan file eksternal CSS yang berada di alamat URL tertentu untuk mengatur tata letak dan gaya halaman.

Pada baris [8], syntax `</head>` digunakan untuk menandakan akhir dari bagian kepala dokumen HTML.

Pada baris [9], syntax `<body class="login">` digunakan untuk menetapkan kelas CSS "login" ke elemen `<body>`, yang dapat digunakan untuk mengatur gaya halaman khusus untuk halaman login.

Pada baris [10], syntax `<div class="container login">` digunakan untuk memulai div dengan kelas "container" dan "login", yang umumnya digunakan untuk mengelompokkan elemen-elemen dalam tata letak halaman.

Pada baris [11], syntax `<h1 class="login">Login</h1>` digunakan untuk menampilkan judul halaman dengan teks "Login" dan kelas CSS "login".

Pada baris [12] - [14], PHP script digunakan untuk menampilkan pesan error menggunakan session Flashdata jika ada, dengan kelas CSS "error" dan "login".

Pada baris [15], syntax `<form action="/login" method="post" class="login">` digunakan untuk memulai formulir login yang mengirim data ke `/login` menggunakan metode POST, dengan kelas CSS "login".

Pada baris [16], syntax `<input type="text" name="identifier" placeholder="Username or Email" required class="login">` digunakan untuk membuat input teks untuk username atau email, yang diperlukan (required), dengan kelas CSS "login".

Pada baris [17], syntax `<input type="password" name="password" placeholder="Password" required class="login">` digunakan untuk membuat input password, yang diperlukan (required), dengan kelas CSS "login".

Pada baris [18], syntax `<button type="submit" class="btn login">Login</button>` digunakan untuk membuat tombol submit dengan teks "Login" dan kelas CSS "btn" dan "login".

Pada baris [19], syntax `</form>` digunakan untuk menutup elemen formulir.

Pada baris [20], syntax `<div class="register-link login">` digunakan untuk memulai div dengan kelas "register-link" dan "login", yang berisi tautan untuk mendaftar.

Pada baris [21], syntax `<p>Don't have an account? <a href="/register">Register here</a></p>` digunakan untuk menampilkan teks dan tautan untuk menuju halaman pendaftaran.



Pada baris [22], syntax `</div>` digunakan untuk menutup div dengan kelas "register-link" dan "login".

Pada baris [23], syntax `</div>` digunakan untuk menutup div dengan kelas "container" dan "login", yang menandakan akhir dari konten utama halaman.

Pada baris [24], syntax `</body>` digunakan untuk menutup elemen `<body>`, menandakan akhir dari konten utama halaman web.

Pada baris [25], syntax `</html>` digunakan untuk menutup elemen `<html>`, menandakan akhir dari dokumen HTML secara keseluruhan.

### **register.php**

Pada baris [1], syntax `<!DOCTYPE html>` digunakan untuk mendefinisikan jenis dokumen HTML yang digunakan, yaitu dokumen HTML5.

Pada baris [2], syntax `<html lang="en">` digunakan untuk menandakan awal dari elemen HTML dan menetapkan bahasa default dokumen ke bahasa Inggris (English).

Pada baris [3], syntax `<head>` digunakan untuk menandai awal dari bagian kepala dokumen HTML yang berisi informasi meta-data dan tautan ke file eksternal seperti CSS atau JavaScript.

Pada baris [4], syntax `<meta charset="UTF-8">` digunakan untuk menginformasikan browser bahwa karakter encoding yang digunakan dalam dokumen adalah UTF-8, yang mendukung karakter internasional.

Pada baris [5], syntax `<title>Register</title>` digunakan untuk menetapkan judul halaman web yang akan ditampilkan di tab browser atau judul jendela.

Pada baris [6], syntax `<link rel="stylesheet" href="http://localhost/Modul6/app/Views/styles.css">` digunakan untuk menghubungkan dokumen dengan file eksternal CSS yang berada di alamat URL tertentu untuk mengatur tata letak dan gaya halaman.

Pada baris [7], syntax `</head>` digunakan untuk menandakan akhir dari bagian kepala dokumen HTML.

Pada baris [8], syntax `<body class="register">` digunakan untuk menetapkan kelas CSS "register" ke elemen `<body>`, yang dapat digunakan untuk mengatur gaya halaman khusus untuk halaman registrasi.

Pada baris [9], syntax `<div class="container register">` digunakan untuk memulai div dengan kelas "container" dan "register", yang umumnya digunakan untuk mengelompokkan elemen-elemen dalam tata letak halaman.

Pada baris [10], syntax `<h1 class="register">Register</h1>` digunakan untuk menampilkan judul halaman dengan teks "Register" dan kelas CSS "register".

Pada baris [11], syntax `<form class="register" action="/register" method="post">` digunakan untuk memulai formulir registrasi yang mengirim data ke `/register` menggunakan metode POST, dengan kelas CSS "register".

Pada baris [12] - [20], PHP script digunakan untuk menampilkan pesan kesalahan menggunakan session Flashdata jika ada, atau pesan sukses jika registrasi berhasil, dengan kelas CSS "alert" dan "register".

Pada baris [21], syntax `<div class="form-group">` digunakan untuk memulai div dengan kelas "form-group", yang umumnya digunakan untuk mengelompokkan input dan label dalam formulir.

Pada baris [27] - [28], syntax `<label for="username">Username:</label>` dan `<input type="text" name="username" id="username" class="register" value="<?= old('username') ?>" required>` digunakan untuk menampilkan input teks untuk username, dengan atribut `required` untuk validasi, dan menggunakan nilai "old" untuk mengembalikan nilai input sebelumnya.

Pada baris [31] - [32], syntax `<label for="email">Email:</label>` dan `<input type="email" name="email" id="email" class="register" value="<?= old('email') ?>" required>` digunakan untuk menampilkan input email, dengan atribut `required` untuk validasi, dan menggunakan nilai "old" untuk mengembalikan nilai input sebelumnya.

Pada baris [35] - [36], syntax `<label for="password">Password:</label>` dan `<input type="password" name="password" id="password" class="register" required>` digunakan untuk menampilkan input password, dengan atribut `required` untuk validasi.

Pada baris [38], syntax `<button type="submit" class="btn register">Register</button>` digunakan untuk menampilkan tombol submit dengan teks "Register" dan kelas CSS "btn" dan "register".

Pada baris [39], syntax `</form>` digunakan untuk menutup elemen formulir.

Pada baris [33], syntax `<div class="login-link register">` digunakan untuk memulai div dengan kelas "login-link" dan "register", yang berisi tautan untuk masuk ke halaman login.

Pada baris [41], syntax `Already have an account? <a href="/login" class="register">Login here</a>` digunakan untuk menampilkan teks dan tautan untuk menuju halaman login dengan kelas CSS "register".

Pada baris [42], syntax `</div>` digunakan untuk menutup div dengan kelas "login-link" dan "register".

Pada baris [43], syntax `</div>` digunakan untuk menutup div dengan kelas "container" dan "register", yang menandakan akhir dari konten utama halaman.

Pada baris [44], syntax `</body>` digunakan untuk menutup elemen `<body>`, menandakan akhir dari konten utama halaman web.

Pada baris [45], syntax `</html>` digunakan untuk menutup elemen `<html>`, menandakan akhir dari dokumen HTML secara keseluruhan.

## **Tautan Git**

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/aisyainind/PRAKTIKUM-PEMROGRAMAN-WEB-II/blob/main/Modul%206.zip>