

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Zulfa Auliya Akbar NIM. 2210817210026

**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 : Zulfa Auliya Akbar
NIM : 2210817210026

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.....	18
C. Pembahasan.....	19
D. Tautan Git	25

DAFTAR GAMBAR

Gambar 1 Screenshot Login	18
Gambar 2 Screenshot Buku	18
Gambar 3 Screenshot book_list.....	18
Gambar 4 Screenshot editbook (book_form)	19
Gambar 5 Screenshot addbook (book_form).....	19

DAFTAR TABEL

Tabel 1 Source Code Routes.php	7
Tabel 2 Source Code AuthController	9
Tabel 3 Source Code BookController	11
Tabel 4 Source Code AuthFilter	12
Tabel 5 Source Code CreateUserTable	13
Tabel 6 Source Code CreateBookTable	14
Tabel 7 Source Code UserModel.....	14
Tabel 8 Source Code UserModel.....	15
Tabel 9 Source Code login	15
Tabel 10 Source Code register.....	16
Tabel 11 Source Code BookForm	17
Tabel 12 Source Code book_list.....	18

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 dalam folder github PRAK 601

A. Source Code

```
1 <?php
2
3 use CodeIgniter\Router\RouteCollection;
4
5 /**
6  * @var RouteCollection $routes
7  */
8 // app/Config/Routes.php
9
10 // User routes
11 $routes->get('/', 'AuthController::login');
12 $routes->match(['get', 'post'], 'register',
13 'AuthController::register');
14 $routes->match(['get', 'post'], 'login',
15 'AuthController::login');
16 $routes->get('logout', 'AuthController::logout');
17
18 $routes->get('books', 'BookController::index', ['filter' =>
19 'auth']);
20 $routes->match(['get', 'post'], 'books/create',
21 'BookController::create', ['filter' => 'auth']);
22 $routes->match(['get', 'post'], 'books/edit/(:num)',
23 'BookController::edit/$1', ['filter' => 'auth']);
24 $routes->get('books/delete/(:num)', 'BookController::delete/$1',
25 ['filter' => 'auth']);
```

Tabel 1 Source Code Routes.php

```
1 <?php
2
3 namespace App\Controllers;
4
5 use App\Models\UserModel;
6 use CodeIgniter\Controller;
7
8 class AuthController extends Controller
9 {
10     public function register()
11     {
12         helper(['form']);
13         $data = [];
14
15         if ($this->request->getMethod() == 'post') {
16             $rules = [
17                 'username' =>
18                 'required|min_length[3]|max_length[20]',
```

```

18         'email'      => 'required|valid_email',
19         'password'    =>
20 'required|min_length[6]|max_length[255]',
21         'password_confirm' => 'matches[password]'
22     ];
23
24     if ($this->validate($rules)) {
25         $model = new UserModel();
26         $newData = [
27             'username'    =>    $this->request-
28 >getPost('username'),
29             'email'        =>    $this->request-
29 >getPost('email'),
30             'password'    =>    password_hash($this-
30 >request->getPost('password'), PASSWORD_DEFAULT),
31         ];
32         $model->save($newData);
33         return redirect()->to('/login');
34     } else {
35         $data['validation'] = $this->validator;
36     }
37 }
38 echo view('register', $data);
39 }
40 public function login()
41 {
42     helper(['form']);
43     $data = [];
44
45     if ($this->request->getMethod() == 'post') {
46         $rules = [
47             'email'      => 'required|valid_email',
48             'password'    =>
49 'required|min_length[6]|max_length[255]',
50         ];
51
52         if (!$this->validate($rules)) {
53             $data['validation'] = $this->validator;
54         } else {
55             $model = new UserModel();
56             $user = $model->where('email', $this-
56 >request->getPost('email'))->first();
57
58             if ($user && password_verify($this-
58 >request->getPost('password'), $user['password'])) {

```


59	session()->set([
60	'id' => \$user['id'],
61	'username' => \$user['username'],
62	'isLoggedIn' => true,
63]);
64	return redirect()->to('/books');
65	} else {
66	\$data['error'] = 'Invalid email or
67	password';
68	}
69	echo view('login', \$data);
70	}
71	
72	
73	
74	public function logout()
75	{
76	session()->destroy();
77	return redirect()->to('/login');
78	}
79	}
80	

Tabel 2 Source Code AuthController

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Models\BookModel;
6	use CodeIgniter\Controller;
7	
8	class BookController extends Controller
9	{
10	public function index()
11	{
12	\$model = new BookModel();
13	\$data['books'] = \$model->findAll();
14	echo view('book_list', \$data);
15	}
16	
17	public function create()
18	{
19	helper(['form']);
20	\$data = [];

```

21
22         if ($this->request->getMethod() == 'post') {
23             $rules = [
24                 'judul' =>
25                 'required|min_length[3]|max_length[255]',
26                 'penulis' =>
27                 'required|min_length[3]|max_length[255]',
28                 'penerbit' =>
29                 'required|min_length[3]|max_length[255]',
30                 'tahun_terbit' =>
31                 'required|numeric|min_length[4]|max_length[4]',
32             ];
33
34             if (!$this->validate($rules)) {
35                 $data['validation'] = $this->validator;
36             } else {
37                 $model = new BookModel();
38
39                 $newData = [
40                     'judul' => $this->request-
41                     >getPost('judul'),
42                     'penulis' => $this->request-
43                     >getPost('penulis'),
44                     'penerbit' => $this->request-
45                     >getPost('penerbit'),
46                     'tahun_terbit' => $this->request-
47                     >getPost('tahun_terbit'),
48                 ];
49                 $model->save($newData);
50                 return redirect()->to('/books');
51             }
52
53             echo view('book_form', $data);
54         }
55
56         public function edit($id)
57         {
58             helper(['form']);
59             $model = new BookModel();
60             $data['book'] = $model->find($id);
61
62             if ($this->request->getMethod() == 'post') {
63                 $rules = [
64                     'judul' =>
65                     'required|min_length[3]|max_length[255]',

```

	<pre> 'penulis' 59 'required min_length[3] max_length[255]', 60 'penerbit' 'required min_length[3] max_length[255]', 61 'tahun_terbit' 'required numeric min_length[4] max_length[4]', 62]; 63 64 if (!\$this->validate(\$rules)) { 65 \$data['validation'] = \$this->validator; 66 } else { 67 \$newData = [68 'id' => \$id, 69 'judul' => \$this->request- >getPost('judul'), 70 'penulis' => \$this->request- >getPost('penulis'), 71 'penerbit' => \$this->request- >getPost('penerbit'), 72 'tahun_terbit' => \$this->request- >getPost('tahun_terbit'), 73]; 74 \$model->save(\$newData); 75 return redirect()->to('/books'); 76 } 77 } 78 79 echo view('book_form', \$data); 80 } 81 82 public function delete(\$id) 83 { 84 \$model = new BookModel(); 85 \$model->delete(\$id); 86 return redirect()->to('/books'); 87 } 88 } </pre>	
--	---	--

Tabel 3 Source Code BookController

1	<?php
2	
3	namespace App\Filters;
4	
5	use CodeIgniter\HTTP\RequestInterface;
6	use CodeIgniter\HTTP\ResponseInterface;
7	use CodeIgniter\Filters\FilterInterface;

```

8
9 class AuthFilter implements FilterInterface
10 {
11     public function before(RequestInterface $request,
12     $arguments = null)
13     {
14         if (!session()->get('logged_in')) {
15             return redirect()->to('/login');
16         }
17
18     public function after(RequestInterface $request,
19     ResponseInterface $response, $arguments = null)
20     {
21         // Do nothing
22     }

```

Tabel 4 Source Code AuthFilter

```

1 <?php
2
3 // app/Database/Migrations/2023-xx-xx-
xxxxxx_CreateUsersTable.php
4 namespace App\Database\Migrations;
5
6 use CodeIgniter\Database\Migration;
7
8 class CreateUsersTable extends Migration
9 {
10     public function up()
11     {
12         $this->forge->addField([
13             'id' => ['type' => 'INT',
14             'auto_increment' => true],
15             'username' => ['type' => 'VARCHAR',
16             'constraint' => '100'],
17             'email' => ['type' => 'VARCHAR',
18             'constraint' => '100'],
19             'password' => ['type' => 'VARCHAR',
20             'constraint' => '255'],
21             'created_at' => ['type' => 'DATETIME', 'null'
22             => true],
23             'updated_at' => ['type' => 'DATETIME', 'null'
24             => true],
25         ]);
26         $this->forge->addKey('id', true);
27         $this->forge->createTable('users');
28     }

```

23	
24	public function down()
25	{
26	\$this->forge->dropTable('users');
27	}
28	}

Tabel 5 Source Code CreateUserTable

1	<?php
2	
3	// app/Database/Migrations/2023-xx-xx-xxxxxx_CreateBooksTable.php
4	namespace App\Database\Migrations;
5	
6	use CodeIgniter\Database\Migration;
7	
8	class CreateBooksTable extends Migration
9	{
10	public function up()
11	{
12	\$this->forge->addField([
13	'id' => ['type' => 'INT',
14	'auto_increment' => true],
15	'judul' => ['type' => 'VARCHAR',
16	'constraint' => '255'],
17	'penulis' => ['type' => 'VARCHAR',
18	'constraint' => '255'],
19	'penerbit' => ['type' => 'VARCHAR',
20	'constraint' => '255'],
21	'tahun_terbit' => ['type' => 'YEAR'],
22	'created_at' => ['type' => 'DATETIME', 'null'
23	=> true],
24	'updated_at' => ['type' => 'DATETIME', 'null'
25	=> true],
26]);
27	\$this->forge->addKey('id', true);
28	\$this->forge->createTable('books');
29	}
30	
31	public function down()
32	{
33	\$this->forge->dropTable('books');
34	}
35	}

Tabel 6 Source Code CreateBookTable

```
1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  class UserModel extends Model
8  {
9      protected $table      = 'users';
10     protected $primaryKey = 'id';
11
12     protected $useAutoIncrement = true;
13
14     protected $returnType     = 'array';
15     protected $useSoftDeletes = false;
16
17     protected $allowedFields = ['username', 'email',
18     'password'];
19
20     protected $useTimestamps = true;
21     protected $createdField   = 'created_at';
22     protected $updatedField   = 'updated_at';
23
24     protected $validationRules     = [];
25     protected $validationMessages = [];
26     protected $skipValidation      = false;
27 }
```

Tabel 7 Source Code UserModel

```
1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  class BookModel extends Model
8  {
9      protected $table      = 'books';
10     protected $primaryKey = 'id';
11
12     protected $useAutoIncrement = true;
13
14     protected $returnType     = 'array';
15     protected $useSoftDeletes = false;
16 }
```

17	protected \$allowedFields = ['judul', 'penulis', 'penerbit', 'tahun_terbit'];
18	protected \$useTimestamps = true;
19	protected \$createdField = 'created_at';
20	protected \$updatedField = 'updated_at';
21	
22	
23	
24	protected \$validationRules = [];
25	protected \$validationMessages = [];
26	protected \$skipValidation = false;
	}

Tabel 8 Source Code UserModel

1	<!DOCTYPE html>
2	<html lang="en">
3	<head>
4	<meta charset="UTF-8">
5	<title>Login</title>
6	</head>
7	<body>
8	<h1>Login</h1>
9	<form action="/login" method="post">
10	<label for="email">Email</label>
11	<input type="email" name="email" id="email">
12	
13	<label for="password">Password</label>
14	<input type="password" name="password" id="password">
15	
16	<input type="submit" value="Login">
17	</form>
18	<?= isset(\$validation) ? \$validation->listErrors() :
19	' ' ?>
20	<?= isset(\$error) ? \$error : ' ' ?>
21	</body>
22	</html>

Tabel 9 Source Code login

1	<!DOCTYPE html>
2	<html lang="en">
3	<head>
4	<meta charset="UTF-8">
5	<title>Register</title>
6	</head>
7	<body>
8	<h1>Register</h1>

```

9      <form action="/register" method="post">
10          <label for="username">Username</label>
11          <input type="text" name="username" id="username"
required>
12          <br>
13          <label for="email">Email</label>
14          <input type="email" name="email" id="email"
required>
15          <br>
16          <label for="password">Password</label>
17          <input type="password" name="password"
id="password" required>
18          <br>
19          <label for="password_confirm">Confirm
Password</label>
20          <input type="password" name="password_confirm"
id="password_confirm" required>
21          <br>
22          <input type="submit" value="Register">
23      </form>
24      <?= \Config\Services::validation()->listErrors(); ?>
25 </body>
26 </html>

```

Tabel 10 Source Code register

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title><?= isset($book) ? 'Edit Book' : 'Add Book'
?></title>
6  </head>
7  <body>
8      <h1><?= isset($book) ? 'Edit Book' : 'Add Book'
?></h1>
9      <form action="<?= isset($book) ?
'/books/edit/' . $book['id'] : '/books/create' ?>"
method="post">
10          <label for="judul">Judul</label>
11          <input type="text" name="judul" id="judul"
value="<?= isset($book) ? $book['judul'] : '' ?>">
12          <br>
13          <label for="penulis">Penulis</label>
14          <input type="text" name="penulis" id="penulis"
value="<?= isset($book) ? $book['penulis'] : '' ?>">
15          <br>
16          <label for="penerbit">Penerbit</label>
17

```


18	<code><input type="text" name="penerbit" id="penerbit"</code>
19	<code>value="<?= isset(\$book) ? \$book['penerbit'] : ' ' ?>"></code>
20	<code>
</code>
21	<code><label for="tahun_terbit">Tahun Terbit</label></code>
22	<code><input type="text" name="tahun_terbit"</code>
23	<code>id="tahun_terbit" value="<?= isset(\$book) ?</code>
24	<code>\$book['tahun_terbit'] : ' ' ?>"></code>
25	<code>
</code>
26	<code><input type="submit" value="<?= isset(\$book) ?</code>
	<code>'Update' : 'Save' ?>"></code>
	<code></form></code>
	<code><?= \Config\Services::validation()->listErrors(); ?></code>
	<code></body></code>
	<code></html></code>

Tabel 11 Source Code BookForm

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code><meta charset="UTF-8"></code>
5	<code><title>Book List</title></code>
6	<code></head></code>
7	<code><body></code>
8	<code><h1>Book List</h1></code>
9	<code>Add Book</code>
10	<code><table></code>
11	<code><thead></code>
12	<code><tr></code>
13	<code><th>Judul</th></code>
14	<code><th>Penulis</th></code>
15	<code><th>Penerbit</th></code>
16	<code><th>Tahun Terbit</th></code>
17	<code><th>Actions</th></code>
18	<code></tr></code>
19	<code></thead></code>
20	<code><tbody></code>
21	<code><?php foreach(\$books as \$book): ?></code>
22	<code><tr></code>
23	<code><td><?= \$book['judul'] ?></td></code>
24	<code><td><?= \$book['penulis'] ?></td></code>
25	<code><td><?= \$book['penerbit'] ?></td></code>
26	<code><td><?= \$book['tahun_terbit'] ?></td></code>
27	<code><td></code>
28	<code><a href="/books/edit/<?=</code>
29	<code>\$book['id'] ?>">Edit</code>
30	<code><a href="/books/delete/<?=</code>
	<code>\$book['id'] ?>">Delete</code>
	<code></td></code>

31	</tr>
32	<?php endforeach; ?>
33	</tbody>
34	</table>
35	</body>
36	</html>

Tabel 12 Source Code book_list

B. Output Program

Login

Email

Password

Gambar 1 Screenshot Login

Register

Username

Email

Password

Confirm Password

Gambar 2 Screenshot Buku

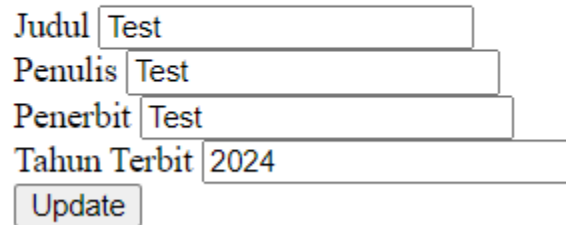
Book List

[Add Book](#)

Judul Penulis Penerbit Tahun Terbit				Actions
Test	Test	Test	2024	Edit Delete

Gambar 3 Screenshot book_list

Edit Book



Judul

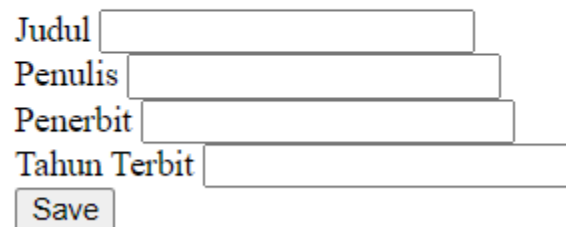
Penulis

Penerbit

Tahun Terbit

Gambar 4 Screenshot editbook (book_form)

Add Book



Judul

Penulis

Penerbit

Tahun Terbit

Gambar 5 Screenshot addbook (book_form)

C. Pembahasan

1. Routes

Pada baris [1] - [3] mengimpor namespace dan class yang digunakan

Pada baris [4] memberikan komentar yang menunjukkan bahwa file ini merupakan konfigurasi routes

Pada baris [7] mendefinisikan route untuk halaman login dengan metode GET dan menunjuk ke `AuthController::login`

Pada baris [8] mendefinisikan route untuk halaman registrasi dengan metode GET dan POST dan menunjuk ke `AuthController::register`

Pada baris [9] mendefinisikan route untuk halaman login dengan metode GET dan POST dan menunjuk ke `AuthController::login`

Pada baris [10] mendefinisikan route untuk logout dengan metode GET dan menunjuk ke `AuthController::logout`

Pada baris [12] mendefinisikan route untuk halaman daftar buku dengan metode GET dan menunjuk ke ``BookController::index`` serta menambahkan filter 'auth' untuk autentikasi

Pada baris [13] mendefinisikan route untuk halaman membuat buku baru dengan metode GET dan POST, menunjuk ke ``BookController::create`` serta menambahkan filter 'auth' untuk autentikasi

Pada baris [14] mendefinisikan route untuk halaman mengedit buku dengan metode GET dan POST, menunjuk ke ``BookController::edit/$1`` di mana \$1 adalah parameter ID buku serta menambahkan filter 'auth' untuk autentikasi

Pada baris [15] mendefinisikan route untuk menghapus buku dengan metode GET, menunjuk ke ``BookController::delete/$1`` di mana \$1 adalah parameter ID buku serta menambahkan filter 'auth' untuk autentikasi

2. AuthController

Pada baris [1] menunjukkan pembukaan blok PHP.

Pada baris [2] mengimpor file `UserModel.php`.

Pada baris [3] mendefinisikan namespace `App\Controllers`.

Pada baris [4] mendeklarasikan kelas `AuthController` yang merupakan turunan dari `CodeIgniter\Controller`.

Pada baris [5] membuka metode `register()`.

Pada baris [6] memanggil helper form.

Pada baris [7] mendefinisikan variabel `$data` tanpa nilai awal.

Pada baris [9] memulai kondisi untuk mengecek metode HTTP request, apakah POST atau tidak.

Pada baris [10] mendefinisikan aturan validasi untuk input form registrasi.

Pada baris [21] memulai kondisi untuk mengecek hasil validasi.

Pada baris [22] membuat instance objek `UserModel`.

Pada baris [23-27] mendefinisikan data baru untuk user berdasarkan input dari form.

Pada baris [28] menyimpan data baru ke dalam database melalui model.

Pada baris [29] melakukan redirect ke halaman login.

Pada baris [30] mengakhiri blok kondisi validasi.

Pada baris [32] menghasilkan pesan validasi jika validasi gagal.

Pada baris [35] menampilkan halaman registrasi dengan menggunakan view `register` dan menyertakan data yang diperlukan.

Pada baris [37] menutup metode `register()`.

Pada baris [39] membuka metode `login()`.

Pada baris [40] memanggil helper form.

Pada baris [41] mendefinisikan variabel `$data` tanpa nilai awal.

Pada baris [43] memulai kondisi untuk mengecek metode HTTP request, apakah POST atau tidak.

Pada baris [44] mendefinisikan aturan validasi untuk input form login.

Pada baris [47] memulai kondisi untuk mengecek hasil validasi.

Pada baris [48] membuat instance objek `UserModel`.

Pada baris [49] mencari user berdasarkan alamat email yang diinputkan.
Pada baris [51-54] melakukan pengecekan password dan mengatur session jika login berhasil.
Pada baris [56-59] menangani kasus ketika login gagal.
Pada baris [62] menampilkan halaman login dengan menggunakan view login dan menyertakan data yang diperlukan.
Pada baris [64] menutup metode login().
Pada baris [66] membuka metode logout().
Pada baris [67] menghapus semua data session.
Pada baris [68] melakukan redirect ke halaman login.
Pada baris [69] menutup metode logout().
Pada baris [70] menutup blok PHP.

3. BookController

Pada baris [1] hingga [6], kode mendefinisikan namespace, menggunakan model BookModel, dan mengimpor kelas Controller dari CodeIgniter.
Pada baris [7], dibuka metode index() yang bertugas untuk menampilkan daftar buku.
Pada baris [8], dibuat instance objek BookModel.
Pada baris [9], data dari semua buku diambil menggunakan metode findAll() dan disimpan dalam array \$data['books'].
Pada baris [10], halaman book_list.php ditampilkan dengan menyertakan data buku.
Pada baris [12], dibuka metode create() yang bertugas untuk menampilkan formulir pembuatan buku.
Pada baris [13], helper form diaktifkan untuk memudahkan penggunaan form.
Pada baris [14], variabel \$data dideklarasikan tanpa nilai awal.
Pada baris [16] hingga [26], dilakukan pengecekan apakah request yang diterima adalah metode POST. Jika ya, aturan validasi untuk input form dibuat.
Pada baris [28] hingga [37], jika validasi gagal, variabel \$data['validation'] akan diisi dengan instance objek validator, sehingga pesan kesalahan dapat ditampilkan di halaman.
Pada baris [39] hingga [49], jika validasi sukses, data baru buku dibuat dan disimpan menggunakan model BookModel, kemudian pengguna akan diarahkan kembali ke halaman daftar buku.
Pada baris [52], metode edit() dibuka dengan parameter \$id yang merepresentasikan ID buku yang akan diubah.

Pada baris [53], seperti pada metode create(), helper form diaktifkan dan variabel \$data dideklarasikan.

Pada baris [54], data buku dengan ID yang sesuai diambil dari model BookModel dan disimpan dalam array \$data['book'].

Pada baris [56] hingga [66], dilakukan pengecekan jika request yang diterima adalah metode POST, aturan validasi untuk form pembuatan buku dibuat. Jika validasi gagal, pesan kesalahan akan ditampilkan di halaman.

Pada baris [68] hingga [77], jika validasi sukses, data buku yang diubah disimpan menggunakan model BookModel, dan pengguna diarahkan kembali ke halaman daftar buku.

Pada baris [80], metode delete() dibuka dengan parameter \$id yang merepresentasikan ID buku yang akan dihapus.

Pada baris [81], objek BookModel dibuat dan metode delete() dipanggil untuk menghapus buku dengan ID yang sesuai.

Pada baris [82], pengguna diarahkan kembali ke halaman daftar buku setelah buku dihapus.

4. AuthFilter

Pada baris [1], kode mendefinisikan namespace App\Filters.

Pada baris [3], kelas AuthFilter dideklarasikan dan mengimplementasikan FilterInterface.

Pada baris [4] hingga [12], metode before() diimplementasikan. Metode ini akan dieksekusi sebelum eksekusi dari controller yang terpengaruh oleh filter ini. Jika sesi pengguna tidak menunjukkan bahwa pengguna telah login, pengguna akan diarahkan kembali ke halaman login menggunakan redirect()->to('/login').

Pada baris [14] hingga [18], metode after() diimplementasikan. Metode ini akan dieksekusi setelah eksekusi dari controller yang terpengaruh oleh filter ini. Namun, dalam kasus ini, metode after() tidak melakukan apa pun.

Pada baris [1] menandakan blok php

Pada baris [2] mengimport file Model.php

Pada baris [3] pengkondisian ketika menghapus data Buku

Pada baris [9]-[39] membuat file html yang isinya adalah table dari data buku dengan memanggil function tampilkan dengan ada tombol button untuk menambah data

5. BookModel

Pada baris [3], kode mendefinisikan namespace App\Models.

Pada baris [5], kelas BookModel dideklarasikan dan merupakan turunan dari kelas Model yang disediakan oleh CodeIgniter.

Pada baris [6] hingga [9], beberapa properti dilakukan pengaturan terkait dengan nama tabel, primary key, dan penggunaan auto-increment untuk primary key. Pada baris [11], properti `$returnType` diatur sebagai 'array', yang menunjukkan bahwa data yang dikembalikan oleh model akan berbentuk array. Pada baris [12], properti `$useSoftDeletes` diatur sebagai false, yang menunjukkan bahwa tidak ada penggunaan fitur soft delete. Pada baris [14], properti `$allowedFields` diatur untuk menentukan kolom mana yang diperbolehkan untuk disimpan. Pada baris [16] hingga [21], properti `$useTimestamps`, `$createdField`, dan `$updatedField` diatur untuk menggunakan timestamps pada saat penciptaan dan pembaruan data, serta menentukan nama kolom yang digunakan untuk menyimpan informasi timestamp. Pada baris [23] hingga [28], properti `$validationRules`, `$validationMessages`, dan `$skipValidation` diatur untuk mengatur validasi data. Namun, dalam kode yang diberikan, properti tersebut tidak diisi sehingga validasi tidak diterapkan.

6. UserModel

Pada baris [3], kode mendefinisikan namespace `App\Models`.

Pada baris [5], kelas `UserModel` dideklarasikan dan merupakan turunan dari kelas `Model` yang disediakan oleh `CodeIgniter`.

Pada baris [6] hingga [9], beberapa properti dilakukan pengaturan terkait dengan nama tabel, primary key, dan penggunaan auto-increment untuk primary key.

Pada baris [11], properti `$returnType` diatur sebagai 'array', yang menunjukkan bahwa data yang dikembalikan oleh model akan berbentuk array.

Pada baris [12], properti `$useSoftDeletes` diatur sebagai false, yang menunjukkan bahwa tidak ada penggunaan fitur soft delete.

Pada baris [14], properti `$allowedFields` diatur untuk menentukan kolom mana yang diperbolehkan untuk disimpan.

Pada baris [16] hingga [21], properti `$useTimestamps`, `$createdField`, dan `$updatedField` diatur untuk menggunakan timestamps pada saat penciptaan dan pembaruan data, serta menentukan nama kolom yang digunakan untuk menyimpan informasi timestamp.

Pada baris [23] hingga [28], properti `$validationRules`, `$validationMessages`, dan `$skipValidation` diatur untuk mengatur validasi data. Namun, dalam kode yang diberikan, properti tersebut tidak diisi sehingga validasi tidak diterapkan.

7. Login

Baris [1]-[4] menetapkan metadata dan style untuk halaman HTML.

Baris [6]-[9] menampilkan judul halaman "Login".

Baris [10]-[15] membuat form HTML untuk pengguna memasukkan email dan password.

Baris [11]-[13] menampilkan input untuk email.

Baris [14]-[16] menampilkan input untuk password.

Baris [17] menampilkan tombol submit dengan label "Login" untuk mengirimkan form.

Baris [18] menampilkan pesan kesalahan validasi jika ada menggunakan PHP ternary operator.

Baris [19] menampilkan pesan kesalahan umum jika ada menggunakan PHP ternary operator

8. Register

Baris [1]-[4] menetapkan metadata dan style untuk halaman HTML.

Baris [6]-[9] menampilkan judul halaman "Register".

Baris [10]-[20] membuat form HTML untuk pengguna memasukkan informasi pendaftaran.

Baris [11]-[13] menampilkan input untuk username.

Baris [14]-[16] menampilkan input untuk email.

Baris [17]-[19] menampilkan input untuk password.

Baris [20]-[22] menampilkan input untuk mengonfirmasi password.

Baris [23] menampilkan tombol submit dengan label "Register" untuk mengirimkan form.

Baris [24] menampilkan pesan kesalahan validasi jika ada menggunakan PHP inline.

9. Book_list

Baris [1]-[4] menetapkan metadata dan style untuk halaman HTML.

Baris [6]-[9] menampilkan judul halaman "Book List".

Baris [10] menampilkan tautan "Add Book" untuk menambahkan buku baru.

Baris [11]-[26] membuat tabel untuk menampilkan daftar buku.

Baris [12]-[16] mendefinisikan header kolom tabel.

Baris [17]-[25] menggunakan loop foreach PHP untuk menampilkan setiap buku dalam array \$books.

Baris [18]-[23] menampilkan data buku, termasuk judul, penulis, penerbit, dan tahun terbit.

Baris [24]-[25] menampilkan tautan "Edit" dan "Delete" untuk setiap buku dengan mengirimkan ID buku ke rute yang sesuai.

10. Book_form

Baris [1]-[4] menetapkan metadata dan style untuk halaman HTML.

Baris [6]-[7] menampilkan judul halaman "Edit Book" jika variabel \$book di-set, dan "Add Book" jika tidak.

Baris [8]-[9] menampilkan judul halaman yang sama dengan baris sebelumnya dalam tag <h1>.

Baris [10]-[15] membuat form untuk menambah atau mengedit buku.

Baris [11]-[18] menampilkan input untuk judul, penulis, penerbit, dan tahun terbit buku, dengan nilai awal yang diambil dari variabel \$book jika tersedia.

Baris [19] menampilkan tombol submit dengan label "Update" jika variabel \$book di-set, dan "Save" jika tidak.

Baris [20] menampilkan pesan kesalahan validasi jika ada menggunakan PHP inline.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/Zetelep/Pemrograman-Web-2/tree/main/MODUL%206>