

**LAPORAN PRAKTIKUM
PEMROGRAMAN II
MODUL 6**



CRUD dan Login pada Codeigniter

Oleh:

Mohammad Zaki Firmansah

NIM. 2210817110006

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN II
MODUL 6

Laporan Praktikum Pemrograman Web II Modul 6: CRUD dan Login pada Codeigniter ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Web II. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Mohammad Zaki Firmansah
NIM : 2210817110006

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

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

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code	6
B. Output Program.....	24
C. Pembahasan.....	26
D. Tautan Git	31

DAFTAR GAMBAR

Gambar 1. Screenshot Tampilan Awal Login	24
Gambar 2. Screenshot Tampilan Awal Setelah Login	24
Gambar 3. Screenshot Tampilan Untuk Menambah Buku	25
Gambar 4. Screenshot Tampilan Edit Buku	25
Gambar 5. Screenshot Tampilan Hapus Data.....	26

DAFTAR TABEL

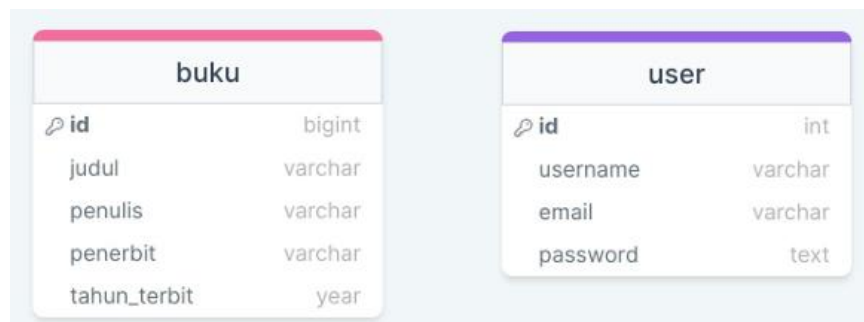
Tabel 1. Source Code AuthController.php	6
Tabel 2. Source Code CRUDController.php	7
Tabel 3. Source Code USER (Migrations).php	9
Tabel 4. Source Code CRUD (Migrations).php	10
Tabel 5. Source Code AuthFilter.php	11
Tabel 6. Source Code BukuModel.php	12
Tabel 7. Source Code UserModel.php.....	13
Tabel 8. Source Code Create.php	14
Tabel 9. Source Code Edit.php	16
Tabel 10. Source Code Index.php	18
Tabel 11. Source Code Login.php	20

SOAL 1

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

Tabel 1. Source Code AuthController.php

AuthController.php	
1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Controllers\BaseController;
6	use App\Models\UserModel;
7	
8	class AuthController extends BaseController
9	{
10	public function index()
11	{
12	return view('login');
13	}

14	
15	public function login()
16	{
17	\$model = new UserModel();
18	\$username = request()->getPost('username');
19	\$email = request()->getPost('email');
20	\$password = request()->getVar('password');
21	
22	\$dataUser = \$model->where(["username" =>
23	\$username]) -> first();
24	if (\$dataUser) {
25	\$pas = \$dataUser['password'];
26	\$authenticatePassword =
27	password_verify(\$password, \$pas);
28	if(\$authenticatePassword) {
29	session() -> set([
30	"username" => \$username,
31	"email" => \$email,
32	"isLoggedIn" => true
33]);
34	return redirect()->to(base_url('/'));
35	} else{
36	session() -> setFlashdata(["pesan" =>
37	"Login gagal, username, email atau password salah"]);
38	return
39	redirect()-
40	>to(base_url('/login'));
41	}
42	} else {
43	session() -> setFlashdata(["pesan" => "Login
44	gagal, username, email atau password salah"]);
45	return redirect()->to(base_url('/login'));
46	}
47	}
48	public function logout()
49	{
50	session()->destroy();
51	return redirect()->to(base_url('login'));
52	}

Tabel 2. Source Code CRUDController.php

CRUDController.php	
1	<?php
2	

```

3 namespace App\Controllers;
4
5 use App\Controllers\BaseController;
6 use App\Models\BukuModel;
7
8 class CRUDController extends BaseController
9 {
10     public function index()
11     {
12
13         $model = new BukuModel();
14         $semuaData = $model->findAll();
15         return view("index", [
16             "data" => $semuaData
17         ]);
18     }
19
20     public function tambah()
21     {
22         return view("create");
23     }
24
25     public function simpan()
26     {
27         $model = new BukuModel();
28         $judul = request()->getPost("judul");
29         $penulis = request()->getPost("penulis");
30         $penerbit = request()->getPost("penerbit");
31         $tahun_terbit = request()->getPost("tahun_terbit");
32
33         $model -> insert([
34             "judul" => $judul,
35             "penulis" => $penulis,
36             "penerbit" => $penerbit,
37             "tahun_terbit" => $tahun_terbit
38         ]);
39         return redirect()->to(base_url('/'));
40     }
41
42     public function edit($id)
43     {
44         $model = new BukuModel();
45         $data = $model-> find($id);
46
47         return view('edit', [
48

```



```

49         "data" => $data
50     });
51 }
52
53 public function update($id)
54 {
55     $model = new BukuModel();
56     $data = [
57         'judul' => $this->request->getPost('judul'),
58         'penulis' => $this->request->getPost('penulis'),
59         'penerbit' => $this->request-
60 >getPost('penerbit'),
61         'tahun_terbit' => $this->request-
62 >getPost('tahun_terbit'),
63     ];
64
65     $model -> update($id, $data);
66     return redirect()->to(base_url('/'));
67 }
68
69 public function hapus($id)
70 {
71     $model = new BukuModel();
72     $model -> delete($id);
73     return redirect()->to(base_url('/'));
74 }
75 }

```

Tabel 3. Source Code USER (Migrations).php

USER (Migrations).php	
1	<?php
2	
3	namespace App\Database\Migrations;
4	
5	use CodeIgniter\Database\Migration;
6	
7	class User extends Migration
8	{
9	public function up()
10	{
11	\$fields = [
12	"id" => [
13	"type" => "INT",
14	"constraint" => 5,
15	"auto_increment" => true,
16],

```

17
18         "username" => [
19             "type" => "VARCHAR",
20             "constraint" =>100,
21         ],
22
23         "email" => [
24             "type" => "VARCHAR",
25             "constraint" =>100,
26         ],
27
28         "password" => [
29             "type" => "TEXT",
30         ],
31     ];
32
33     $this->forge->addField($fields);
34     $this->forge->addKey("id", true);
35     $this->forge->createTable("user");
36 }
37
38 public function down()
39 {
40     $this->forge->dropTable('user');
41 }
42 }

```

Tabel 4. Source Code CRUD (Migrations).php

CRUD (Migrations).php	
1	<?php
2	
3	namespace App\Database\Migrations;
4	
5	use CodeIgniter\Database\Migration;
6	
7	class Crud extends Migration
8	{
9	//Membuat
10	public function up()
11	{
12	\$fields = [
13	"id" => [
14	"type" => "BIGINT",
15	"constraint" => 5,
16	"auto_increment" => true
17],
18	
19	"judul" => [

```

20         "type" => "VARCHAR",
21         "constraint" => 255
22
23     ],
24
25     "penulis" => [
26         "type" => "VARCHAR",
27         "constraint" => 255
28     ],
29
30     "penerbit" => [
31         "type" => "VARCHAR",
32         "constraint" => 255
33     ],
34
35     "tahun_terbit" => [
36         "type" => "YEAR",
37         "constraint" => 255
38     ],
39 ];
40
41 $this->forge->addField($fields);
42 $this->forge->addKey("id", true);
43 $this->forge->createTable("buku");
44 }
45
46 //menghapus
47 public function down()
48 {
49     $this->forge->dropTable("buku", true);
50 }
51 }

```

Tabel 5. Source Code AuthFilter.php

AuthFilter.php	
1	<?php
2	
3	namespace App\Filters;
4	
5	use CodeIgniter\Filters\FilterInterface;
6	use CodeIgniter\HTTP\RequestInterface;
7	use CodeIgniter\HTTP\ResponseInterface;
8	
9	class AuthFilter implements FilterInterface
10	{
11	/**
12	* Do whatever processing this filter needs to do.
13	* By default it should not return anything during

14	* normal execution. However, when an abnormal state
15	* is found, it should return an instance of
16	* CodeIgniter\HTTP\Response. If it does, script
17	* execution will end and that Response will be
18	* sent back to the client, allowing for error pages,
19	* redirects, etc.
20	*
21	* @param RequestInterface \$request
22	* @param array null \$arguments
23	*
24	* @return mixed
25	*/
26	public function before(RequestInterface \$request,
	\$arguments = null)
27	{
28	if(!session()->get('isLoggedIn')) {
29	session() ->setFlashdata('error','Login
	terlebih dahulu!');
30	return redirect()->to(base_url('login'));
31	}
32	}
33	
34	/**
35	* Allows After filters to inspect and modify the
	response
36	* object as needed. This method does not allow any
	way
37	* to stop execution of other after filters, short of
38	* throwing an Exception or Error.
39	*
40	* @param RequestInterface \$request
41	* @param ResponseInterface \$response
42	* @param array null \$arguments
43	*
44	* @return mixed
45	*/
46	public function after(RequestInterface \$request,
	ResponseInterface \$response, \$arguments = null)
47	{
48	//
49	}
50	}

Tabel 6. Source Code BukuModel.php

BukuModel.php	
1	<?php
2	
3	namespace App\Models;

4	
5	use CodeIgniter\Model;
6	
7	class BukuModel extends Model
8	{
9	protected \$DBGroup = 'default';
10	protected \$table = 'buku';
11	protected \$primaryKey = 'id';
12	protected \$useAutoIncrement = true;
13	protected \$returnType = 'array';
14	protected \$useSoftDeletes = false;
15	protected \$protectFields = true;
16	protected \$allowedFields = ["judul", "penulis", "penerbit", "tahun_terbit"];
17	
18	// Dates
19	protected \$useTimestamps = false;
20	protected \$dateFormat = 'datetime';
21	protected \$createdField = 'created_at';
22	protected \$updatedField = 'updated_at';
23	protected \$deletedField = 'deleted_at';
24	
25	// Validation
26	protected \$validationRules = [];
27	protected \$validationMessages = [];
28	protected \$skipValidation = false;
29	protected \$cleanValidationRules = true;
30	
31	// Callbacks
32	protected \$allowCallbacks = true;
33	protected \$beforeInsert = [];
34	protected \$afterInsert = [];
35	protected \$beforeUpdate = [];
36	protected \$afterUpdate = [];
37	protected \$beforeFind = [];
38	protected \$afterFind = [];
39	protected \$beforeDelete = [];
40	protected \$afterDelete = [];
41	}

Tabel 7. Source Code UserModel.php

UserModel.php	
1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	

```

7 class UserModel extends Model
8 {
9     protected $DBGroup          = 'default';
10    protected $table             = 'user';
11    protected $primaryKey        = 'id';
12    protected $useAutoIncrement = true;
13    protected $returnType        = 'array';
14    protected $useSoftDeletes     = false;
15    protected $protectFields     = true;
16    protected $allowedFields      = ["username",
17    "email","password"];
18
19    // Dates
20    protected $useTimestamps = false;
21    protected $dateFormat    = 'datetime';
22    protected $createdField   = 'created_at';
23    protected $updatedField   = 'updated_at';
24    protected $deletedField   = 'deleted_at';
25
26    // Validation
27    protected $validationRules = [];
28    protected $validationMessages = [];
29    protected $skipValidation = false;
30    protected $cleanValidationRules = true;
31
32    // Callbacks
33    protected $allowCallbacks = true;
34    protected $beforeInsert   = [];
35    protected $afterInsert    = [];
36    protected $beforeUpdate   = [];
37    protected $afterUpdate    = [];
38    protected $beforeFind     = [];
39    protected $afterFind      = [];
40    protected $beforeDelete   = [];
41    protected $afterDelete    = [];
42 }

```

Tabel 8. Source Code Create.php

Create.php	
1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">

8	<title>Tambah Data</title>
9	<link
	href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet"
10	integrity="sha384-
	KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy
	+dN9+nJOZ" crossorigin="anonymous">
11	</head>
12	<style>
13	body {
14	background-image:
	url('https://img.freepik.com/free-vector/hand-painted-
	watercolor-abstract-background_23-2148998533.jpg');
15	background-size: cover;
16	background-repeat: no-repeat;
17	background-attachment: fixed;
18	}
19	</style>
20	<body>
21	<div class="container">
22	
23	
24	<div class="row">
25	<div class="col-12">
26	<h2>Tambah Data Buku</h2>
27	</div>
28	
29	<div class="col-3">
30	<form action="<?=
	base_url('/tambahdata')?>" method="post">
31	<div class="mb-3">
32	<label for="judul" class="form-
	label">Judul</label>
33	<input name="judul" type="text"
	class="form-control" id="judul">
34	</div>
35	<div class="mb-3">
36	<label for="penulis"
	class="form-label">Penulis</label>
37	<input type="text" class="form-
	control" id="penulis" name="penulis">
38	</div>
39	<div class="mb-3">
40	<label for="penerbit"
	class="form-label">Penerbit</label>
41	<input type="text" class="form-
	control" id="penerbit" name="penerbit">

42	</div>
43	<div class="mb-3">
45	<label for="tahun_terbit"
	class="form-label">Tahun Terbit</label>
46	<input type="number" min="1800"
	max="2024" class="form-control" id="tahun_terbit"
	name="tahun_terbit">
47	</div>
48	
49	<button type="submit" class="btn
	btn-primary">Tambah</button>
50	<button type="button" class="btn
	btn-primary" onclick="window.location.href='<?=
	base_url('/') ?>'>Kembali</button>
51	</form>
52	</div>
53	
54	</div>
55	<script
	src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0-
	alpha3/dist/js/bootstrap.bundle.min.js"
	integrity="sha384-
	ENjdo4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NT
	KfAdVQSZe" crossorigin="anonymous">
56	</script>
57	</body>
58	</html>

Tabel 9. Source Code Edit.php

Edit.php	
1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta http-equiv="X-UA-Compatible"
	content="IE=edge">
7	<meta name="viewport" content="width=device-width,
	initial-scale=1.0">
8	<title>Edit Data</title>
9	<link
	href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0-
	alpha3/dist/css/bootstrap.min.css" rel="stylesheet"
10	integrity="sha384-
	KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy
	+dN9+nJOZ" crossorigin="anonymous">
11	</head>
12	<style>

13	body {
14	background-image:
	url('https://img.freepik.com/free-vector/hand-painted-watercolor-abstract-background_23-2148998533.jpg');
15	background-size: cover;
16	background-repeat: no-repeat;
17	background-attachment: fixed;
18	}
19	</style>
20	<body>
21	<div class="container">
22	
23	
24	<div class="row">
25	<div class="col-12">
26	<h2>Edit Data Buku</h2>
27	</div>
28	
29	<div class="col-3">
30	<form action="<?=
	base_url('/editdata/' . \$data['id']) ?>" method="post">
31	<div class="mb-3">
32	<label for="judul" class="form-
	label">Judul</label>
33	<input type="text" class="form-
	control" id="judul" value="<?= \$data['judul'] ?>"
	name="judul">
34	</div>
35	<div class="mb-3">
36	<label for="penulis"
	class="form-label">Penulis</label>
37	<input type="text" class="form-
	control" id="penulis" value="<?= \$data['penulis'] ?>"
	name="penulis">
38	</div>
39	<div class="mb-3">
40	<label for="penerbit"
	class="form-label">penerbit</label>
41	<input type="text" class="form-
	control" id="penerbit" value="<?= \$data['penerbit'] ?>"
	name="penerbit">
42	</div>
43	<div class="mb-3">
44	<label for="tahun_terbit"
	class="form-label">Tahun Terbit</label>
45	

46	control" id="tahun_terbit" value="<?=\$data['tahun_terbit'] ?>" name="tahun_terbit">
47	
48	</div>
49	<button type="submit" class="btn btn-primary">Update</button>
50	<button type="button" class="btn btn-primary" onclick="window.location.href='<?='
51	base_url('/') ?>'>Kembali</button>
52	</form>
53	</div>
54	</div>
55	<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
56	ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NT
57	KfAdVQSZe" crossorigin="anonymous">
58	</script>
	</body>
	</html>

Tabel 10. Source Code Index.php

Index.php	
1	<!DOCTYPE html>
2	<html lang="en">
3	<head>
4	<meta charset="UTF-8">
5	<meta http-equiv="X-UA-Compatible" content="IE=edge">
6	<meta name="viewport" content="width=device-width, initial-scale=1.0">
7	<title>Home</title>
8	<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
9	KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy
10	+dN9+nJOZ" crossorigin="anonymous">
11	</head>
12	<style>
13	body { background-image: url('https://img.freepik.com/free-vector/hand-painted-watercolor-abstract-background_23-2148998533.jpg'); background-size: cover;

```

14         background-repeat: no-repeat;
15         background-attachment: fixed;
16     }
17     </style>
18 <body>
19     <div class="container">
20         <div class="pt-5 d-flex justify-content-
between">
21             <div class="col">
22                 <h1>Selamat Datang di database
Perpustakaan</h1>
23                 <p>Selamat Datang di perpustakaan</p>
24             </div>
25             <div class="">
26                 <a href="<?= base_url('/logout')?>"
class="btn btn-danger">Logout</a>
27             </div>
28         </div>
29
30         <!-- create list of books -->
31         <div class="row">
32             <div class="col-12">
33                 <h2>Daftar Buku</h2>
34
35             </div>
36
37             <br><br><br><br>
38
39             <div class="col-12">
40                 <table class="table">
41                     <thead class="table-dark">
42                         <tr>
43                             <th>Judul</th>
44                             <th>Penulis</th>
45                             <th>Penerbit</th>
46                             <th>Tahun Terbit</th>
47                             <th>Aksi</th>
48                         </tr>
49                     </thead>
50                     <tbody>
51                         <?php foreach($data as $row): ?>
52                             <tr>
53                                 <td><?= $row['judul']
?></td>
54                                 <td><?= $row['penulis']
?></td>
55

```

56	<td><td><?= \$row['penerbit'] ?></td></td> <td><td><?=</td>	<td><?= \$row['penerbit'] ?></td>	<td><?=
57	\$row['tahun_terbit'] ?></td>		
58	<td><td> <a href="<?=</td>	<td> <a href="<?=	
59	base_url('/editdata/' . \$row['id']) ?>" class="btn btn-sm btn-warning">Edit		
60	<td><form action="<?=</td>	<form action="<?=	
61	base_url('/hapusdata/' . \$row['id']) ?>" style = "display:inline" method="post"> <input type="hidden" name="_method" value="DELETE"> <button class="btn btn-sm btn-danger" type="submit" onclick="return confirm('Apakah anda yakin?')">hapus</button> </form> </td> </tr> <?php endforeach ?> </tbody> </table> <div class="col-12"> <a class="btn btn-success" href="<?=		
71	base_url('/tambahdata') ?>">Tambah Data		
72	</div>		
73	</div>		
74			
75	</div> <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0- alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384- ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NT KfAdVQSZe" crossorigin="anonymous"></script> </body> </html>		

Tabel 11. Source Code Login.php

Login.php	
1	<!DOCTYPE html>
2	<html lang="en">
3	<head>
4	<meta charset="utf-8">
5	<meta name="author" content="Muhamad Nauval Azhar">
6	

7	<pre> <meta name="viewport" content="width=device- width,initial-scale=1"> <meta name="description" content="This is a login page template based on Bootstrap 5"> <title>Login</title> <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.0- beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384- BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj6ltLrc4wSX0szH/Ev+nYRR uWlolf1fl1" crossorigin="anonymous"> </head> <style> body { background-image: url('https://img.freepik.com/free-vector/hand-painted- watercolor-abstract-background_23-2148998533.jpg'); background-size: cover; background-repeat: no-repeat; background-attachment: fixed; } </style> <body> <section class="h-100"> <div class="container h-100"> <div class="row justify-content-sm-center h-100"> <div class="col-xxl-4 col-xl-5 col- lg-5 col-md-7 col-sm-9"> <div class="text-center my-5"> </div> <?php if(session() -> getFlashdata('pesan')): ?> <div class="alert alert-warning" role="alert"> <?= session()->getFlashdata('pesan') ?> </div> <?php endif?> <?php if(session() -> getFlashdata('error')): ?> </pre>
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	

```

36         <div class="alert alert-warning"
37 role="alert">
38             <?= session()->getFlashdata('error') ?>
39         </div>
40     <?php endif?>
41         <div class="card shadow-lg">
42             <div class="card-body p-
43 5">
44                 <h1 class="fs-4 card-
45 title fw-bold mb-4">Login</h1>
46                 <form action = "<?=
47 base_url('/login')?>" method="POST" class="needs-
48 validation" novalidate="" autocomplete="off">
49                     <div class="mb-3">
50                         <div
51 class="mb-2 w-100">
52                             <label class="text-muted" for="password">User
53 Name</label>
54                             </div>
55                             <input
56 id="user_name" type="text" class="form-control"
57 name="username" required>
58                         </div>
59                         <div
60 class="invalid-feedback">
61                             Username is
62                             required
63                         </div>
64                     </div>
65                     <div class="mb-
66 3">
67                         <label
68 class="mb-2 text-muted" for="email">E-Mail
69 Address</label>
70                         <input
71 id="email" type="email" class="form-control"
72 name="email" value="" required autofocus>
73                     </div>
74                     <div class="invalid-feedback">
75                         Email
76                         is invalid
77                     </div>
78                 </div>
79             </div class="mb-
80 3">
81         </div>

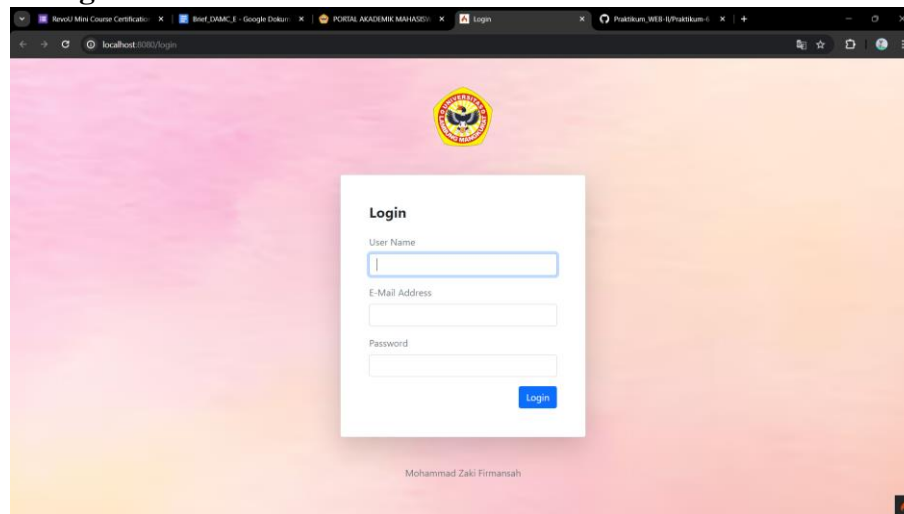
```

```

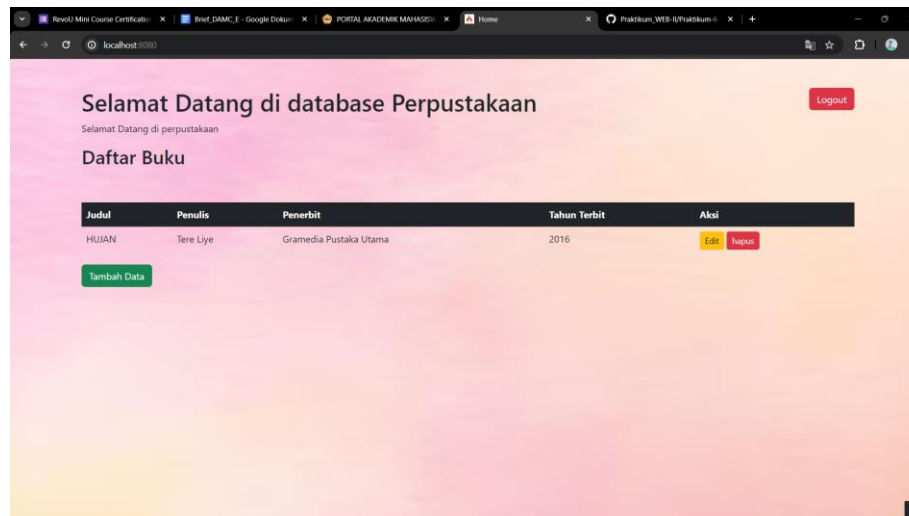
63                                     <div
class="mb-2 w-100">
        <label                                class="text-muted"
64 for="password">Password</label>
                                     </div>
                                     <input
65 id="password"    type="password"    class="form-control"
name="password" required>
66                                     <div
class="invalid-feedback">
67                                     Password is
68 required
69                                     </div>
                                     </div>
70                                     <div    class="d-
flex align-items-center">
71                                     <button
72 type="submit" class="btn btn-primary ms-auto">
73                                     Login
74                                     </button>
75                                     </div>
76                                     </form>
77                                     </div>
                                     </div>
78                                     <div    class="text-center    mt-5
text-muted">
79                                     Copyright    &copy;    2024
80 &mdash; Mohammad Zaki Firmansah
81                                     </div>
82                                     </div>
83                                     </div>
84                                     </div>
85 </section>
</body>
</html>

```

B. Output Program



Gambar 1. Screenshot Tampilan Awal Login



Gambar 2. Screenshot Tampilan Awal Setelah Login

Tambah Data Buku

Judul

Penulis

Penerbit

Tahun Terbit

[Tambah](#) [Kembali](#)

Gambar 3. Screenshot Tampilan Untuk Menambah Buku

Edit Data Buku

Judul

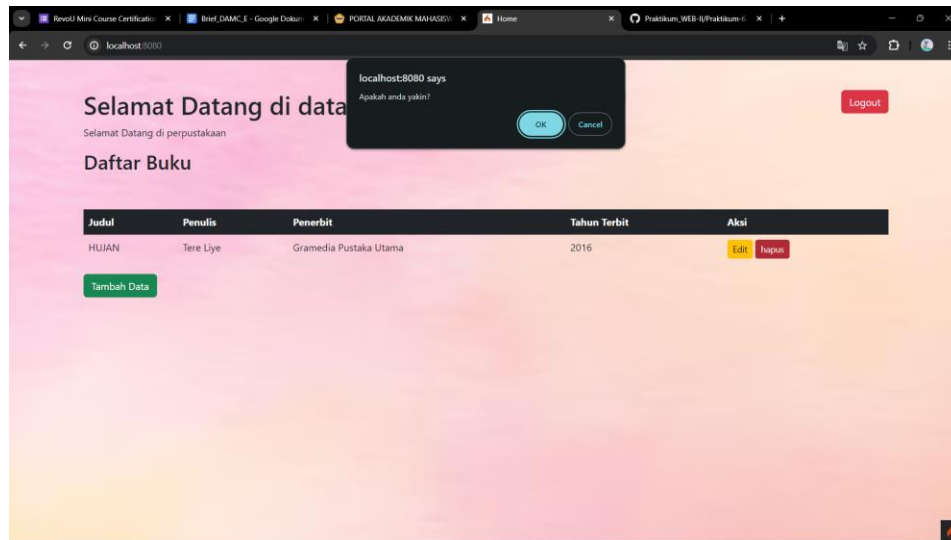
Penulis

penerbit

Tahun Terbit

[Update](#) [Kembali](#)

Gambar 4. Screenshot Tampilan Edit Buku



Gambar 5. Screenshot Tampilan Hapus Data

C. Pembahasan

File AuthController.php

Pada baris [3 – 6], Mendefinisikan namespace dan mengimpor kelas-kelas yang diperlukan.

Pada baris [8 - 18], Mendefinisikan kelas AuthController yang merupakan turunan dari kelas BaseController. Kelas ini berfungsi sebagai pengontrol untuk fitur otentikasi.

Pada baris [20 – 24], Metode index() digunakan untuk menampilkan halaman login. 4. Baris 26-44: Metode login() digunakan untuk memeriksa dan memvalidasi data login yang dikirim oleh pengguna. Data login seperti username, email, dan password diambil dari permintaan yang diterima. Selanjutnya, kelas UserModel digunakan untuk mencari data pengguna berdasarkan username yang diberikan. Jika pengguna ditemukan, metode password_verify() digunakan untuk memeriksa apakah kata sandi yang dimasukkan cocok dengan kata sandi yang disimpan dalam database. Jika cocok, sesi pengguna akan diset dengan informasi seperti username, email, dan status isLoggedIn yang menunjukkan bahwa pengguna berhasil login. Pengguna akan diarahkan kembali ke halaman beranda. Jika kata sandi tidak cocok, pesan kesalahan akan ditampilkan dan pengguna akan diarahkan kembali ke halaman login.

Pada baris [46 - 53], Metode logout() digunakan untuk menghapus sesi pengguna dan mengarahkan pengguna kembali ke halaman login.

File CRUDController.php

Pada baris [3 - 6], Mendefinisikan namespace dan mengimpor kelas-kelas yang diperlukan.

Pada baris [8 - 21], Mendefinisikan kelas CRUDController yang merupakan turunan dari kelas BaseController. Kelas ini berfungsi sebagai pengontrol untuk fitur CRUD pada entitas "Buku".

Pada baris [23 – 35], Metode `index()` digunakan untuk menampilkan halaman indeks yang menampilkan semua data buku. Pertama, objek `$model` dari kelas `BukuModel` dibuat. Kemudian, metode `findAll()` digunakan untuk mengambil semua data buku dari model. Data tersebut kemudian dikirim ke tampilan "index".

Pada baris [37 - 41], Metode `tambah()` digunakan untuk menampilkan halaman formulir tambah buku.

Pada baris [43 – 56], Metode `simpan()` digunakan untuk menyimpan data buku yang diinputkan dari formulir tambah buku. Data yang diinputkan, seperti judul, penulis, penerbit, dan tahun_terbit, diambil menggunakan `request()->getPost()`. Selanjutnya, metode `insert()` dari objek `$model` digunakan untuk menambahkan data buku ke dalam database. Setelah itu, pengguna diarahkan kembali ke halaman indeks.

Pada baris [58 - 70], Metode `edit($id)` digunakan untuk menampilkan halaman formulir edit buku. Data buku yang akan diedit dengan ID yang diberikan diambil menggunakan metode `find()` dari objek `$model`. Data tersebut kemudian dikirim ke tampilan "edit".

Pada baris [72 – 86], Metode `update($id)` digunakan untuk menyimpan perubahan data buku setelah diedit. Data yang diinputkan dari formulir edit buku diambil menggunakan `request()->getPost()`. Kemudian, metode `update()` dari objek `$model` digunakan untuk mengupdate data buku di database berdasarkan ID yang diberikan. Setelah itu, pengguna diarahkan kembali ke halaman indeks.

Pada baris [88 – 92], Metode `hapus($id)` digunakan untuk menghapus data buku berdasarkan ID yang diberikan. Metode `delete()` dari objek `$model` digunakan untuk menghapus data buku dari database. Setelah itu, pengguna diarahkan kembali ke halaman indeks.

File USER (Migrations).php

Pada baris [3 – 11], Mendefinisikan namespace dan mengimpor kelas-kelas yang diperlukan.

Pada baris [13 – 31], Mendefinisikan kelas migrasi User yang merupakan turunan dari kelas Migration. Kelas ini berfungsi untuk membuat tabel user dalam database ketika migrasi dijalankan.

Pada baris [15 – 26], Metode `up()` digunakan untuk membuat tabel user. Tabel ini memiliki beberapa kolom seperti id, username, email, dan password. Setiap kolom didefinisikan menggunakan array dengan konfigurasi seperti tipe data (type) dan batasan (constraint). Kolom id juga diberi opsi `auto_increment` untuk menghasilkan nilai ID secara otomatis. Metode `$this->forge->addField()` digunakan untuk menambahkan kolom-kolom ke dalam definisi tabel. Metode `$this->forge->addKey()` digunakan untuk menetapkan kolom id sebagai kunci utama. Terakhir, metode `$this->forge->createTable()` digunakan untuk membuat tabel user berdasarkan definisi yang telah dibuat.

Pada baris [29 - 31], Metode `down()` digunakan untuk menghapus tabel user. Metode `$this->forge->dropTable()` digunakan untuk menghapus tabel user dari database.

File CRUD (Migrations).php

Pada baris [3 - 11], Mendefinisikan namespace dan mengimpor kelas-kelas yang diperlukan.

Pada baris [13 - 43], Mendefinisikan kelas migrasi Crud yang merupakan turunan dari kelas Migration. Kelas ini berfungsi untuk membuat tabel buku dalam database ketika migrasi dijalankan.

Pada baris [15 - 37], Metode up() digunakan untuk membuat tabel buku. Tabel ini memiliki beberapa kolom seperti id, judul, penulis, penerbit, dan tahun_terbit. Setiap kolom didefinisikan menggunakan array dengan konfigurasi seperti tipe data (type), batasan (constraint), dan opsi tambahan seperti auto_increment untuk kolom id. Metode \$this->forge->addField() digunakan untuk menambahkan kolom-kolom ke dalam definisi tabel. Metode \$this->forge->addKey() digunakan untuk menetapkan kolom id sebagai kunci utama. Terakhir, metode \$this->forge->createTable() digunakan untuk membuat tabel buku berdasarkan definisi yang telah dibuat.

Pada baris [40 - 42], Metode down() digunakan untuk menghapus tabel buku. Metode \$this->forge->dropTable() digunakan untuk menghapus tabel buku dari database. Opsi true yang diberikan pada metode dropTable() menandakan bahwa jika tabel sudah ada, maka akan dihapus terlebih dahulu sebelum dibuat.

File AuthFilter.php

Pada baris [3 - 12], Mendefinisikan namespace dan mengimpor kelas-kelas yang diperlukan.

Pada baris [14 - 37], Mendefinisikan kelas filter AuthFilter yang merupakan turunan dari FilterInterface. Kelas ini memiliki dua metode utama yaitu before() dan after().

- Metode before(): Metode ini akan dijalankan sebelum eksekusi tindakan yang difilter. Di sini digunakan untuk memeriksa apakah pengguna telah login sebelum mengakses halaman tersebut. Jika pengguna belum login, maka akan ditetapkan pesan flash "silahkan login terlebih dahulu" dan pengguna akan diarahkan ke halaman login dengan menggunakan metode redirect()->to(base_url('login')).
- Metode after(): Metode ini akan dijalankan setelah eksekusi tindakan yang difilter. Dalam contoh ini, metode ini tidak memiliki implementasi apa pun.

File BukuModel.php

Pada baris [3 - 12], Mendefinisikan namespace dan mengimpor kelas-kelas yang diperlukan.

Pada baris [14 - 46], Mendefinisikan kelas model BukuModel yang merupakan turunan dari Model. Kelas ini memiliki properti dan metode yang dikonfigurasi untuk berinteraksi dengan tabel buku.

- Properti \$DBGroup: Menetapkan grup database yang digunakan untuk koneksi. Dalam contoh ini, digunakan grup 'default'.
- Properti \$table: Menetapkan nama tabel yang akan digunakan oleh model ini, yaitu 'buku'.

- Properti \$primaryKey: Menetapkan nama kolom kunci utama pada tabel, yaitu 'id'.
- Properti \$useAutoIncrement: Menetapkan apakah kolom kunci utama menggunakan increment otomatis. Dalam contoh ini, diatur sebagai true.
- Properti \$returnType: Menetapkan tipe data yang akan dikembalikan oleh model ini. Dalam contoh ini, diatur sebagai 'array'.
- Properti \$useSoftDeletes: Menetapkan apakah penghapusan lunak (soft delete) diaktifkan atau tidak. Dalam contoh ini, diatur sebagai false.
- Properti \$protectFields: Menetapkan apakah kolom-kolom yang dilindungi harus diatur saat menyimpan atau mengupdate data. Dalam contoh ini, diatur sebagai true.
- Properti \$allowedFields: Menetapkan daftar kolom yang diizinkan untuk disimpan atau diperbarui. Dalam contoh ini, diatur sebagai ["judul", "penulis",

File UserModel.php

Pada baris [3 – 12], Mendefinisikan namespace dan mengimpor kelas-kelas yang diperlukan.

Pada baris [14 – 46], Mendefinisikan kelas model BukuModel yang merupakan turunan dari Model. Kelas ini memiliki properti dan metode yang dikonfigurasi untuk berinteraksi dengan tabel buku.

- Properti \$DBGroup: Menetapkan grup database yang digunakan untuk koneksi. Dalam contoh ini, digunakan grup 'default'.
- Properti \$table: Menetapkan nama tabel yang akan digunakan oleh model ini, yaitu 'buku'.
- Properti \$primaryKey: Menetapkan nama kolom kunci utama pada tabel, yaitu 'id'.
- Properti \$useAutoIncrement: Menetapkan apakah kolom kunci utama menggunakan increment otomatis. Dalam contoh ini, diatur sebagai true.
- Properti \$returnType: Menetapkan tipe data yang akan dikembalikan oleh model ini. Dalam contoh ini, diatur sebagai 'array'.
- Properti \$useSoftDeletes: Menetapkan apakah penghapusan lunak (soft delete) diaktifkan atau tidak. Dalam contoh ini, diatur sebagai false.
- Properti \$protectFields: Menetapkan apakah kolom-kolom yang dilindungi harus diatur saat menyimpan atau mengupdate data. Dalam contoh ini, diatur sebagai true.
- Properti \$allowedFields: Menetapkan daftar kolom yang diizinkan untuk disimpan atau diperbarui. Dalam contoh ini, diatur sebagai ["judul", "penulis", "penerbit", "tahun_terbit"].

File Create.php

- Terdapat empat input field yang harus diisi: judul, penulis, penerbit, dan tahun terbit.
- Input field judul, penulis, dan penerbit menggunakan tipe teks, sedangkan input field tahun terbit menggunakan tipe angka.
- Input field tahun terbit memiliki batasan nilai minimum (1800) dan nilai maksimum (2024).

- Terdapat tombol "tambah" yang digunakan untuk mengirimkan data tugas yang telah diisi ke endpoint /tambahdata menggunakan metode POST.

File Edit.php

- Terdapat empat input field yang mewakili judul, penulis, penerbit, dan tahun terbit tugas yang akan diedit.
- Setiap input field diisi dengan nilai dari data tugas yang akan diedit menggunakan atribut value.
- Pengguna dapat mengubah nilai-nilai tersebut sesuai dengan kebutuhan.
- Terdapat tombol "update" yang akan melakukan operasi pembaruan data saat diklik.
- Form ini memiliki tampilan yang serupa dengan form tambah data sebelumnya, namun diisi dengan nilai-nilai yang ada pada data tugas yang akan diedit.

File Index.php

- Halaman menampilkan judul "Selamat Datang di website Perpustakaan" dan paragraf "Selamat Datang".
- Terdapat tombol "Logout" yang mengarah ke URL /logout dengan menggunakan kelas Bootstrap "btn" dan "btn-danger" untuk memberikan gaya pada tombol.
- Di bawahnya, terdapat judul "Daftar Buku" dan tombol "Tambah Data" yang mengarah ke URL /tambahdata. Tombol ini juga menggunakan kelas Bootstrap untuk gaya tombol "btn" dan "btn-success".
- Terdapat tabel yang menampilkan data buku. Setiap baris pada tabel menampilkan judul, penulis, penerbit, dan tahun terbit buku.
- Setiap baris juga memiliki tombol "Edit" yang mengarah ke URL /editdata/{id} dengan menggunakan kelas Bootstrap "btn", "btn-sm", dan "btn-warning" untuk gaya tombol.
- Setiap baris juga memiliki tombol "hapus" yang menggunakan metode POST untuk menghapus data buku. Tombol ini menggunakan kelas Bootstrap "btn", "btn-sm", dan "btn-danger" untuk gaya tombol. Terdapat juga konfirmasi pada tombol "hapus" menggunakan fungsi confirm().

File Login.php

- Halaman menampilkan judul "Login" di bagian tab atau judul halaman.
- Form login ditampilkan menggunakan layout Bootstrap dengan menggunakan kelas "container w-50" untuk mengatur lebar kontainer dan "form-signin" untuk mengatur tampilan form secara keseluruhan.
- Terdapat tiga input field dalam form yaitu "Username", "Email", dan "Password".
- Setiap input field menggunakan elemen <input> dengan atribut type yang sesuai (text atau password)
- Setiap input field juga menggunakan elemen <label> yang terhubung dengan input field menggunakan atribut for dan id input field yang sama untuk meningkatkan aksesibilitas.

- Terdapat pesan kesalahan yang ditampilkan di atas form jika terdapat pesan kesalahan yang dikirim melalui session menggunakan kelas Bootstrap "alert" dan "alertwarning".
- Tombol "Sign in" digunakan untuk mengirimkan form dengan metode POST ke URL /login. Tombol ini menggunakan kelas Bootstrap "btn", "btn-lg", dan "btn-primary"

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

https://github.com/ZakiFir/Praktikum_WEB-II/tree/main/Praktikum-6