

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

M. Fithra Siroja

NIM. 2210817210005

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN I
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 : M. Fithra Siroja
NIM : 2210817210005

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	22
C. Pembahasan.....	23
D. Tautan git	28

DAFTAR GAMBAR

Gambar 1 ERD Soal 1.....	6
Gambar 2 Output Program Halaman Index Soal Praktikum 6	22
Gambar 3 Output Program Halaman Login Soal Praktikum 6	22
Gambar 4 Output Program Tampilan Hapus Soal Praktikum 6	22
Gambar 5 Output Program Halaman Edit Soal Praktikum 6.....	23
Gambar 6 Output Program Halaman Tambah Buku Soal Praktikum 6.....	23

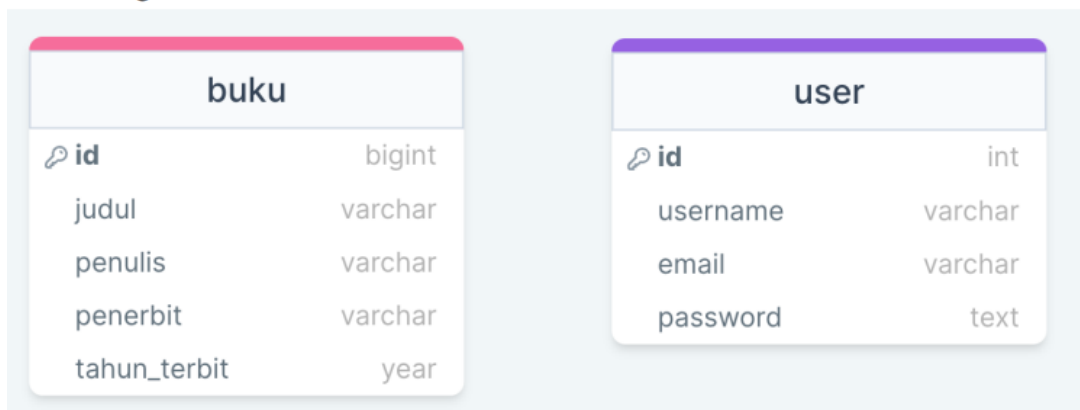
DAFTAR TABEL

Tabel 1 Source Code AuthController.php Soal Praktikum 6.....	7
Tabel 2 Source Code CRUDController.php Soal Praktikum 6.....	9
Tabel 3 Source Code User.php (Migration) Soal Praktikum 6.....	9
Tabel 4 Source Code CRUD.php (Migration) Soal Praktikum 6	10
Tabel 5 Source Code AuthFilter.php Soal Praktikum 6	11
Tabel 6 Source Code UserSeeder.php Soal Praktikum 6.....	12
Tabel 7 Source Code BukuModel.php Soal Praktikum 6	12
Tabel 8 Source Code UserModel.php Soal Praktikum 6	13
Tabel 9 Source Code Create.php Soal Praktikum 6.....	15
Tabel 10 Source Code Edit.php Soal Praktikum 6	17
Tabel 11 Source Code Index.php Soal Praktikum 6	19
Tabel 12 Source Code Login.php Soal Praktikum 6.....	22

SOAL 1

Soal Praktikum:

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



Gambar 1 ERD Soal 1

Simpan dalam folder github PRAK 001

A. Source Code

```
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	\$pass = \$dataUser['password'];
26	\$authenticatePassword =
27	password_verify(\$password, \$pass);
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 redirect()->to(base_url('/login'));;
39	}
40	} else {
41	session() -> setFlashdata(["pesan" => "Login
42	terlebih dahulu!"]);
43	return redirect()->to(base_url('/login'));;
44	}
45	
46	public function logout()
47	{
48	session()->destroy();
49	return redirect()->to(base_url('login'));
50	}
51	}

Tabel 1 Source Code AuthController.php Soal Praktikum 6

CRUController.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             "data" => $data
49         ]);
50     }
51
52     public function update($id)
53     {
54         $model = new BukuModel();
55         $data = [
56             'judul' => $this->request->getPost('judul'),
57             'penulis' => $this->request->getPost('penulis'),
58             'penerbit' => $this->request->getPost('penerbit'),
```


59	<pre> 'tahun_terbit' => \$this->request- >getPost('tahun_terbit'),]; </pre>
60	
61	<pre> \$model -> update(\$id, \$data); return redirect()->to(base_url('/')); </pre>
62	
63	<pre> } </pre>
64	
65	<pre> public function hapus(\$id) { </pre>
66	
67	<pre> \$model = new BukuModel(); </pre>
68	<pre> \$model -> delete(\$id); </pre>
69	<pre> return redirect() ->to(base_url('/')); </pre>
70	<pre> } </pre>
71	<pre> } </pre>
72	
73	
74	

Tabel 2 Source Code CRUDController.php Soal Praktikum 6

User.php(Migration)

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	\$this->forge->addField([
12	'id' => ['type' => 'INT',
13	'auto_increment' => true],
14	'username' => ['type' => 'VARCHAR',
15	'constraint' => '100'],
16	'email' => ['type' => 'VARCHAR',
17	'constraint' => '100'],
18	'password' => ['type' => 'TEXT'],
19]);
20	\$this->forge->addKey('id', true);
21	\$this->forge->createTable('user');
22	}
23	
24	public function down()
25	{
26	\$this->forge->dropTable('user');
27	}
28	}

Tabel 3 Source Code User.php (Migration) Soal Praktikum 6

Crud.php(Migration)

```

1  <?php
2
3  namespace App\Database\Migrations;
4
5  use CodeIgniter\Database\Migration;
6
7  class Crud extends Migration
8  {
9      public function up()
10     {
11         $this->forge->addField([
12             'id' => ['type' => 'BIGINT',
13             'auto_increment' => true],
14             'judul' => ['type' => 'VARCHAR',
15             'constraint' => '255'],
16             'penulis' => ['type' => 'VARCHAR',
17             'constraint' => '255'],
18             'penerbit' => ['type' => 'VARCHAR',
19             'constraint' => '255'],
20             'tahun_terbit' => ['type' => 'YEAR'],
21         ]);
22         $this->forge->addKey('id', true);
23         $this->forge->createTable('buku');
24     }
25
26     public function down()
27     {
28         $this->forge->dropTable('buku');
29     }
30 }

```

Tabel 4 Source Code CRUD.php (Migration) Soal Praktikum 6

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      *
13      * @param RequestInterface $request
14      * @param array|null $arguments
15      *
16      * @return mixed
17      */
18     public function before(RequestInterface $request,

```

	<code>\$arguments = null)</code>
19	<code>{</code>
20	<code> if(!session()->get('isLoggedIn')) {</code>
21	<code> session() ->setFlashdata('error','silahkan</code>
	<code>login terlebih dahulu');</code>
22	<code> return redirect()->to(base_url('login'));</code>
23	<code> }</code>
24	<code>}</code>
25	
26	<code>/**</code>
27	<code> * Allows After filters to inspect and modify the</code>
	<code>response</code>
28	<code> * object as needed. This method does not allow any way</code>
29	<code> * to stop execution of other after filters, short of</code>
30	<code> *</code>
31	<code> * @param RequestInterface \$request</code>
32	<code> * @param ResponseInterface \$response</code>
33	<code> * @param array null \$arguments</code>
34	<code> *</code>
35	<code> * @return mixed</code>
36	<code> */</code>
37	<code>public function after(RequestInterface \$request,</code>
	<code>ResponseInterface \$response, \$arguments = null)</code>
38	<code>{</code>
39	<code> //</code>
40	<code>}</code>
41	<code>}</code>

Tabel 5 Source Code AuthFilter.php Soal Praktikum 6

UserSeeder.php

1	<code><?php</code>
2	
3	<code>namespace App\Database\Seeds;</code>
4	
5	<code>use CodeIgniter\Database\Seeder;</code>
6	<code>use App\Models\UserModel;</code>
7	
8	<code>class UserSeeder extends Seeder</code>
9	<code>{</code>
10	<code> public function run()</code>
11	<code> {</code>
12	<code> \$model = new UserModel();</code>
13	
14	<code> // Memanggil metode insert dengan array asosiatif</code>
15	<code>\$data = [</code>
16	<code> "username" => "fithra",</code>
17	<code> "email" => "fithra@gmail.com",</code>
18	<code> "password" => password_hash("12345678",</code>
	<code>PASSWORD_DEFAULT),</code>
19	<code>];</code>
20	
21	<code> try {</code>
22	<code> \$model->insert(\$data);</code>
23	<code> echo "User inserted successfully.";</code>

24	} catch (\Exception \$e) {
25	echo "Error: " . \$e->getMessage();
26	}
27	}
28	}

Tabel 6 Source Code UserSeeder.php Soal Praktikum 6

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",
17	"penulis", "penerbit", "tahun_terbit"];
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 7 Source Code BukuModel.php Soal Praktikum 6

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 UserModel.php Soal Praktikum 6

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">
      <title>Tambah Data</title>

```

```

8      <link
9      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
      alpha3/dist/css/bootstrap.min.css"          rel="stylesheet"
      integrity="sha384-
      KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN
      9+nJOZ" crossorigin="anonymous">
      <style>
10      body {
11          background-color: #A7E6FF;
12          display: flex;
13          justify-content: center;
14          align-items: center;
15          height: 100vh;
16          margin: 0;
17      }
18      .form-container {
19          background-color: #ffffff;
20          border-radius: 15px;
21          box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
22          padding: 30px;
23          max-width: 500px;
24          width: 100%;
25      }
26      .form-title {
27          margin-bottom: 20px;
28          color: #007bff;
29      }
30  </style>
31  </head>
32
33  <body>
34      <div class="form-container">
35          <h2 class="form-title text-center">Tambah Data
36  Buku</h2>
37          <form action="<?=<br>      base_url('/tambahdata')?>"
      method="post">
38              <div class="mb-3">
39                  <label for="judul" class="form-
      label">Judul</label>
40                  <input name="judul" type="text"
      class="form-control" id="judul" required>
41              </div>
42              <div class="mb-3">
43                  <label for="penulis" class="form-
      label">Penulis</label>
44                  <input type="text" class="form-control"
      id="penulis" name="penulis" required>
45              </div>
46              <div class="mb-3">
47                  <label for="penerbit" class="form-
      label">Penerbit</label>
48                  <input type="text" class="form-control"
      id="penerbit" name="penerbit" required>
49              </div>
          </form>
      </div>
  </body>

```

50	<code><label for="tahun_terbit" class="form-</code>
51	<code>label">Tahun Terbit</label></code>
52	<code><input type="number" min="1800" max="2024"</code>
	<code>class="form-control" id="tahun_terbit" name="tahun_terbit"</code>
	<code>required></code>
	<code></div></code>
53	<code><div class="d-grid"></code>
54	<code><button type="submit" class="btn btn-</code>
55	<code>primary btn-block">Tambah</button></code>
	<code></div></code>
56	<code></form></code>
57	<code></div></code>
58	<code><script</code>
59	<code>src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-</code>
	<code>alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-</code>
	<code>ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfA</code>
	<code>dVQSZe" crossorigin="anonymous"></script></code>
	<code></body></code>
60	<code></html></code>
61	
62	

Tabel 9 Source Code Create.php Soal Praktikum 6

Edit.php

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	
4	<code><head></code>
5	<code><meta charset="UTF-8"></code>
6	<code><meta http-equiv="X-UA-Compatible" content="IE=edge"></code>
	<code><meta name="viewport" content="width=device-width,</code>
7	<code>initial-scale=1.0"></code>
	<code><title>Edit Data</title></code>
8	<code><link</code>
9	<code>href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-</code>
	<code>alpha3/dist/css/bootstrap.min.css" rel="stylesheet"</code>
	<code>integrity="sha384-</code>
	<code>KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN</code>
	<code>9+nJOZ" crossorigin="anonymous"></code>
	<code><style></code>
10	<code>body {</code>
11	<code>background-color: #A7E6FF;</code>
12	<code>display: flex;</code>
13	<code>justify-content: center;</code>
14	<code>align-items: center;</code>
15	<code>height: 100vh;</code>
16	<code>margin: 0;</code>
17	<code>}</code>
18	
19	<code>.form-container {</code>
20	<code>background-color: #ffffff;</code>
21	<code>border-radius: 15px;</code>

```

22         box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
23         padding: 30px;
24         max-width: 500px;
25         width: 100%;
26     }
27
28     .form-title {
29         margin-bottom: 20px;
30         color: #007bff;
31     }
32 </style>
33 </head>
34
35 <body>
36     <div class="form-container">
37         <h2 class="form-title text-center">Edit Data
38 Buku</h2>
39         <form action="<?=
base_url('/editdata/'. $data['id']) ?>" method="post">
40             <div class="mb-3">
41                 <label for="judul" class="form-
label">Judul</label>
42                 <input type="text" class="form-control"
id="judul" value="<?= $data['judul'] ?>" name="judul"
required>
43             </div>
44             <div class="mb-3">
45                 <label for="penulis" class="form-
label">Penulis</label>
46                 <input type="text" class="form-control"
id="penulis" value="<?= $data['penulis'] ?>"
name="penulis" required>
47             </div>
48             <div class="mb-3">
49                 <label for="penerbit" class="form-
label">Penerbit</label>
50                 <input type="text" class="form-control"
id="penerbit" value="<?= $data['penerbit'] ?>"
name="penerbit" required>
51             </div>
52             <div class="mb-3">
53                 <label for="tahun_terbit" class="form-
label">Tahun Terbit</label>
54                 <input type="number" class="form-control"
id="tahun_terbit" value="<?= $data['tahun_terbit'] ?>"
name="tahun_terbit" min="1800" max="2024" required>
55             </div>
56             <div class="d-grid">
57                 <button type="submit" class="btn btn-
primary btn-block">Update</button>
58             </div>
59         </form>
60     </div>
</body>
</html>

```


61	<code>alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZ" crossorigin="anonymous"></script></code>
62	<code></body></code>
63	<code></html></code>
64	

Tabel 10 Source Code Edit.php Soal Praktikum 6

Index.php

1	<code><!DOCTYPE html></code>
2	<code><html lang="en"></code>
3	<code><head></code>
4	<code> <meta charset="UTF-8"></code>
5	<code> <meta http-equiv="X-UA-Compatible" content="IE=edge"></code>
6	<code> <meta name="viewport" content="width=device-width, initial-scale=1.0"></code>
7	<code> <title>Home</title></code>
8	<code> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJ0Z" crossorigin="anonymous"></code>
9	<code> <style></code>
10	<code> body {</code>
11	<code> background-color: #A7E6FF;</code>
12	<code> }</code>
13	<code> .container {</code>
14	<code> margin-top: 50px;</code>
15	<code> }</code>
16	<code> .welcome-card {</code>
17	<code> background-color: #ffffff;</code>
18	<code> border-radius: 10px;</code>
19	<code> box-shadow: 0 4px 8px rgba(0,0,0,0.1);</code>
20	<code> padding: 20px;</code>
21	<code> margin-bottom: 30px;</code>
22	<code> }</code>
23	<code> .table-container {</code>
24	<code> margin-top: 30px;</code>
25	<code> }</code>
26	<code> .table thead {</code>
27	<code> background-color: #343a40;</code>
28	<code> color: #ffffff;</code>
29	<code> }</code>
30	<code> .btn-add {</code>
31	<code> margin-top: 20px;</code>
32	<code> }</code>
33	<code> .btn-logout {</code>
	<code> margin-top: 20px;</code>

```

34     }
35     </style>
36 </head>
37 <body>
38     <div class="container">
39         <div class="welcome-card">
40             <h1>Selamat Datang di Database
41 Perpustakaan</h1>
42             <p>Selamat datang! Berikut adalah daftar buku
yang tersedia.</p>
43             <div class="table-container">
44                 <div class="card">
45                     <div class="card-body">
46                         <table class="table table-hover">
47                             <thead>
48                                 <tr>
49                                     <th>Judul</th>
50                                     <th>Penulis</th>
51                                     <th>Penerbit</th>
52                                     <th>Tahun Terbit</th>
53                                     <th class="text-
center">Aksi</th>
54                             </tr>
55                         </thead>
56                         <tbody>
57                             <?php foreach($data as $row):
?>
58                                 <tr>
59                                     <td><?= $row['judul']
?></td>
60                                     <td><?=
$row['penulis'] ?></td>
61                                     <td><?=
$row['penerbit'] ?></td>
62                                     <td><?=
$row['tahun_terbit'] ?></td>
63                                     <td class="text-
center">
64                                         <a href="<?=
base_url('/editdata/'.$row['id']) ?>" class="btn btn-sm
btn-warning">Edit</a>
65                                         <form action="<?=
base_url('/hapusdata/'.$row['id'])?>"
style="display:inline" method="post">
66                                             <input
type="hidden" name="_method" value="DELETE">
67                                             <button
class="btn btn-sm btn-danger" type="submit"
onclick="return confirm('Apakah anda
yakin?')">Hapus</button>
68                                         </form>
69                                     </td>
70                                 </tr>
71                             <?php endforeach ?>
</tbody>

```



```

14         justify-content: center;
15         align-items: center;
16         margin: 0;
17     }
18     .container {
19         display: flex;
20         justify-content: center;
21         align-items: center;
22         height: 100%;
23     }
24     .card {
25         border-radius: 15px;
26         box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
27         padding: 50px;
28         max-width: 500px;
29         width: 100%;
30     }
31     .btn-primary {
32         background-color: #4caf50;
33         border-color: #4caf50;
34     }
35     .btn-primary:hover {
36         background-color: #45a049;
37         border-color: #45a049;
38     }
39     .card-title {
40         color: #4caf50;
41     }
42     .alert-warning {
43         background-color: #ffcc00;
44         border-color: #ffcc00;
45         color: #333;
46     }
47 </style>
48 </head>
49 <body>
50     <div class="container">
51         <div>
52             <?php if(session() -> getFlashdata('pesan')):
53 ?>
54                 <div class="alert alert-warning"
55 role="alert">
56                     <?= session()->getFlashdata('pesan')
57 ?>
58                 </div>
59                 <?php endif?>
60             <?php if(session() -> getFlashdata('error')):
61 ?>
62                 <div class="alert alert-warning"
63 role="alert">
64                     <?= session()->getFlashdata('error')
65 ?>
66                 </div>

```

	<?php endif?>
63	
64	<div class="card shadow-lg">
65	<div class="card-body p-5">
66	<h1 class="fs-4 card-title fw-bold
67	mb-4">Login</h1>
68	<form action="<?=
	base_url('/login')?>" method="POST" class="needs-
69	validation" novalidate="" autocomplete="off">
	<div class="mb-3">
	<div class="mb-2 w-120">
70	<label class="text-muted"
71	for="user_name">User Name</label>
72	</div>
	<input id="user_name"
73	type="text" class="form-control" name="username"
74	required>
	<div class="invalid-
	feedback">
75	Username is required
	</div>
76	</div>
77	<div class="mb-3">
78	<label class="mb-2 text-muted
79	w-120" for="email">E-Mail Address</label>
80	<input id="email"
	type="email" class="form-control" name="email" value=""
81	required autofocus>
	<div class="invalid-
	feedback">
82	Email is invalid
	</div>
83	</div>
84	
85	<div class="mb-3">
86	<div class="mb-2 w-120">
87	<label class="text-muted"
88	for="password">Password</label>
89	</div>
	<input id="password"
90	type="password" class="form-control" name="password"
91	required>
	<div class="invalid-
	feedback">
92	Password is required
	</div>
93	</div>
94	<div class="d-flex align-items-
95	center">
96	<button type="submit"
	class="btn btn-primary ms-auto">
97	Login
	</button>
98	</div>
99	</form>

```

100         </div>
101     </div>
102 </div>
103 </div>
104 </body>
105 </html>
106
107

```

Tabel 12 Source Code Login.php Soal Praktikum 6

B. Output Program

Gambar 3 Output Program Halaman Login Soal Praktikum 6

Selamat Datang di Database Perpustakaan

Selamat datang! Berikut adalah daftar buku yang tersedia.

Judul	Penulis	Penerbit	Tahun Terbit	Aksi
Dilan 1990	Pidi Baiq	PT. Mizan Pustaka	2018	Seri Hapus
Dilan 1991	Pidi Baiq	PT. Mizan Pustaka	2015	Seri Hapus
Harry Potter and The Sorcerer's Stone	J.K. Rowling	PT. Gramedia Pustaka Utama	2000	Seri Hapus
Harry Potter And The Prisoner of Azkaba	J. K Rowling	PT. Gramedia Pustaka Utama	2008	Seri Hapus
Harry Potter And The Chamber Of Secret	J.K. Rowling	PT. Gramedia Pustaka Utama	2000	Seri Hapus
Harry Potter and The Cursed Child	JK Rowling, John Tiffany, Jack Thorne	PT. Gramedia Pustaka Utama	2018	Seri Hapus

[Logout](#) [Tambah Data](#)

Gambar 2 Output Program Halaman Index Soal Praktikum 6

Selamat Datang di Database Perpustakaan

Selamat datang! Berikut adalah daftar buku yang tersedia.

Judul	Penulis	Penerbit	Tahun Terbit	Aksi
Dilan 1990	Pidi Baiq	PT. Mizan Pustaka	2018	Seri Hapus
Dilan 1991	Pidi Baiq	PT. Mizan Pustaka	2015	Seri Hapus
Harry Potter and The Sorcerer's Stone	J.K. Rowling	PT. Gramedia Pustaka Utama	2000	Seri Hapus
Harry Potter And The Prisoner of Azkaba	J. K Rowling	PT. Gramedia Pustaka Utama	2008	Seri Hapus
Harry Potter And The Chamber Of Secret	J.K. Rowling	PT. Gramedia Pustaka Utama	2000	Seri Hapus
Harry Potter and The Cursed Child	JK Rowling, John Tiffany, Jack Thorne	PT. Gramedia Pustaka Utama	2018	Seri Hapus

[Logout](#) [Tambah Data](#)

Gambar 4 Output Program Tampilan Hapus Soal Praktikum 6

Gambar 5 Output Program Halaman Edit Soal Praktikum 6

Gambar 6 Output Program Halaman Tambah Buku Soal Praktikum 6

C. Pembahasan

AuthController.php

- Namespace dan Penggunaan: Kode dimulai dengan deklarasi namespace App\Controllers yang menunjukkan lokasi file ini dalam struktur aplikasi. Kemudian, menggunakan use untuk mengimpor BaseController dan UserModel.
- Class AuthController: Class ini adalah turunan dari BaseController dan berfungsi sebagai controller untuk proses autentikasi.
- Method index: Method ini mengembalikan view bernama login, yang kemungkinan adalah halaman login untuk aplikasi.
- Method login:
 - Membuat instance dari UserModel yang digunakan untuk berinteraksi dengan tabel pengguna di database.
 - Mengambil nilai username, email, dan password dari form login yang dikirim melalui metode POST.
 - Melakukan query ke database untuk mencari pengguna berdasarkan username.
 - Jika pengguna ditemukan, memverifikasi password yang diinput menggunakan password_verify dengan hash password yang disimpan di database.
 - Jika password benar, membuat session dengan informasi username, email, dan status login (isLoggedIn), lalu mengarahkan pengguna ke halaman utama.
 - Jika password salah, atau pengguna tidak ditemukan, mengatur flash data dengan pesan error dan mengarahkan kembali ke halaman login.

- Method logout:
 - Menghancurkan session untuk mengakhiri sesi pengguna.
 - Mengarahkan pengguna kembali ke halaman login.

CRUDController.php

- Method index:
 - Membuat instance dari BukuModel yang digunakan untuk berinteraksi dengan tabel buku di database.
 - Mengambil semua data buku menggunakan findAll() dan menyimpan hasilnya ke dalam variabel \$semuaData.
 - Mengembalikan view bernama index dengan data buku yang ditemukan.
- Method tambah:
 - Mengembalikan view bernama create, yang mungkin merupakan form untuk menambahkan data buku baru.
- Method simpan:
 - Membuat instance dari BukuModel.
 - Mengambil data dari form yang dikirimkan melalui metode POST (judul, penulis, penerbit, tahun_terbit).
 - Menyimpan data buku baru ke database menggunakan metode insert pada model.
 - Mengarahkan pengguna kembali ke halaman utama setelah data disimpan.
- Method edit:
 - Menerima parameter \$id untuk mengidentifikasi buku yang akan diedit.
 - Mengambil data buku berdasarkan ID menggunakan metode find.
 - Mengembalikan view bernama edit dengan data buku yang ditemukan.
- Method update:
 - Menerima parameter \$id untuk mengidentifikasi buku yang akan diperbarui.
 - Mengambil data baru dari form yang dikirimkan melalui metode POST.
 - Memperbarui data buku di database menggunakan metode update pada model.
 - Mengarahkan pengguna kembali ke halaman utama setelah data diperbarui.
- Method hapus:
 - Menerima parameter \$id untuk mengidentifikasi buku yang akan dihapus.
 - Menghapus data buku dari database menggunakan metode delete pada model.
 - Mengarahkan pengguna kembali ke halaman utama setelah data dihapus.

User.php(Migration)

- Di dalam metode up(), struktur tabel 'user' didefinisikan menggunakan \$this->forge->addField([]), di mana kolom-kolom yang akan ada di tabel ditentukan. Ada empat kolom yang ditentukan: 'id', 'username', 'email', dan 'password'. Setiap kolom memiliki tipe data dan konstrain tertentu. 'id' adalah kolom primary key yang auto increment, 'username' dan 'email' adalah kolom VARCHAR dengan panjang maksimum 100 karakter, sementara 'password' adalah kolom TEXT. Setelah mendefinisikan struktur tabel, primary key juga ditentukan menggunakan \$this->forge->addKey(). Terakhir, tabel 'user' dibuat dengan menggunakan \$this->forge->createTable().

- Di dalam metode `down()`, terdapat kode untuk menghapus tabel 'user' jika diperlukan. Dalam konteks migrasi database, metode `down()` sering digunakan untuk mereset atau mengembalikan perubahan yang telah dilakukan, dalam hal ini, dengan menggunakan `$this->forge->dropTable('user')`, tabel 'user' akan dihapus dari basis data.

Crud.php(Migration)

- Di dalam metode `up()`, kita mendefinisikan struktur tabel 'buku' menggunakan `$this->forge->addField([])`. Ada lima kolom yang didefinisikan: 'id', 'judul', 'penulis', 'penerbit', dan 'tahun_terbit'. 'id' adalah kolom primary key dengan tipe `BIGINT` yang diatur untuk auto increment. Kolom 'judul', 'penulis', dan 'penerbit' adalah kolom `VARCHAR` dengan panjang maksimum 255 karakter. Sedangkan kolom 'tahun_terbit' memiliki tipe `YEAR`, yang menunjukkan tahun penerbitan buku. Setelah mendefinisikan struktur tabel, primary key juga ditentukan menggunakan `$this->forge->addKey()` dan tabel 'buku' dibuat dengan menggunakan `$this->forge->createTable()`.
- Di dalam metode `down()`, terdapat kode untuk menghapus tabel 'buku' jika diperlukan. Seperti yang disebutkan sebelumnya, metode `down()` sering digunakan untuk mereset atau mengembalikan perubahan yang telah dilakukan dalam migrasi database.

AuthFilter.php

- Di dalam metode `before()`, filter ini melakukan pengecekan apakah sesi pengguna (`isLoggedIn`) sudah di-set atau belum. Jika tidak (`!session()->get('isLoggedIn')`), maka filter akan mengatur pesan flashdata yang memberi tahu pengguna untuk login terlebih dahulu dengan menggunakan `session()->setFlashdata()`, kemudian pengguna akan di-redirect ke halaman login dengan `redirect()->to(base_url('login'))`.
- Metode `after()` pada filter ini tidak mengandung implementasi apapun. Secara umum, metode `after()` memungkinkan filter untuk memeriksa dan memodifikasi objek respons jika diperlukan, tetapi dalam kasus ini, tidak ada yang dilakukan setelah respons dihasilkan.

UserSeeder.php

- Di dalam metode `run()`, pertama-tama membuat instance dari model pengguna (`UserModel`). Kemudian, kita mendefinisikan data pengguna yang akan dimasukkan ke dalam basis data dalam bentuk array asosiatif. Data tersebut terdiri dari `username`, `email`, dan `password` yang di-hash menggunakan fungsi `password_hash()` untuk meningkatkan keamanan. Setelah mendefinisikan data, mencoba untuk memasukkan data tersebut ke dalam basis data menggunakan metode `insert()` dari model pengguna. Jika operasi penyisipan berhasil, pesan "User inserted successfully." akan ditampilkan. Namun, jika terjadi kesalahan, pesan error akan ditampilkan bersama dengan pesan kesalahan yang diambil dari pengecualian yang ditangkap.

BukuModel.php

- `$table`: Menentukan nama tabel yang akan digunakan oleh model. Dalam hal ini, model akan berinteraksi dengan tabel 'buku'.
- `$primaryKey`: Menentukan nama kolom primary key dari tabel. Di sini, primary key tabel 'buku' adalah 'id'.
- `$useAutoIncrement`: Menentukan apakah model menggunakan auto-increment untuk primary key atau tidak. Pada kasus ini, auto-increment diaktifkan.
- `$returnType`: Menentukan tipe data yang akan dikembalikan oleh model saat melakukan operasi seperti pencarian. Dalam hal ini, data akan dikembalikan dalam bentuk array.
- `$allowedFields`: Menentukan kolom-kolom mana saja yang diizinkan untuk diisi atau diubah oleh model. Hanya kolom-kolom yang terdaftar di sini yang akan diizinkan.
- `$useTimestamps`, `$dateFormat`, `$createdField`, `$updatedField`, `$deletedField`: Mengatur apakah model menggunakan timestamps (seperti `created_at` dan `updated_at`) dan nama kolom yang akan digunakan untuk timestamp tersebut.
- `$validationRules`, `$validationMessages`, `$skipValidation`: Menentukan aturan validasi untuk data yang dimasukkan atau diubah oleh model. Dalam kode tersebut, validasi tidak diaktifkan.
- `$allowCallbacks`, `$beforeInsert`, `$afterInsert`, `$beforeUpdate`, `$afterUpdate`, `$beforeFind`, `$afterFind`, `$beforeDelete`, `$afterDelete`: Menentukan apakah dan kapan callback akan dipanggil sebelum atau setelah operasi tertentu (seperti insert, update, atau delete) dilakukan oleh model.

UserModel.php

- `$DBGroup`: Menentukan grup basis data yang akan digunakan oleh model. Dalam hal ini, grup basis data yang digunakan adalah 'default'.
- `$table`: Menentukan nama tabel yang akan digunakan oleh model. Dalam hal ini, model akan berinteraksi dengan tabel 'User'.
- `$primaryKey`: Menentukan nama kolom primary key dari tabel. Di sini, primary key tabel 'User' adalah 'id'.
- `$useAutoIncrement`: Menentukan apakah model menggunakan auto-increment untuk primary key atau tidak. Pada kasus ini, auto-increment diaktifkan.
- `$returnType`: Menentukan tipe data yang akan dikembalikan oleh model saat melakukan operasi seperti pencarian. Dalam hal ini, data akan dikembalikan dalam bentuk array.
- `$allowedFields`: Menentukan kolom-kolom mana saja yang diizinkan untuk diisi atau diubah oleh model. Hanya kolom-kolom yang terdaftar di sini yang akan diizinkan.
- `$useTimestamps`, `$dateFormat`, `$createdField`, `$updatedField`, `$deletedField`: Mengatur apakah model menggunakan timestamps (seperti `created_at` dan `updated_at`) dan nama kolom yang akan digunakan untuk timestamp tersebut.
- `$validationRules`, `$validationMessages`, `$skipValidation`: Menentukan aturan validasi untuk data yang dimasukkan atau diubah oleh model. Dalam kode tersebut, validasi tidak diaktifkan.

- \$allowCallbacks, \$beforeInsert, \$afterInsert, \$beforeUpdate, \$afterUpdate, \$beforeFind, \$afterFind, \$beforeDelete, \$afterDelete: Menentukan apakah dan kapan callback akan dipanggil sebelum atau setelah operasi tertentu (seperti insert, update, atau delete) dilakukan oleh model.

Create.php

- DOCTYPE dan Head: Deklarasi DOCTYPE yang menunjukkan bahwa dokumen adalah dokumen HTML5, diikuti oleh tag <head>. Di dalam <head>, meta tag ditentukan untuk mengatur karakter set, kompatibilitas tampilan, dan viewport untuk responsifitas, kemudian judul halaman ditentukan. Selain itu, tautan ke berkas CSS Bootstrap juga disertakan untuk memperindah tampilan halaman.
- 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

Edit.php

- DOCTYPE dan Head: Seperti pada umumnya, mendefinisikan tipe dokumen HTML, pengaturan karakter, dan viewport untuk responsifitas. Juga termasuk judul halaman dan tautan ke berkas CSS Bootstrap.
- Body: Bagian utama dari halaman, dengan latar belakang warna biru muda dan konten halaman yang ditengahkan secara vertikal dan horizontal
- Form Container: Kontainer utama yang memiliki tata letak yang mirip dengan halaman tambah data buku.
- Form: Formulir ini memiliki aksi yang diarahkan ke base_url('/editdata/'.\$data['id']) dengan metode POST. Ini berarti ketika formulir dikirim, data akan dikirimkan ke URL tersebut untuk diedit. Di dalam formulir, ada empat input: Judul, Penulis, Penerbit, dan Tahun Terbit. Setiap input memiliki nilai awal yang diambil dari data buku yang sedang diedit, yang disediakan melalui variabel \$data. Semua input diberi atribut required untuk memastikan bahwa data yang diperlukan tidak kosong sebelum formulir dikirimkan.
- Button Update: Tombol submit untuk mengirimkan formulir, memberikan tampilan yang konsisten dan responsif dengan kelas Bootstrap.
- Script: Tautan ke berkas JavaScript Bootstrap untuk mengaktifkan fitur-fitur Bootstrap yang diperlukan.

Indeks.php

- DOCTYPE dan Head: Mendefinisikan tipe dokumen HTML dan mengimpor berkas CSS Bootstrap untuk styling.
- Body: Konten utama halaman, dengan latar belakang biru muda.
- Container: Digunakan untuk menempatkan konten halaman agar terpusat dan terlihat baik pada berbagai ukuran layar.
- Welcome Card: Sebuah kartu sambutan yang menampilkan pesan selamat datang dan instruksi singkat kepada pengguna. Ini memberikan tampilan yang menarik dengan latar belakang putih, border-radius, dan bayangan yang lembut.
- Tabel Daftar Buku: Tabel yang menampilkan informasi buku seperti judul, penulis, penerbit, dan tahun terbit. Setiap buku direpresentasikan sebagai satu baris dalam tabel. Terdapat juga kolom aksi yang berisi tombol untuk mengedit dan menghapus setiap buku.
- Tombol Edit: Tombol "Edit" yang terletak di dalam kolom aksi memungkinkan pengguna untuk mengedit detail buku tertentu. Tombol ini akan mengarahkan pengguna ke halaman edit yang sesuai dengan buku yang dipilih.
- Form Hapus: Setiap buku juga dilengkapi dengan formulir untuk menghapusnya. Formulir ini mengirimkan permintaan penghapusan ke URL yang sesuai dengan ID buku yang dipilih.
- Tombol Logout: Tombol "Logout" yang memungkinkan pengguna untuk keluar dari aplikasi. Ini memberikan aksi logout dengan mengarahkan pengguna ke halaman logout.
- Tombol Tambah Data: Tombol "Tambah Data" yang memungkinkan pengguna untuk menambahkan buku baru ke dalam daftar. Ini mengarahkan pengguna ke halaman tambah data buku.

Login.php

- DOCTYPE dan Head: Mendefinisikan tipe dokumen HTML, metadata seperti pengarang, viewport untuk responsifitas, dan deskripsi halaman. Juga termasuk tautan ke berkas CSS Bootstrap untuk styling.
- Body: Konten utama halaman, dengan latar belakang biru muda dan diatur untuk ditengahkan secara vertikal dan horizontal pada layar.
- Container: Digunakan untuk menempatkan konten halaman agar terpusat dan terlihat baik pada berbagai ukuran layar.
- Flash Messages: Pesan flash disertakan untuk menampilkan pesan kesalahan atau informasi setelah tindakan pengguna seperti gagal login atau validasi formulir.
- Card Form: Formulir login ditempatkan dalam sebuah kartu (card) dengan bayangan dan padding yang memberikan tampilan menarik. Formulir ini memiliki input untuk username, email, dan password, serta tombol untuk melakukan login.
- Styling: Beberapa gaya CSS ditambahkan untuk memberikan tampilan yang estetik, termasuk perubahan warna pada tombol dan pengaturan warna pada pesan peringatan.

D. Tautan git

<https://github.com/mhmmdfithra/PRAKTIKUM6>