

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Hari Octavian Delrossi NIM. 2210817210033

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 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 : Hari Octavian Delrossi
NIM : 2210817210033

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 PRAKTIKUM	6
A. Source Code	6
B. Output Program.....	25
C. Pembahasan.....	28
D. Tautan Git	33

DAFTAR GAMBAR

Gambar 1. Screenshot halaman Login.....	25
Gambar 2. Screenshot halaman Register.....	25
Gambar 3. Screenshot halaman Dashboard.....	26
Gambar 4. Screenshot halaman Daftar Buku	26
Gambar 5. Screenshot halaman Tambah Buku	27
Gambar 6. Screenshot halaman Edit Buku.....	27
Gambar 7. Screenshot halaman Hapus Buku	28
Gambar 8. Skema Database.....	28

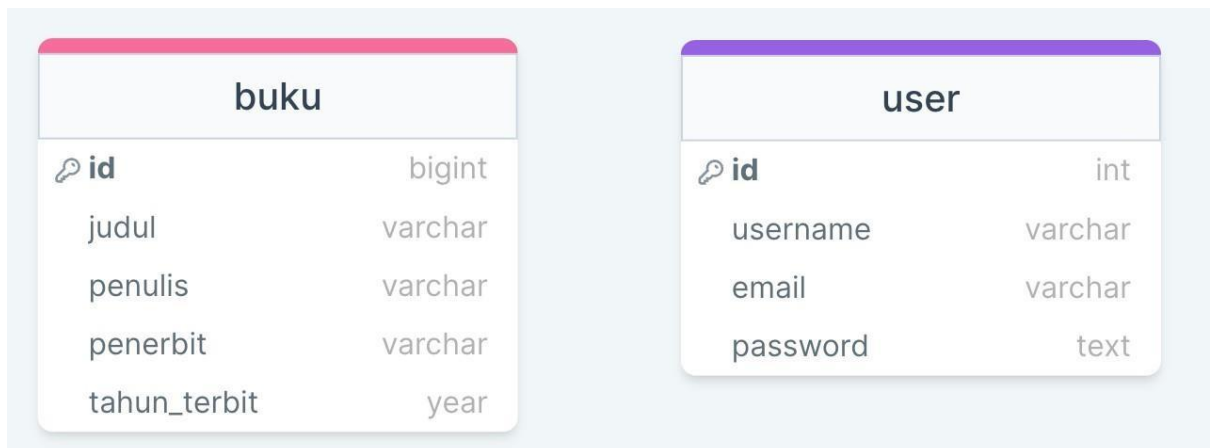
DAFTAR TABEL

Tabel 1. Source Code Hasil Jawaban Routes.php	7
Tabel 2. Source Code Hasil Jawaban LoginController.php	9
Tabel 3. Source Code Hasil Jawaban RegisterController.php.....	11
Tabel 4. Source Code Hasil Jawaban LogoutController.php	11
Tabel 5. Source Code Hasil Jawaban BookController.php	16
Tabel 6. Source Code Hasil Jawaban Auth.php	17
Tabel 7. Source Code Hasil Jawaban login.php	18
Tabel 8. Source Code Hasil Jawaban create.php.....	20
Tabel 9. Source Code Hasil Jawaban edit.php	22
Tabel 10. Source Code Hasil Jawaban index.php	23
Tabel 11. Source Code Hasil Jawaban UserModel.php	23
Tabel 12. Source Code Hasil Jawaban BookModel.php	24

SOAL PRAKTIKUM

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

- Routes.php

```
1 <?php
2
3 use CodeIgniter\Router\RouteCollection;
4
5 /**
6  * @var RouteCollection $routes
7  */
8
```

9	\$routes->get('/', 'LoginController::index');
10	
11	\$routes->get('/login', 'LoginController::index');
12	\$routes->post('/login', 'LoginController::login');
13	
14	\$routes->get('/register', 'RegisterController::index');
15	\$routes->post('/register', 'RegisterController::store');
16	
17	\$routes->get('/logout', 'LoginController::logout');
18	
19	\$routes->get('/dashboard', 'Dashboard::index', ['filter' => 'auth']);
20	
21	\$routes->group('books', ['filter' => 'auth'], function(\$routes) {
22	\$routes->get('/', 'BookController::index');
23	\$routes->get('create', 'BookController::create');
24	\$routes->post('store', 'BookController::store');
25	\$routes->get('edit/(:segment)', 'BookController::edit/\$1');
26	\$routes->post('update/(:segment)', 'BookController::update/\$1');
27	\$routes->get('delete/(:segment)', 'BookController::delete/\$1');
28	});

Tabel 1. Source Code Hasil Jawaban Routes.php

- **LoginController.php**

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Models\UserModel;
6	
7	class LoginController extends BaseController
8	{
9	protected \$model;
10	
11	public function __construct()
12	{

```
13         $this->model = new UserModel();
14         helper(['form', 'url']);
15     }
16
17     public function index()
18     {
19         if ($this->isLoggedIn()) {
20             return redirect()-
21 >to(base_url('dashboard'));
22         }
23
24         $data = [
25             'title' => 'Login | CRUD dan Login pada
26 Codeigniter @hari.oct_'
27         ];
28
29         return view('auth/login', $data);
30     }
31
32     public function login()
33     {
34         $data = $this->request->getPost(['email',
35 'password']);
36
37         if (! $this->validateData($data, [
38             'email' => 'required|valid_email',
39             'password' => 'required'
40         ])) {
41             return $this->index();
42         }
43
44         $email = $this->request->getPost('email');
45         $password = $this->request-
46 >getPost('password');
47
48         $credentials = ['email' => $email];
49
50         $user = $this->model->where($credentials)-
51 >first();
```


49	<pre> if (! \$user !password_verify(\$password, \$user['password'])) { session()->setFlashdata('error', 'Email atau password anda salah.');</pre>
50	
51	<pre> return redirect()->back()->withInput();</pre>
52	<pre> }</pre>
53	
54	<pre> \$userData = [</pre>
55	<pre> 'name' => \$user['name'],</pre>
56	<pre> 'email' => \$user['email'],</pre>
57	<pre> 'logged_in' => TRUE</pre>
58	<pre>];</pre>
59	
60	<pre> session()->set(\$userData);</pre>
61	<pre> return redirect()->to(base_url('dashboard'));</pre>
62	<pre> }</pre>
63	
64	<pre> public function logout()</pre>
65	<pre> {</pre>
66	<pre> session()->destroy();</pre>
67	<pre> return redirect()->to(base_url('login'));</pre>
68	<pre> }</pre>
69	
70	<pre> private function isLoggedIn(): bool</pre>
71	<pre> {</pre>
72	<pre> return session()->get('logged_in') ? true :</pre>
73	<pre> false;</pre>
	<pre> }</pre>

Tabel 2. Source Code Hasil Jawaban LoginController.php

- **RegisterController.php**

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Models\UserModel;
6	
7	class RegisterController extends BaseController
8	{
9	protected \$model;

```
10
11     public function __construct()
12     {
13         $this->model = new UserModel();
14         $this->helpers = ['form', 'url'];
15     }
16
17     public function index()
18     {
19         $data = [
20             'title' => 'Register | CRUD dan Login pada
Codeigniter @hari.oct_'
21         ];
22
23         return view('auth/register', $data);
24     }
25
26     public function store()
27     {
28         $data = $this->request->getPost(['name',
'email', 'password']);
29
30         if (! $this->validate([
31             'name' => 'required',
32             'email' =>
'required|valid_email|is_unique[users.email]',
33             'password' => 'required|min_length[6]'
34         ])) {
35             return $this->index();
36         }
37
38         $save = $this->model->save($data);
39
40         if ($save) {
41             session()->setFlashdata('success',
'Register Berhasil!');
42             return redirect()->to(base_url('login'));
43         } else {
44             session()->setFlashdata('error', 'Gagal
menyimpan data. ');
45             return redirect()->back()->withInput();

```

46	}
47	}
48	}

Tabel 3. Source Code Hasil Jawaban RegisterController.php

- LogoutController.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	class LogoutController extends BaseController
6	{
7	public function index()
8	{
9	
10	\$userData = [
11	'name',
12	'email',
13	'logged_in'
14];
15	
16	session()->remove(\$userData);
17	
18	return redirect()->to(base_url('login'));
19	
20	}
21	}

Tabel 4. Source Code Hasil Jawaban LogoutController.php

- BookController.php

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	protected \$model;
11	

```
12     public function __construct()
13     {
14         $this->model = new BookModel();
15         helper(['form', 'url']);
16     }
17
18     public function index()
19     {
20         $data['books'] = $this->model->findAll();
21         return view('books/index', $data);
22     }
23
24     public function create()
25     {
26         return view('books/create');
27     }
28
29     public function store()
30     {
31         $validation = \Config\Services::validation();
32
33         $validation->setRules([
34             'judul' => 'required|string',
35             'penulis' => 'required|string',
36             'penerbit' => 'required|string',
37             'tahun_terbit' =>
38             'required|integer|greater_than[1800]|less_than[2024]',
39             [
40                 'judul' => [
41                     'required' => 'Judul harus diisi.',
42                     'string' => 'Judul harus berupa
43                     string.',
44                 ],
45                 'penulis' => [
46                     'required' => 'Penulis harus diisi.',
47                     'string' => 'Penulis harus berupa
48                     string.',
49                 ],
50                 'penerbit' => [
51                     'required' => 'Penerbit harus diisi.',
```

50	<pre> 'string' => 'Penerbit harus berupa string.',], 'tahun_terbit' => ['required' => 'Tahun terbit harus 53 diisi.', 'integer' => 'Tahun terbit harus 54 berupa angka.', 'greater_than' => 'Tahun terbit harus 55 lebih besar dari 1800.', 'less_than' => 'Tahun terbit harus 56 lebih kecil dari 2024.',],]); 57 58 59 if (!\$validation->withRequest(\$this->request)- 60 >run()) { return redirect()->back()->withInput()- 61 >with('errors', \$validation->getErrors()); 62 } 63 64 \$data = ['judul' => \$this->request- 65 >getPost('judul'), 'penulis' => \$this->request- 66 >getPost('penulis'), 'penerbit' => \$this->request- 67 >getPost('penerbit'), 'tahun_terbit' => \$this->request- 68 >getPost('tahun_terbit'), 69]; 70 71 if (\$this->model->insert(\$data)) { session()->setFlashdata('success', 'Buku 72 berhasil ditambahkan.');</pre>
73	<pre> return redirect()->to('/books');</pre>
74	<pre> } else { session()->setFlashdata('error', 'Terjadi 75 kesalahan. Silakan coba lagi.');</pre>
76	<pre> return view('books/create');</pre>
77	<pre> } } }</pre>

```

78     }
79
80     public function edit($id)
81     {
82         $data['book'] = $this->model->find($id);
83         return view('books/edit', $data);
84     }
85
86     public function update($id)
87     {
88         $validation = \Config\Services::validation();
89
90         $validation->setRules([
91             'judul' => 'required|string',
92             'penulis' => 'required|string',
93             'penerbit' => 'required|string',
94             'tahun_terbit' =>
95             'required|integer|greater_than[1800]|less_than[2024]',
96             ], [
97                 'judul' => [
98                     'required' => 'Judul harus diisi.',
99                     'string' => 'Judul harus berupa
100 string.',
101                 ],
102                 'penulis' => [
103                     'required' => 'Penulis harus diisi.',
104                     'string' => 'Penulis harus berupa
105 string.',
106                 ],
107                 'penerbit' => [
108                     'required' => 'Penerbit harus diisi.',
109                     'string' => 'Penerbit harus berupa
110 string.',
111                 ],
112                 'tahun_terbit' => [
113                     'required' => 'Tahun terbit harus
114                     diisi.',
115                     'integer' => 'Tahun terbit harus
116                     berupa angka.',
117                     'greater_than' => 'Tahun terbit harus
118                     lebih besar dari 1800.',

```

```

112         'less_than' => 'Tahun terbit harus
lebih kecil dari 2024.',
113     ],
114 ];
115
116     if (!$validation->withRequest($this->request)-
>run()) {
117         return redirect()->back()->withInput()-
>with('errors', $validation->getErrors());
118     }
119
120     $data = [
121         'judul' => $this->request-
>getPost('judul'),
122         'penulis' => $this->request-
>getPost('penulis'),
123         'penerbit' => $this->request-
>getPost('penerbit'),
124         'tahun_terbit' => $this->request-
>getPost('tahun_terbit'),
125     ];
126
127     if ($this->model->update($id, $data)) {
128         session()->setFlashdata('success', 'Buku
berhasil diperbarui.');
```

```

129         return redirect()->to('/books');
```

```

130     } else {
131         session()->setFlashdata('error', 'Terjadi
kesalahan. Silakan coba lagi.');
```

```

132         return view('books/edit', ['book' =>
$this->model->find($id)]);
133     }
134 }
135
136 public function delete($id)
137 {
138     $this->model->delete($id);
139     session()->setFlashdata('success', 'Buku
berhasil dihapus.');
```

```

140     return redirect()->to('/books');
```

```

141 }

```

	}
--	---

Tabel 5. Source Code Hasil Jawaban BookController.php

- Auth.php

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 Auth implements FilterInterface
10	{
11	public function before(RequestInterface \$request,
12	\$arguments = null)
13	{
14	\$excludedRoutes = [
15	'login',
16	'register'
17];
18	\$currentURI = \$request->getUri()->getPath();
19	log_message('debug', 'Current URI: ' .
20	\$currentURI);
21	if (in_array(\$currentURI, \$excludedRoutes)) {
22	return;
23	}
24	
25	if (!session()->get('logged_in')) {
26	log_message('debug', 'User not logged in,
27	redirecting to login.');
28	return redirect()->to('/login')-
29	>with('error', 'Login terlebih dahulu!');
30	}
31	public function after(RequestInterface \$request,
32	ResponseInterface \$response, \$arguments = null)
33	{

33	
34	}
35	}

Tabel 6. Source Code Hasil Jawaban Auth.php

- login.php

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

29	<pre> <label for="email">Email</label> <input type="email" 30 name="email" class="form-control" required> 31 </div> 32 <div class="form-group"> <label 33 for="password">Password</label> <input type="password" 34 name="password" class="form-control" required> 35 </div> 36 <div class="form-group"> <button class="btn btn- 37 primary">Login</button> </div> 38 <?= form_close(); ?> 39 40 41 </div> 42 43 </div> 44 <div class="text-center mt-2"> Belum punya akun? <a href="<?php echo 45 base_url('register'); ?>">Silakan daftar. 46 </div> 47 </div> 48 </div> 49 </div> 50 <?= \$this->endSection() ?> </pre>
----	--

Tabel 7. Source Code Hasil Jawaban login.php

- create.php

1	<pre><?= \$this->extend('layouts/main') ?></pre>
2	
3	<pre><?= \$this->section('content') ?></pre>
4	
5	<pre><div class="container mt-5"></pre>
6	<pre> <div class="d-flex justify-content-between align- items-center mb-3"></pre>
7	<pre> <a href="<?= base_url('books') ?>" class="btn btn-secondary">Kembali</pre>

```

8         <h2>Tambah Buku Baru</h2>
9     </div>
10
11     <?php if (session()->getFlashdata('errors')): ?>
12         <div class="alert alert-danger">
13             <ul>
14                 <?php foreach (session()-
15 >getFlashdata('errors') as $error): ?>
16                     <li><?= $error ?></li>
17                 <?php endforeach ?>
18             </ul>
19         </div>
20     <?php endif ?>
21
22     <form action="<?= base_url('books/store') ?>"
method="post">
23         <?= csrf_field() ?>
24
25         <div class="form-group">
26             <label for="judul">Judul Buku</label>
27             <input type="text" class="form-control"
id="judul" name="judul" value="<?= old('judul') ?>"
required>
28         </div>
29
30         <div class="form-group">
31             <label for="penulis">Penulis</label>
32             <input type="text" class="form-control"
id="penulis" name="penulis" value="<?= old('penulis')
?>" required>
33         </div>
34
35         <div class="form-group">
36             <label for="penerbit">Penerbit</label>
37             <input type="text" class="form-control"
id="penerbit" name="penerbit" value="<?=
old('penerbit') ?>" required>
38         </div>
39
40         <div class="form-group">

```

41	<pre> <label for="tahun_terbit">Tahun Terbit</label> <input type="number" class="form-control" id="tahun_terbit" name="tahun_terbit" value="<?= 42 old('tahun_terbit') ?>" required> 43 </div> 44 <button type="submit" class="btn btn- 45 primary">Simpan</button> 46 </form> 47 </div> 48 <?= \$this->endSection() ?> </pre>
----	---

Tabel 8. Source Code Hasil Jawaban create.php

- edit.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	
5	<div class="container mt-5">
6	<pre> <div class="d-flex justify-content-between align- items-center mb-3"> 7 <a href="<?= base_url('books') ?>" class="btn btn-secondary">Kembali 8 <h2>Edit Buku</h2> 9 </div> 10 11 <?php if (session()->getFlashdata('errors')): ?> 12 <div class="alert alert-danger"> 13 14 <?php foreach (session()- >getFlashdata('errors') as \$error): ?> 15 <?= \$error ?> 16 <?php endforeach ?> 17 18 </div> 19 <?php endif ?> 20 21 </pre>

```

22     <form action="<?=
23     base_url('books/update/' . $book['id']) ?>"
24     method="post">
25         <?= csrf_field() ?>
26
27         <div class="form-group">
28             <label for="judul">Judul Buku</label>
29             <input type="text" class="form-control"
30             id="judul" name="judul" value="<?= old('judul',
31             $book['judul']) ?>" required>
32         </div>
33
34         <div class="form-group">
35             <label for="penulis">Penulis</label>
36             <input type="text" class="form-control"
37             id="penulis" name="penulis" value="<?= old('penulis',
38             $book['penulis']) ?>" required>
39         </div>
40
41         <div class="form-group">
42             <label for="penerbit">Penerbit</label>
43             <input type="text" class="form-control"
44             id="penerbit" name="penerbit" value="<?=
45             old('penerbit', $book['penerbit']) ?>" required>
46         </div>
47
48         <div class="form-group">
49             <label for="tahun_terbit">Tahun
50             Tahun Terbit</label>
51             <input type="number" class="form-control"
52             id="tahun_terbit" name="tahun_terbit" value="<?=
53             old('tahun_terbit', $book['tahun_terbit']) ?>"
54             required>
55         </div>
56
57         <button type="submit" class="btn btn-
58         primary">Simpan</button>
59     </form>
60 </div>
61
62 <?= $this->endSection() ?>

```

Tabel 9. Source Code Hasil Jawaban edit.php

- index.php

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	
5	<div class="container mt-5">
6	<div class="d-flex justify-content-between align-items-center mb-3">
7	<a href="<?= base_url('dashboard') ?>"
	class="btn btn-secondary">Kembali
8	<h2>Daftar Buku</h2>
9	<a href="<?= base_url('books/create') ?>"
	class="btn btn-primary">Tambah Buku Baru
10	</div>
11	
12	<table class="table table-striped table-bordered">
13	<thead class="thead-dark">
14	<tr>
15	<th>ID</th>
16	<th>Judul Buku</th>
17	<th>Penulis</th>
18	<th>Penerbit</th>
19	<th>Tahun Terbit</th>
20	<th>Opsis</th>
21	</tr>
22	</thead>
23	<tbody>
24	<?php foreach (\$books as \$book): ?>
25	<tr>
26	<td><?= \$book['id'] ?></td>
27	<td><?= \$book['judul'] ?></td>
28	<td><?= \$book['penulis'] ?></td>
29	<td><?= \$book['penerbit'] ?></td>
30	<td><?= \$book['tahun_terbit']
	?></td>
31	<td>
32	<a href="<?=
	base_url('books/edit/' . \$book['id']) ?>" class="btn btn-
	warning btn-sm">Edit

33	<code><a href="<?=</code>
	<code>base_url('books/delete/' . \$book['id']) ?>" class="btn</code>
	<code>btn-danger btn-sm" onclick="return confirm('Apakah Anda</code>
	<code>yakin ingin menghapus data ini?');">Delete</code>
34	<code></td></code>
35	<code></tr></code>
36	<code><?php endforeach; ?></code>
37	<code></tbody></code>
38	<code></table></code>
39	<code></div></code>
40	
41	<code><?= \$this->endSection() ?></code>

Tabel 10. Source Code Hasil Jawaban index.php

- **UserModel.php**

1	<code><?php namespace App\Models;</code>
2	
3	<code>use CodeIgniter\Model;</code>
4	
5	<code>class UserModel extends Model</code>
6	<code>{</code>
7	<code> protected \$table = 'users';</code>
8	<code> protected \$primaryKey = 'id';</code>
9	<code> protected \$allowedFields = ['name', 'email',</code>
	<code>'password'];</code>
10	
11	<code> protected \$beforeInsert = ['hashPassword'];</code>
12	<code> protected \$beforeUpdate = ['hashPassword'];</code>
13	
14	<code> protected function hashPassword(array \$data)</code>
15	<code> {</code>
16	<code> if (isset(\$data['data']['password'])) {</code>
17	<code> \$data['data']['password'] =</code>
	<code>password_hash(\$data['data']['password'],</code>
	<code>PASSWORD_DEFAULT);</code>
18	<code> }</code>
19	<code> return \$data;</code>
20	<code> }</code>
21	<code>}</code>

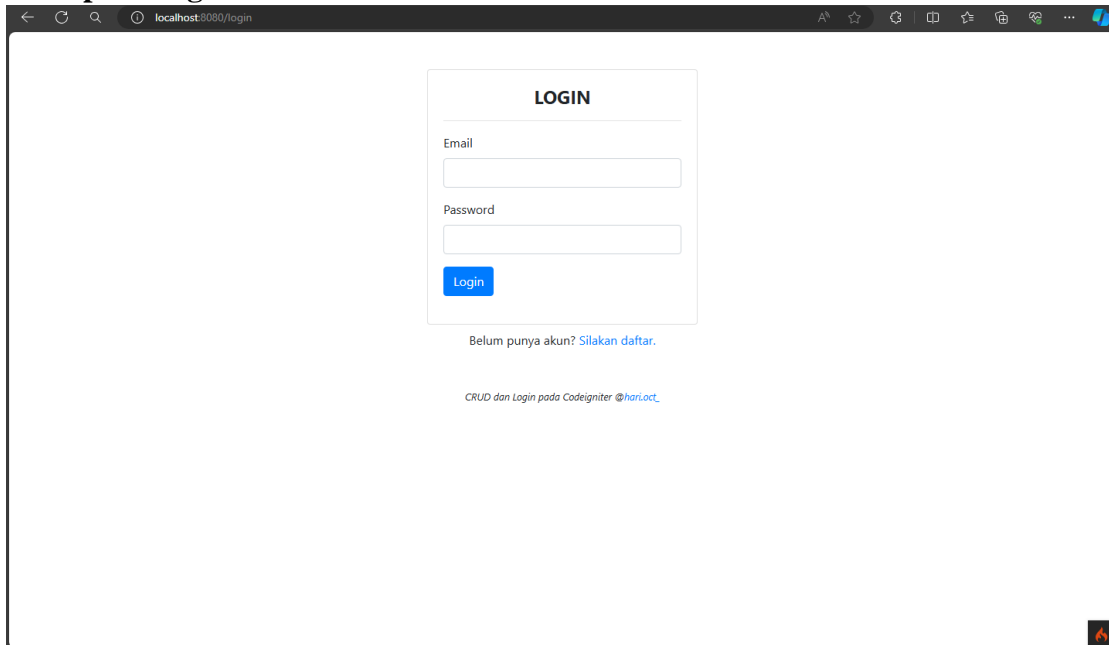
Tabel 11. Source Code Hasil Jawaban UserModel.php

- **BookModel.php**

```
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',
18     'penerbit', 'tahun_terbit'];
19
20     protected $useTimestamps = true;
21     protected $createdField  = 'created_at';
22     protected $updatedField  = 'updated_at';
23     protected $deletedField  = 'deleted_at';
24
25     protected $validationRules      = [];
26     protected $validationMessages = [];
27     protected $skipValidation       = false;
28 }
```

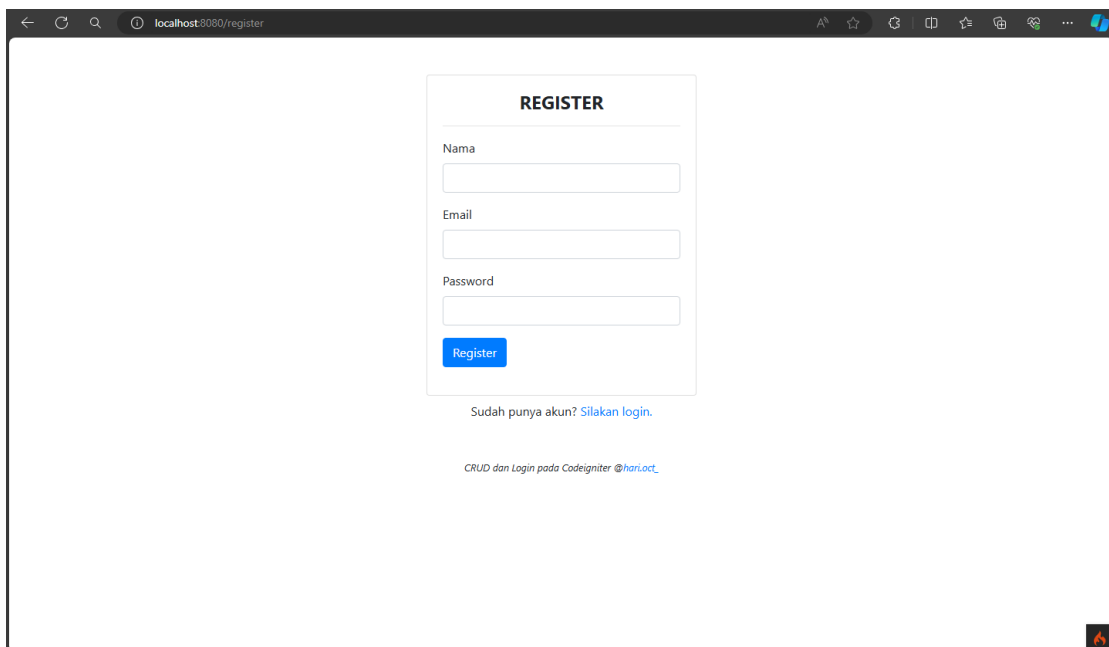
Tabel 12. Source Code Hasil Jawaban BookModel.php

B. Output Program



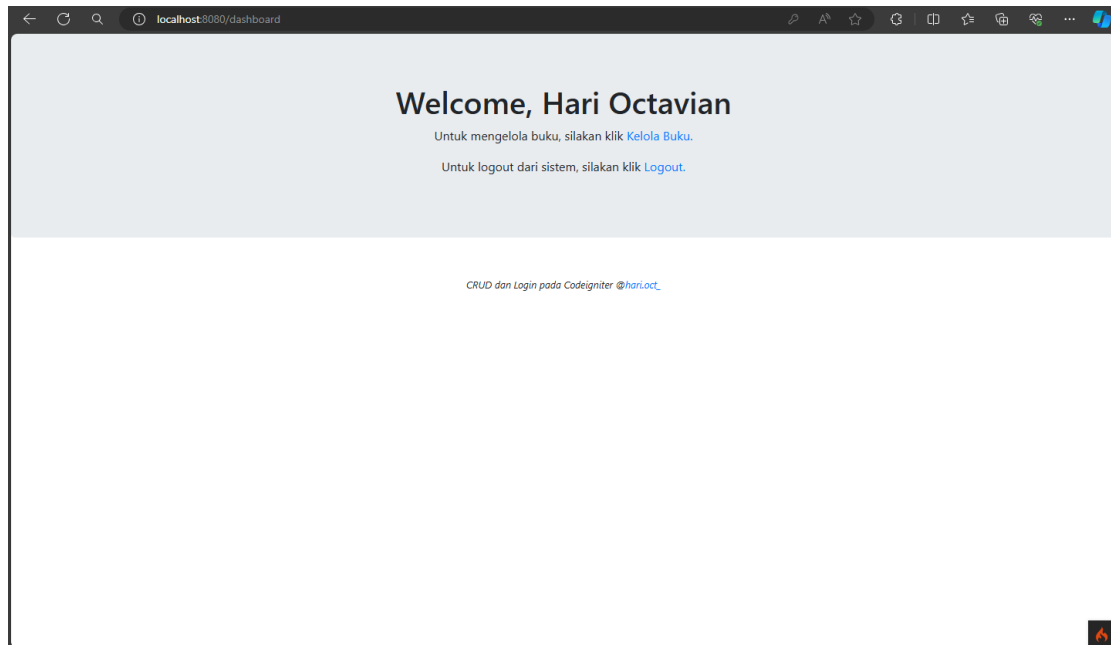
A screenshot of a web browser displaying a login page. The browser's address bar shows 'localhost:8080/login'. The page features a central white box with a light gray border. At the top of this box is the title 'LOGIN' in bold black text. Below the title are two input fields: 'Email' and 'Password'. Under the 'Password' field is a blue button labeled 'Login'. Below the input fields, there is a link that says 'Belum punya akun? Silakan daftar.' and a footer note that reads 'CRUD dan Login pada Codeigniter @hariOct_'. The browser's toolbar at the top includes back, forward, search, and other navigation icons.

Gambar 1. Screenshot halaman Login

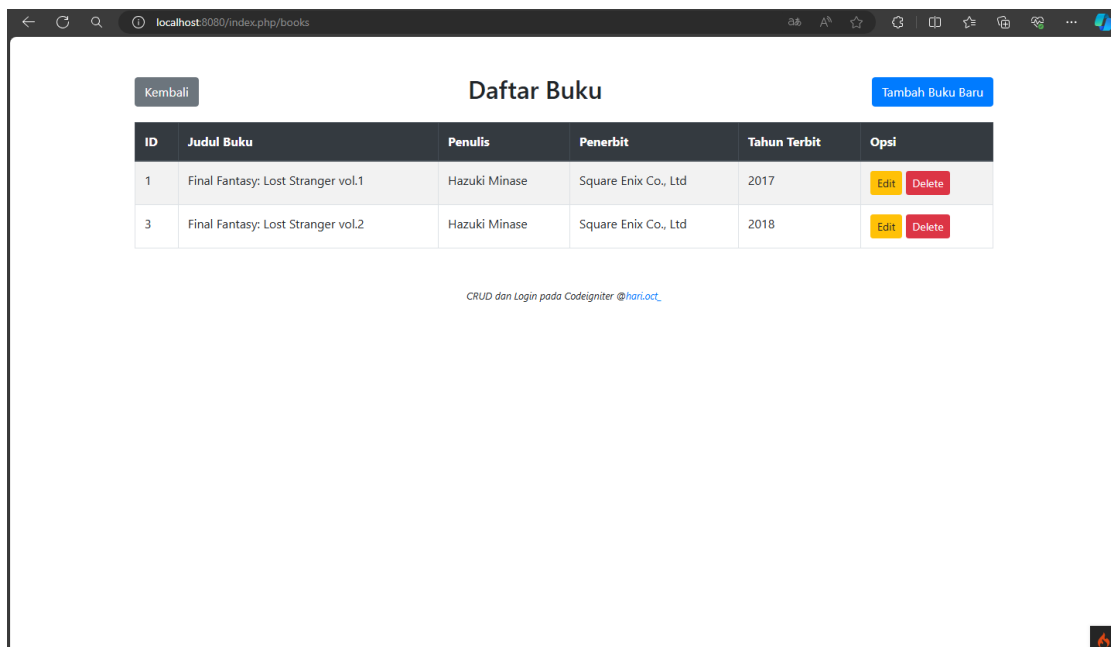


A screenshot of a web browser displaying a register page. The browser's address bar shows 'localhost:8080/register'. The page features a central white box with a light gray border. At the top of this box is the title 'REGISTER' in bold black text. Below the title are three input fields: 'Nama', 'Email', and 'Password'. Under the 'Password' field is a blue button labeled 'Register'. Below the input fields, there is a link that says 'Sudah punya akun? Silakan login.' and a footer note that reads 'CRUD dan Login pada Codeigniter @hariOct_'. The browser's toolbar at the top includes back, forward, search, and other navigation icons.

Gambar 2. Screenshot halaman Register



Gambar 3. Screenshot halaman Dashboard



Gambar 4. Screenshot halaman Daftar Buku

localhost:8080/books/create

Kembali

Tambah Buku Baru

Judul Buku

Penulis

Penerbit

Tahun Terbit

Simpan

CRUD dan Login pada Codeigniter @hariOct_

Gambar 5. Screenshot halaman Tambah Buku

localhost:8080/books/edit/1

Kembali

Edit Buku

Judul Buku

Final Fantasy: Lost Stranger vol.1

Penulis

Hazuki Minase

Penerbit

Square Enix Co., Ltd

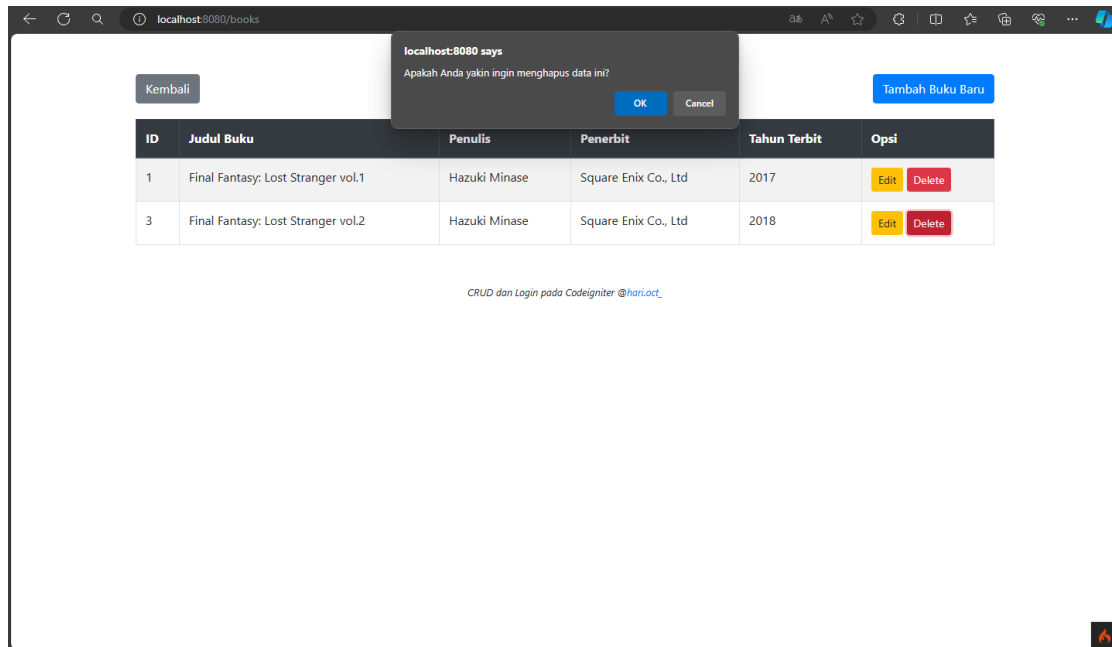
Tahun Terbit

2017

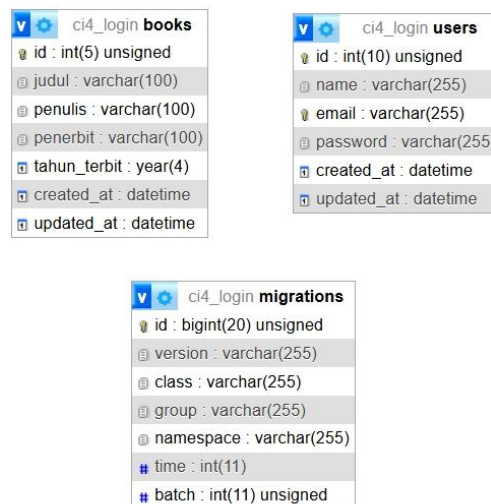
Simpan

CRUD dan Login pada Codeigniter @hariOct_

Gambar 6. Screenshot halaman Edit Buku



Gambar 7. Screenshot halaman Hapus Buku



Gambar 8. Skema Database

C. Pembahasan

- Routes.php

Mendefinisikan rute-rute dalam aplikasi berbasis CodeIgniter. Pada bagian awal, rute-rute untuk halaman login dan register diatur, di mana rute GET dan POST untuk login serta register mengarah ke metode yang relevan di controller masing-masing (LoginController dan RegisterController). Rute untuk logout juga ditentukan. Selanjutnya, ada rute untuk

dashboard yang dilindungi dengan filter auth yang memastikan hanya pengguna yang terautentikasi yang dapat mengaksesnya. Selain itu, terdapat grup rute untuk manajemen buku yang juga dilindungi oleh filter auth. Dalam grup ini, rute-rute seperti melihat daftar buku, membuat, mengedit, memperbarui, dan menghapus buku didefinisikan dan diarahkan ke metode yang sesuai dalam BookController.

- LoginController.php

Berfungsi untuk menangani proses autentikasi pengguna. Pada konstruktor, model UserModel dan beberapa helper (form dan url) diinisialisasi. Metode index memeriksa apakah pengguna sudah login menggunakan metode isLoggedIn(), dan jika ya, pengguna diarahkan ke halaman dashboard; jika tidak, halaman login ditampilkan. Metode login memproses data login yang diterima melalui metode POST, memvalidasi email dan password, dan jika validasi berhasil serta kredensial benar, pengguna disimpan dalam sesi dan diarahkan ke halaman dashboard. Jika gagal, pesan kesalahan ditampilkan dan pengguna diarahkan kembali ke halaman login. Metode logout menghancurkan sesi pengguna dan mengarahkan mereka kembali ke halaman login.

- RegisterController.php

RegisterController bertanggung jawab untuk menangani proses pendaftaran pengguna baru. Dalam konstruktor, model UserModel diinisialisasi dan helper form serta url dimuat. Metode index menampilkan halaman registrasi dengan judul yang sesuai. Metode store memproses data yang diterima dari form registrasi melalui metode POST, memvalidasi input untuk memastikan nama, email, dan password diisi dengan benar sesuai aturan (misalnya, email harus unik dan password minimal 6 karakter). Jika validasi gagal, pengguna diarahkan kembali ke halaman registrasi. Jika berhasil, data pengguna disimpan ke database menggunakan metode save dari UserModel. Jika penyimpanan berhasil, pesan sukses disimpan dalam sesi dan pengguna diarahkan ke halaman login; jika gagal, pesan kesalahan ditampilkan dan pengguna diarahkan kembali ke halaman registrasi dengan input sebelumnya.

- LogoutController.php

LogoutController menangani proses logout pengguna. Metode index bertugas untuk menghapus data sesi pengguna, seperti nama, email, dan status login (logged_in),

menggunakan metode `session()->remove()`. Setelah data sesi dihapus, pengguna diarahkan kembali ke halaman login menggunakan metode `redirect()->to(base_url('login'))`. Ini memastikan bahwa setelah logout, pengguna harus login kembali untuk mengakses area yang dilindungi dalam aplikasi.

- **BookController.php**

Berfungsi untuk menangani operasi CRUD (Create, Read, Update, Delete) terkait buku. Controller ini menggunakan model `BookModel` untuk berinteraksi dengan database buku. Pada metode `index`, semua buku diambil dari database dan ditampilkan dalam tampilan `books/index`. Metode `create` menampilkan formulir untuk menambah buku baru. Metode `store` menerima data dari formulir melalui metode `POST`, memvalidasi input (judul, penulis, penerbit, dan tahun terbit) sesuai aturan yang ditentukan, dan jika validasi berhasil, data buku disimpan ke database. Jika penyimpanan berhasil, pengguna diberi pesan sukses; jika gagal, pesan kesalahan ditampilkan dan pengguna tetap di halaman formulir.

Metode `edit` mengambil data buku berdasarkan ID dan menampilkan formulir pengeditan dengan data buku tersebut. Metode `update` memproses data yang diperbarui dari formulir, memvalidasi input, dan jika validasi berhasil, memperbarui data buku di database. Jika pembaruan berhasil, pengguna diberi pesan sukses; jika gagal, pesan kesalahan ditampilkan dan pengguna tetap di halaman formulir pengeditan. Metode `delete` menghapus buku berdasarkan ID dan mengarahkan pengguna kembali ke halaman daftar buku dengan pesan sukses jika penghapusan berhasil. Secara keseluruhan, controller ini mengatur alur kerja penuh dari penambahan, pembaruan, penghapusan, dan penampilan daftar buku dengan validasi dan penanganan pesan kesalahan yang tepat.

- **Auth.php**

Merupakan filter yang memastikan hanya pengguna yang sudah login yang dapat mengakses rute tertentu. Metode `before` memeriksa apakah rute saat ini termasuk dalam daftar pengecualian (`login` dan `register`). Jika ya, filter tidak melakukan apa-apa. Jika tidak, filter memeriksa status login pengguna melalui sesi. Jika pengguna belum login, mereka akan diarahkan ke halaman login dengan pesan kesalahan yang mengharuskan mereka untuk login terlebih dahulu. Metode `after` saat ini tidak melakukan apa-apa, namun bisa digunakan untuk

logika yang perlu dieksekusi setelah respons dikirim ke pengguna. Filter ini membantu melindungi rute-rute sensitif dengan memastikan hanya pengguna yang terautentikasi yang dapat mengaksesnya.

- login.php

Merupakan template tampilan untuk halaman login dalam aplikasi berbasis CodeIgniter. Template ini memperluas layout utama (layouts/main) dan mendefinisikan konten utama di dalam section content. Pada bagian utama konten, terdapat form login yang berada dalam sebuah card Bootstrap yang terpusat di halaman. Jika terdapat pesan flash dari sesi, seperti pesan sukses atau error, pesan tersebut akan ditampilkan di atas form login. Terdapat juga validasi error list untuk menampilkan kesalahan validasi. Form login terdiri dari dua input field, satu untuk email dan satu untuk password, keduanya diwajibkan diisi. Setelah form diisi, pengguna dapat menekan tombol "Login" untuk mengirim data. Di bagian bawah card, terdapat link untuk pengguna yang belum memiliki akun untuk menuju halaman registrasi. Tampilan ini menggabungkan penggunaan form helper dan styling Bootstrap untuk membuat halaman login yang user-friendly dan responsif.

- create.php

Merupakan template untuk menambahkan buku baru dalam aplikasi. Halaman ini memperluas layout utama (layouts/main) dan menentukan konten utama di dalam section content. Pada bagian utama konten, terdapat judul halaman dan tombol "Kembali" yang mengarahkan pengguna kembali ke halaman daftar buku. Di bawahnya, terdapat form untuk menambahkan buku baru. Jika terdapat kesalahan validasi, pesan kesalahan akan ditampilkan dalam kotak alert dengan warna merah. Form ini terdiri dari empat input field untuk judul buku, penulis, penerbit, dan tahun terbit. Setiap input field dilengkapi dengan validasi required, dan nilai yang sudah dimasukkan oleh pengguna sebelumnya (jika ada) akan ditampilkan kembali menggunakan fungsi old(). Setelah pengguna mengisi formulir, mereka dapat menekan tombol "Simpan" untuk menyimpan data buku baru. Tampilan ini menggunakan Bootstrap untuk styling dan memberikan tampilan yang menarik dan responsif kepada pengguna.

- edit.php

Merupakan template untuk mengedit buku dalam aplikasi. Template ini memperluas layout utama (layouts/main) dan menentukan konten utama di dalam section content. Pada bagian utama konten, terdapat judul halaman dan tombol "Kembali" yang mengarahkan pengguna kembali ke halaman daftar buku. Di bawahnya, terdapat form untuk mengedit buku yang sudah ada. Jika terdapat kesalahan validasi, pesan kesalahan akan ditampilkan dalam kotak alert dengan warna merah. Form ini sudah diisi dengan data buku yang sudah ada dan dapat diedit oleh pengguna. Setelah pengguna melakukan perubahan, mereka dapat menekan tombol "Simpan" untuk menyimpan perubahan tersebut. Tampilan ini menggunakan Bootstrap untuk styling dan memberikan tampilan yang menarik dan responsif kepada pengguna.

- index.php

Merupakan template untuk menampilkan daftar buku dalam aplikasi. Template ini memperluas layout utama (layouts/main) dan menentukan konten utama di dalam section content. Pada bagian utama konten, terdapat judul halaman, tombol "Kembali" yang mengarahkan pengguna kembali ke halaman dashboard, dan tombol "Tambah Buku Baru" yang mengarahkan pengguna ke halaman formulir tambah buku baru. Di bawahnya, terdapat tabel yang menampilkan daftar buku yang sudah ada, dengan kolom untuk ID, judul buku, penulis, penerbit, tahun terbit, dan opsi untuk mengedit atau menghapus buku. Setiap buku ditampilkan dalam sebuah baris tabel, dengan tombol "Edit" dan "Delete" untuk setiap buku yang memungkinkan pengguna untuk melakukan aksi tersebut. Tombol "Delete" memunculkan konfirmasi sebelum pengguna dapat menghapus buku. Tampilan ini menggunakan Bootstrap untuk styling dan memberikan tampilan yang menarik dan responsif kepada pengguna.

- UserModel.php

UserModel merupakan model dalam aplikasi yang bertanggung jawab untuk mengelola data pengguna. Terhubung dengan tabel users dalam database, model ini memiliki properti yang menentukan field-field yang dapat diisi oleh pengguna. Selain itu, model ini dilengkapi dengan metode proteksi sebelum penyisipan atau pembaruan data, di mana metode hashPassword digunakan untuk mengenkripsi kata sandi pengguna sebelum disimpan ke

dalam database. Dengan menggunakan enkripsi kata sandi menggunakan fungsi `password_hash`, model ini meningkatkan keamanan data pengguna dalam aplikasi.

- `BookModel.php`

`BookModel` mengelola data buku dalam aplikasi. Model ini terhubung ke tabel `books` dalam database, dengan kolom `id`, `judul`, `penulis`, `penerbit`, dan `tahun_terbit`. Properti `allowedFields` menentukan field-field yang dapat diisi oleh penggunaan metode `save` atau `insert`. Model ini menggunakan `auto-increment` untuk nilai `id` dan menggunakan `timestamps` untuk mengatur field `created_at` dan `updated_at`.

Meskipun properti `validationRules`, `validationMessages`, dan `skipValidation` masih kosong, mereka dapat digunakan untuk menentukan aturan validasi untuk data buku, pesan kesalahan yang sesuai, dan apakah validasi harus dilewati atau tidak. Dengan menggunakan validasi, aplikasi dapat memastikan bahwa data yang dimasukkan ke dalam database memenuhi kriteria yang diinginkan, sehingga meningkatkan keandalan dan keamanan aplikasi.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

[Praktikum-Pemrograman-Web-II/PRAK601 at main · harioct/Praktikum-Pemrograman-Web-II \(github.com\)](https://github.com/harioct/Praktikum-Pemrograman-Web-II/PRAK601)