

**LAPORAN PRAKTIKUM  
PEMROGRAMAN WEB II  
MODUL 6**



**CRUD dan Login pada Codeigniter**

**Oleh:**

**Nor Aina Eva Yanti      NIM. 2210817120004**

**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 : Nor Aina Eva Yanti  
NIM : 2210817120004

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.....	6
A.    Source Code .....	6
B.    Output Program.....	44
C.    Pembahasan .....	46
D.    Tautan Git .....	74

## **DAFTAR GAMBAR**

Gambar 1. MODUL 6 Screenshot Hasil Jawaban Output 1 .....	44
Gambar 2. MODUL 6 Screenshot Hasil Jawaban Output 2.....	45
Gambar 3. MODUL 6 Screenshot Hasil Jawaban Output 3.....	45
Gambar 4. MODUL 6 Screenshot Hasil Jawaban Output 4.....	46
Gambar 5. MODUL 6 Screenshot Hasil Jawaban Output 5.....	46

## DAFTAR TABEL

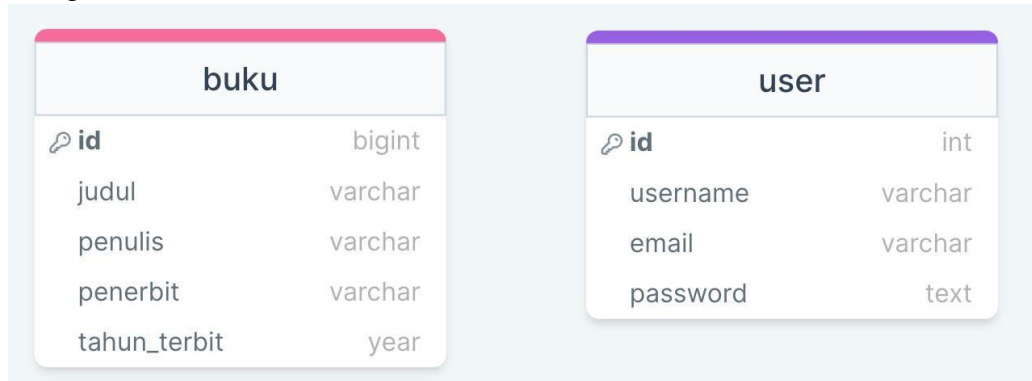
Tabel 1. MODUL 6 Source Code Routes.php.....	7
Tabel 2. MODUL 6 Source Code BaseController.php.....	8
Tabel 3. MODUL 6 Source Code Buku.php .....	12
Tabel 4. MODUL 6 Source Code Home.php .....	12
Tabel 5. MODUL 6 Source Code Login.php .....	13
Tabel 6. MODUL 6 Source Code Register.php.....	15
Tabel 7. MODUL 6 Source Code AuthFilter.php .....	15
Tabel 8. MODUL 6 Source Code vw_add.php .....	19
Tabel 9. MODUL 6 Source Code vw_buku.php .....	24
Tabel 10. MODUL 6 Source Code vw_edit.php .....	27
Tabel 11. MODUL 6 Source Code vw_login.php.....	32
Tabel 12. MODUL 6 Source Code vw_register.php .....	36
Tabel 13. MODUL 6 Source Code BukuModel.php.....	37
Tabel 14. MODUL 6 Source Code UserModel.php .....	37
Tabel 15. MODUL 6 Source Code 2024-06-03-103310_Users.php.....	38
Tabel 16. MODUL 6 Source Code 2024-06-03-122804_Buku.php .....	39
Tabel 17. MODUL 6 Source Code Seeds Buku.php .....	40
Tabel 18. MODUL 6 Source Code UserSeeder.php .....	40
Tabel 19. MODUL 6 Source Code perpustakaan.sql .....	44

## SOAL

1. Buatlah sebuah website yang memiliki fitur login dan logout. Setelah login pengguna website dapat melakukan CRUD yang terhubung langsung dengan database.

Beberapa syarat yang harus ada pada website:

- Jika seorang pengguna belum login dan mengakses halaman yang mengharuskan login, maka tampilan 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 PRAK601

### A. Source Code

```
1 <?php
2
3 use CodeIgniter\Router\RouteCollection;
4
5 /**
6  * @var RouteCollection $routes
7  */
8 $routes->setAutoRoute(true);
9 $routes->get('/', 'Buku::login');
10 $routes->get('home', 'Buku::index');
11 $routes->get('/register', 'Register::index');
12 $routes->post('/register/process', 'Register::process');
```

```

13 $routes->get('/login', 'Login::index');
14 $routes->post('/login/process', 'Login::process');
15 $routes->get('/logout', 'Login::logout');
16 $routes->get('/buku/create', 'Buku::create');
17 $routes->post('buku/store', 'Buku::store');
18 $routes->post('buku/update/', 'Buku::update', ['filter'
=> 'auth']);
19 $routes->get('buku/edit/', 'Buku::edit', ['filter' =>
'auth']);
20 $routes->get('buku/delete/', 'Buku::delete', ['filter' =>
'auth']);

```

*Tabel 1. MODUL 6 Source Code Routes.php*

```

1 <?php
2
3 namespace App\Controllers;
4
5 use CodeIgniter\Controller;
6 use CodeIgniter\HTTP\CLIRequest;
7 use CodeIgniter\HTTP\IncomingRequest;
8 use CodeIgniter\HTTP\RequestInterface;
9 use CodeIgniter\HTTP\ResponseInterface;
10 use Psr\Log\LoggerInterface;
11
12 /**
13  * Class BaseController
14  *
15  * BaseController provides a convenient place for loading
16  * components
17  * and performing functions that are needed by all your
18  * controllers.
19  * Extend this class in any new controllers:
20  *     class Home extends BaseController
21  *
22  * For security be sure to declare any new methods as
23  * protected or private.
24  */
25 abstract class BaseController extends Controller
26 {
27     /**
28      * Instance of the main Request object.
29      *
30      * @var CLIRequest|IncomingRequest
31      */
32     protected $request;
33
34     /**
35      * An array of helpers to be loaded automatically

```

33	upon
34	* class instantiation. These helpers will be available
35	* to all other controllers that extend BaseController.
36	*/
37	* @var list<string>
38	protected \$helpers = [];
39	/**
40	* Be sure to declare properties for any property fetch you initialized.
41	* The creation of dynamic property is deprecated in PHP 8.2.
42	*/
43	// protected \$session;
44	/**
45	* @return void
46	*/
47	public function initController(RequestInterface \$request, ResponseInterface \$response, LoggerInterface \$logger)
48	{
49	// Do Not Edit This Line
50	parent::initController(\$request, \$response, \$logger);
51	// Preload any models, libraries, etc, here.
52	// E.g.: \$this->session = \Config\Services::session();
53	}
54	}
55	
56	
57	
58	

*Tabel 2. MODUL 6 Source Code BaseController.php*

1	<?php namespace App\Controllers;
2	
3	use App\Models\BukuModel;
4	use CodeIgniter\Controller;
5	use App\Models\PostModel;
6	
7	class Buku extends Controller
8	{
9	/**
10	*
11	* @var Model



```
12     */
13     protected $model;
14
15     public function __construct()
16     {
17         $this->model = new BukuModel();
18         $this->helpers = ['form', 'url'];
19
20     }
21
22     public function login()
23     {
24         $data = [
25             'posts' => $this->model->paginate(10),
26             'pager' => $this->model->pager,
27             'title' => 'POST LIST'
28         ];
29
30         return view('vw_login', $data);
31     }
32
33     public function index()
34     {
35         if (!session()->get('logged_in')) {
36             return redirect()->to(base_url('/login'));
37         }
38         $data = [
39             'posts' => $this->model->paginate(10),
40             'pager' => $this->model->pager,
41             'title' => 'POST LIST'
42         ];
43
44         return view('vw_buku', $data);
45     }
46
47     public function create()
48     {
49         // Check if user is authenticated
50         if (!session()->get('logged_in')) {
51             return redirect()->to(base_url('/login'));
52         }
53         $data = ['title' => 'Create new post'];
54
55         return view('vw_add', $data);
56     }
57
58     public function store()
```

```

59     {
60         $data      =      $this->request->getPost(['judul',
        'penulis', 'penerbit', 'tahun_terbit']);
61
62         if (! $this->validateData($data, $this->model-
        >validationRules)) {
63             return $this->create();
64         }
65
66         $post = $this->validator->getValidated();
67
68         $save = $this->model->save([
69             'judul' => $post['judul'],
70             'penulis' => $post['penulis'],
71             'penerbit' => $post['penerbit'],
72             'tahun_terbit' => $post['tahun_terbit'],
73         ]);
74
75         session()->setFlashdata('success',      'Post      has
        been added successfully.');
```

```

76         return redirect()->to(base_url('home'));
77     }
78
79     public function edit($id)
80     {
81         // Check if user is authenticated
82         if (!session()->get('logged_in')) {
83             return redirect()->to(base_url('/login'));
84         }
85         $post = $this->model->find($id);
86         if (empty($post)) {
87             session()->setFlashdata('error', 'Post      not
        found');
```

```

88             return redirect()->back();
89         }
90         $data = [
91             'title' => 'Edit Post',
92             'post' => $post
93         ];
94
95         return view('vw_edit', $data);
96     }
97
98     public function update($id)
99     {
100         // Check if user is authenticated
101         if (!session()->get('logged in')) {
```

```

102         return redirect()->to(base_url('/login'));
103     }
104     $post = $this->model->find($id);
105
106     if (empty($post)) {
107         session()->setFlashdata('error', 'Post not
found');
108         return redirect()->back();
109     }
110
111     $data = $this->request->getPost(['judul',
'penulis', 'penerbit', 'tahun_terbit']);
112
113     if (! $this->validateData($data, $this->model-
>validationRules)) {
114         return $this->create();
115     }
116
117     $updatedPost = $this->validator->getValidated();
118
119     $update = $this->model->update($post['id'],
$updatePost);
120
121     if ($update) {
122         session()->setFlashdata('success', 'Post has
been updated successfully');
123         return redirect()->to(base_url('home'));
124     } else {
125         session()->setFlashdata('error', 'Some
problems occured, please try again. ');
126         return redirect()->back();
127     }
128 }
129
130 public function delete($id)
131 {
132     // Check if user is authenticated
133     if (!session()->get('logged_in')) {
134         return redirect()->to(base_url('home'));
135     }
136
137     if (empty($id)) {
138         return redirect()->to(base_url('home'));
139     }
140
141     $delete = $this->model->delete($id);
142

```

143	if (\$delete) {
144	session()->setFlashdata('success', 'Post has been removed successfully.');
145	return redirect()->to(base_url('home'));
146	} else {
147	session()->setFlashdata('error', 'Some problems occurred, please try again.');
148	return redirect()->to(base_url('home'));
149	}
150	}
151	}

*Tabel 3. MODUL 6 Source Code Buku.php*

1	<?php
2	
3	namespace App\Controllers;
4	
5	class Home extends BaseController
6	{
7	public function index(): string
8	{
9	return view('vw_buku');
10	}
11	}

*Tabel 4. MODUL 6 Source Code Home.php*

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Models\UsersModel;
6	
7	class Login extends BaseController
8	{
9	public function index()
10	{
11	return view('vw_login');
12	}
13	
14	public function process()
15	{
16	\$users = new UsersModel();
17	\$username = \$this->request->getVar('username');
18	\$password = \$this->request->getVar('password');
19	\$dataUser = \$users->where('username', \$username)->first();
20	
21	if (\$dataUser) {

```

22          // Check the password using object property
syntax
23          if (password_verify($password, $dataUser-
>password)) {
24              session()->set([
25                  'username' => $dataUser->username,
26                  'logged_in' => true
27              ]);
28              return redirect()->to(base_url('home'));
29          } else {
30              session()->setFlashdata('error',
'Username & Password Salah');
31              return redirect()->to(base_url('login'));
32          }
33      } else {
34          session()->setFlashdata('error', 'Username &
Password Salah');
35          return redirect()->to(base_url('login'));
36      }
37  }
38
39  public function logout()
40  {
41      session()->destroy();
42      return redirect()->to(base_url('login'));
43  }
44  }

```

*Tabel 5. MODUL 6 Source Code Login.php*

```

1  <?php
2
3  namespace App\Controllers;
4
5  use App\Models\UsersModel;
6
7  class Register extends BaseController
8  {
9      public function index()
10     {
11         return view('vw_register');
12     }
13
14     public function process()
15     {
16         // Validate input
17         if (!$this->validate([
18             'username' => [
19                 'rules'
=>

```

```

20 'required|min_length[4]|max_length[20]|is_unique[users.us
21      'errors' => [
22      'required' => '{field} Harus diisi',
23      'min_length' => '{field} Minimal 4
Karakter',
24      'max_length' => '{field} Maksimal 20
Karakter',
25      'is_unique' => 'Username sudah
digunakan sebelumnya'
26      ]
27      ],
28      'password' => [
29      'rules' => [
30      'required|min_length[0]|max_length[50]',
31      'errors' => [
32      'required' => '{field} Harus diisi',
33      'min_length' => '{field} Minimal 0
Karakter',
34      'max_length' => '{field} Maksimal 50
Karakter',
35      ]
36      ],
37      'password_conf' => [
38      'rules' => 'matches[password]',
39      'errors' => [
40      'matches' => 'Konfirmasi Password
tidak sesuai dengan password',
41      ]
42      ],
43      'email' => [
44      'rules' => [
45      'required|min_length[4]|max_length[100]|valid_email',
46      'errors' => [
47      'required' => '{field} Harus diisi',
48      'min_length' => '{field} Minimal 4
Karakter',
49      'max_length' => '{field} Maksimal 100
Karakter',
50      'valid_email' => 'Email tidak valid'
51      ]
52      ],
53      ])) {
54      session()->setFlashdata('error', $this-
>validator->listErrors());
55      return redirect()->back()->withInput();
56      }

```

```

54
55         $users = new UsersModel();
56
57         // Hash the password before storing it
58         $hashedPassword = password_hash($this->request-
>getVar('password'), PASSWORD_DEFAULT);
59
60         $users->insert([
61             'username' => $this->request-
>getVar('username'),
62             'password' => $hashedPassword,
63             'email' => $this->request->getVar('email')
64         ]);
65
66         return redirect()->to('/login');
67     }
68 }

```

*Tabel 6. MODUL 6 Source Code Register.php*

```

1  <?php
2  // app/Filters/AuthFilter.php
3
4  namespace App\Filters;
5
6  use CodeIgniter\HTTP\RequestInterface;
7  use CodeIgniter\HTTP\ResponseInterface;
8  use CodeIgniter\Filters\FilterInterface;
9
10 class AuthFilter implements FilterInterface
11 {
12     public function before(RequestInterface $request,
13     $arguments = null)
14     {
15         // Check if the user is logged in
16         if (!session()->has('user_id')) {
17             // If not logged in, redirect to the login
18             page
19             return redirect()->to('/login');
20         }
21     }
22
23     public function after(RequestInterface $request,
24     ResponseInterface $response, $arguments = null)
25     {
26         // Do nothing after the request
27     }
28 }

```

*Tabel 7. MODUL 6 Source Code AuthFilter.php*

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	<style>
5	body {
6	background-image:
	url('https://images.unsplash.com/photo-1661097410573-16d926eclec7?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90bylwYWdlfHx8fGVufDB8fHx8fA%3D%3D');
7	background-repeat: no-repeat;
8	background-attachment: fixed;
9	background-position: center;
10	background-size: cover;
11	color: #ececec;
12	}
13	
14	.card {
15	background: rgba(31, 31, 31); /* Opasitas
	background 20% */
16	color: #ececec;
17	box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
18	border-radius: 10px;
19	}
20	
21	.card-header {
22	background-color: #00bfa5;
23	border-bottom: 1px solid #37474f;
24	}
25	
26	.btn-success {
27	background-color: #00bfa5;
28	border: none;
29	}
30	
31	.btn-success:hover {
32	background-color: #008e76;
33	}
34	
35	.btn-secondary {
36	background-color: #000000;
37	border: none;
38	}
39	
40	.btn-secondary:hover {
41	background-color: #333333;
42	}



```

43
44     .alert {
45         background-color: rgba(255, 138, 101, 0.1);
46         color: #ffab40;
47         border: 1px solid #ffab40;
48     }
49
50     .form-control {
51         background-color: #2e2e2e;
52         color: #ececec;
53         border: 1px solid #37474f;
54     }
55
56     .form-control::placeholder {
57         color: #bdbdbd;
58     }
59
60     label {
61         color: #ececec;
62     }
63 </style>
64
65 <div class="container mt-5">
66     <div class="row justify-content-center">
67         <div class="col-md-8">
68             <div class="card shadow-sm">
69                 <div class="card-header text-white">
70                     <h4>Tambah</h4>
71                 </div>
72                 <div class="card-body">
73                     <?php if (session() -
74 >getFlashdata('error')) : ?>
75                         <div class="alert alert-danger">
76                             <? = session() -
77 >getFlashdata('error'); ?>
78                         </div>
79                         <?php endif; ?>
80
81                         <? = validation_list_errors() ?>
82
83                         <? = form_open('buku/store'); ?>
84                         <div class="form-group">
85                             <label for="judul">Judul</label>
86                             <input type="text" class="form-
87 control" id="judul" name="judul" placeholder="Masukkan
88 Judul">
89                         </div>

```

```

86         <div class="form-group">
87             <label
for="penulis">Penulis</label>
88             <input type="text" class="form-
control" id="penulis" name="penulis"
placeholder="Masukkan Penulis">
89         </div>
90         <div class="form-group">
91             <label
for="penerbit">Penerbit</label>
92             <input type="text" class="form-
control" id="penerbit" name="penerbit"
placeholder="Masukkan Penerbit">
93         </div>
94         <div class="form-group">
95             <label for="tahun_terbit">Tahun
Terbit</label>
96             <input type="number"
class="form-control" id="tahun_terbit"
name="tahun_terbit" placeholder="Masukkan Tahun Terbit">
97         </div>
98         <div class="form-group text-right">
99             <button class="btn btn-
success">Simpan</button>
100             <a href="<?= base_url('home')
?>" class="btn btn-secondary">Kembali</a>
101         </div>
102         <?= form_close(); ?>
103     </div>
104 </div>
105 </div>
106 </div>
107 </div>
108
109 <?= $this->endSection() ?>
110
111 <?= $this->section('extra-js') ?>
112 <!-- include summernote css/js -->
113 <link
href="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dis
t/summernote-bs4.min.css" rel="stylesheet">
114 <script
src="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dist
/summernote-bs4.min.js"></script>
115 <script>
116     $(document).ready(function() {
117         $('#post_content').summernote({

```

118	tabsize: 2,
119	height: 500
120	});
121	});
122	</script>
123	<?= \$this->endSection() ?>

*Tabel 8. MODUL 6 Source Code vw\_add.php*

1	<?= \$this->extend('layouts/main') ?>
2	
3	<?= \$this->section('content') ?>
4	<style>
5	body,
6	html {
7	margin: 0;
8	padding: 0;
9	font-family: Arial, sans-serif;
10	height: 100%;
11	display: flex;
12	flex-direction: column;
13	background-color: #121212;
14	color: #ececce;
15	position: relative;
16	overflow: hidden;
17	}
18	
19	body::before {
20	content: "";
21	position: absolute;
22	top: 0;
23	left: 0;
24	width: 100%;
25	height: 100%;
26	background-image:
	url('https://images.unsplash.com/photo-1661097410573-16d926ec1ec7?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90bylwYWdlfHx8fGVufDB8fHx8fA%3D%3D');
27	background-size: cover;
28	background-position: center;
29	opacity: 0.1;
30	z-index: 1;
31	}
32	
33	.navbar {
34	z-index: 2;
35	position: relative;
36	}

```
37
38     .container,
39     .card {
40         z-index: 2;
41         position: relative;
42     }
43
44     .navbar {
45         background-color: #1f1f1f !important;
46         border-bottom: 1px solid #37474f;
47     }
48
49     .navbar-brand,
50     .nav-link {
51         color: #00bfa5 !important;
52     }
53
54     .card {
55         background: #1f1f1f;
56         color: #ececec;
57         box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
58         border-radius: 10px;
59     }
60
61     .card-header {
62         background-color: #00bfa5;
63         border-bottom: 1px solid #37474f;
64     }
65
66     .btn {
67         background-color: #00bfa5;
68         color: white;
69         border: none;
70     }
71
72     .btn:hover {
73         background-color: #008e76;
74     }
75
76     .table thead {
77         background-color: #37474f;
78     }
79
80     .table th,
81     .table td {
82         border: 1px solid #37474f;
83         color: #fff; /* Change table headers and content
```

```

84     text color to white */
85     }
86     .pagination li a {
87         background-color: #1f1f1f !important;
88         color: #00bfa5 !important;
89         border: 1px solid #00bfa5 !important;
90     }
91
92     .pagination li a:hover {
93         background-color: #00bfa5 !important;
94         color: #1f1f1f !important;
95     }
96
97     .alert {
98         background-color: rgba(255, 138, 101, 0.1);
99         color: #ffab40;
100        border: 1px solid #ffab40;
101    }
102 </style>
103
104 <!-- Navbar -->
105 <nav class="navbar navbar-expand-lg navbar-light shadow-
sm">
106     <div class="container">
107         <a class="navbar-brand" href="<?= base_url('/')
?>">My Library</a>
108         <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
109             <span class="navbar-toggler-icon"></span>
110         </button>
111         <div class="collapse navbar-collapse"
id="navbarNav">
112             <ul class="navbar-nav ml-auto">
113                 <?php if (session()->get('logged_in')) :
?>
114                     <li class="nav-item">
115                         <a class="nav-link"
href="#">Welcome, <?= session()->get('username'); ?></a>
116                     </li>
117                     <li class="nav-item">
118                         <a class="nav-link" href="<?=
base_url('login/logout'); ?>">Logout</a>
119                     </li>
120                 <?php else : ?>

```

```

121         <li class="nav-item">
122             <a class="nav-link" href="<?=
base_url('login'); ?>">Login</a>
123         </li>
124         <li class="nav-item">
125             <a class="nav-link" href="<?=
base_url('register'); ?>">Register</a>
126         </li>
127     <?php endif; ?>
128 </ul>
129 </div>
130 </div>
131 </nav>
132
133 <div class="container mt-5">
134     <div class="row justify-content-center">
135         <div class="col-md-10">
136             <div class="card shadow-sm">
137                 <div class="card-header text-white">
138                     <h5 class="mb-0">List Buku</h5>
139                     <a href="<?=
base_url('buku/create'); ?>" class="btn btn-sm float-
right">Tambah Data</a>
140                 </div>
141                 <div class="card-body">
142                     <?php if (session() -
>getFlashdata('success')) : ?>
143                         <div class="alert alert-
success">
144                             <?= session() -
>getFlashdata('success'); ?>
145                         </div>
146                     <?php endif; ?>
147
148                     <?php if (session() -
>getFlashdata('error')) : ?>
149                         <div class="alert alert-danger">
150                             <?= session() -
>getFlashdata('error'); ?>
151                         </div>
152                     <?php endif; ?>
153
154                     <table class="table table-bordered
table-striped table-hover">
155                         <thead class="text-center">
156                             <tr>
157                                 <th>

```

```

158 scope="col">Judul</th>                                <th
159 scope="col">Penulis</th>                                <th
160 scope="col">Penerbit</th>                                <th      scope="col">Tahun
161 Terbit</th>                                              <th
162 scope="col">Action</th>
163                                     </tr>
164                                     </thead>
165                                     <tbody>
166                                     <?php if (!empty($posts) &&
is_array($posts)) : ?>
167                                     <?php foreach ($posts as
$row) : ?>
168                                     <tr>
169                                     <td><?=
$row['judul']; ?></td>
170                                     <td><?=
$row['penulis']; ?></td>
171                                     <td><?=
$row['penerbit']; ?></td>
172                                     <td><?=
$row['tahun_terbit']; ?></td>
173                                     <td class="text-
center">
174                                     <form
onsubmit="return confirm('Apakah Anda Yakin ?');"
action="<?= base_url('buku/delete/' . $row['id']); ?>"
method="POST">
175                                     <input
type="hidden" name="{csrf_token}" value="{csrf_hash}">
176                                     <input
type="hidden" name="_method" value="DELETE">
177                                     <a
href="<?= base_url('buku/edit/' . $row['id']); ?>"
class="btn btn-warning btn-sm">Edit</a>
178                                     <button
type="submit"      class="btn      btn-danger      btn-
sm">Delete</button>
179                                     </form>
180                                     </td>
181                                     </tr>
182                                     <?php endforeach; ?>
183                                     <?php else : ?>
                                     <tr>

```

184	<code>&lt;td colspan="5"</code>
185	<code>class="text-center"&gt;No post found.&lt;/td&gt;</code>
186	<code>&lt;/tr&gt;</code>
187	<code>&lt;?php endif; ?&gt;</code>
188	<code>&lt;/tbody&gt;</code>
189	<code>&lt;/table&gt;</code>
190	<code>&lt;div class="d-flex justify-content-</code>
191	<code>center"&gt;</code>
192	<code>&lt;?= \$pager-&gt;links(); ?&gt;</code>
193	<code>&lt;/div&gt;</code>
194	<code>&lt;/div&gt;</code>
195	<code>&lt;/div&gt;</code>
196	<code>&lt;/div&gt;</code>
197	<code>&lt;/div&gt;</code>
198	
199	<code>&lt;?= \$this-&gt;endSection() ?&gt;</code>
200	
201	<code>&lt;?= \$this-&gt;section('extra-js') ?&gt;</code>
202	<code>&lt;script&gt;</code>
203	<code>\$(document).ready(function() {</code>
204	<code>\$('.pagination li').addClass('page-item');</code>
205	<code>\$('.pagination li a').addClass('page-link');</code>
206	<code>});</code>
207	<code>&lt;/script&gt;</code>
208	<code>&lt;?= \$this-&gt;endSection() ?&gt;</code>

*Tabel 9. MODUL 6 Source Code vw\_buku.php*

1	<code>&lt;?= \$this-&gt;extend('layouts/main') ?&gt;</code>
2	
3	<code>&lt;?= \$this-&gt;section('content') ?&gt;</code>
4	<code>&lt;style&gt;</code>
5	<code>body {</code>
6	<code>background-image:</code>
7	<code>url('https://images.unsplash.com/photo-1661097410573-</code>
8	<code>16d926ec1ec7?q=80&amp;w=2070&amp;auto=format&amp;fit=crop&amp;ixlib=rb-</code>
9	<code>4.0.3&amp;ixid=M3wxMjA3fDB8MHxwaG90bylwYWdlfHx8fGVufDB8fHx8f</code>
10	<code>A%3D%3D');</code>
11	<code>background-repeat: no-repeat;</code>
12	<code>background-attachment: fixed;</code>
13	<code>background-position: center;</code>
14	<code>background-size: cover;</code>
15	<code>color: #ececec;</code>
	<code>}</code>
	<code>.card {</code>
	<code>background: #1f1f1f;</code>



```
16         color: #ececec;
17         box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
18         border-radius: 10px;
19     }
20
21     .card-header {
22         background-color: #00bfa5;
23         border-bottom: 1px solid #37474f;
24     }
25
26     .btn-success {
27         background-color: #00bfa5;
28         border: none;
29     }
30
31     .btn-success:hover {
32         background-color: #008e76;
33     }
34
35     .btn-secondary {
36         background-color: #000000;
37         border: none;
38     }
39
40     .btn-secondary:hover {
41         background-color: #333333;
42     }
43
44     .alert {
45         background-color: rgba(255, 138, 101, 0.1);
46         color: #ffab40;
47         border: 1px solid #ffab40;
48     }
49
50     .form-control {
51         background-color: #2e2e2e;
52         color: #ececec;
53         border: 1px solid #37474f;
54     }
55
56     .form-control::placeholder {
57         color: #bdbdbd;
58     }
59
60     label {
61         color: #ececec;
62     }
```

```

63 </style>
64
65 <div class="container mt-5">
66     <div class="row justify-content-center">
67         <div class="col-md-8">
68             <div class="card shadow-sm">
69                 <div class="card-header text-white">
70                     <h4>Edit</h4>
71                 </div>
72                 <div class="card-body">
73
74                     <?php          if          (session()-
75 >getFlashdata('error')) : ?>
76                         <div class="alert alert-danger">
77                             <?=          session()-
78 >getFlashdata('error'); ?>
79                             </div>
80                         <?php endif; ?>
81
82                         <?= validation_list_errors() ?>
83
84                         <?=      form_open('buku/update/'      .
85 $post['id']); ?>
86                         <input type="hidden" name="_method"
87 value="PUT">
88
89                         <div class="form-group">
90                             <label for="judul">Judul</label>
91                             <input type="text" class="form-
92 control" id="judul" name="judul" placeholder="Masukkan
93 Judul" value="<?= $post['judul'] ?>">
94                             </div>
95                             <div class="form-group">
96                                 <label
97 for="penulis">Penulis</label>
98                                 <input type="text" class="form-
99 control"
100 id="penulis"          name="penulis"
101 placeholder="Masukkan      Penulis"          value="<?=
102 $post['penulis'] ?>">
103                                 </div>
104                                 <div class="form-group">
105                                     <label
106 for="penerbit">Penerbit</label>
107                                     <input type="text" class="form-
108 control"
109 id="penerbit"          name="penerbit"
110 placeholder="Masukkan      Penerbit"          value="<?=
111 $post['penerbit'] ?>">

```

96	<pre>         &lt;/div&gt;         &lt;div class="form-group"&gt;             &lt;label    for="tahun_terbit"&gt;Tahun Terbit&lt;/label&gt;             &lt;input class="form-control" id="tahun_terbit" name="tahun_terbit" placeholder="Masukkan Tahun Terbit" value="&lt;?= \$post['tahun_terbit'] ?&gt;"&gt;         &lt;/div&gt;         &lt;div class="form-group text-right"&gt;             &lt;button    class="btn    btn- success"&gt;Update&lt;/button&gt;             &lt;a    href="&lt;?=    base_url('home') ?&gt;" class="btn btn-secondary"&gt;Kembali&lt;/a&gt;         &lt;/div&gt;         &lt;?= form_close(); ?&gt;     &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;?= \$this-&gt;endSection() ?&gt; &lt;?= \$this-&gt;section('extra-js') ?&gt; &lt;!-- include summernote css/js --&gt; &lt;link href="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dis t/summernote-bs4.min.css" rel="stylesheet"&gt; &lt;script src="https://cdn.jsdelivrivr.net/npm/summernote@0.8.15/dist /summernote-bs4.min.js"&gt;&lt;/script&gt; &lt;script&gt;     \$(document).ready(function() {         \$('#post_content').summernote({             tabsize: 2,             height: 500         });     }); &lt;/script&gt; &lt;?= \$this-&gt;endSection() ?&gt; </pre>
----	---

*Tabel 10. MODUL 6 Source Code vw\_edit.php*

1	<pre>&lt;!DOCTYPE html&gt;</pre>
2	<pre>&lt;html lang="en"&gt;</pre>
3	
4	<pre>&lt;head&gt;</pre>

5	<meta charset="UTF-8">
6	<meta name="viewport" content="width=device-width, initial-scale=1.0">
7	<title>Login</title>
8	<style>
9	body {
10	display: flex;
11	justify-content: center;
12	align-items: center;
13	flex-direction: column;
14	min-height: 100vh;
15	padding: 0;
16	margin: 0;
17	font-family: Arial, sans-serif;
18	background-color: #121212;
19	color: #ecec;
20	position: relative;
21	overflow: hidden;
22	}
23	
24	body::before {
25	content: "";
26	position: absolute;
27	top: 0;
28	left: 0;
29	width: 100%;
30	height: 100%;
31	background-image:
	url('https://iispsm.sch.id/new/wp-content/uploads/2021/06/library.jpg');
32	background-size: cover;
33	background-position: center;
34	opacity: 0.1; /* Set the opacity to 20% */
35	z-index: 1;
36	}
37	
38	.welcome-message,
39	.form-signin {
40	z-index: 2;
41	position: relative;
42	}
43	
44	.welcome-message {
45	width: 100%;
46	max-width: 600px;
47	padding: 20px;
48	text-align: center;

49	margin-bottom: 20px;
50	background: #1f1f1f;
51	border-radius: 10px;
52	box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
53	color: #00bfa5;
54	font-size: 24px;
55	}
56	
57	.form-signin {
58	width: 100%;
59	max-width: 330px;
60	padding: 20px;
61	margin: auto;
62	background: #1f1f1f;
63	border-radius: 10px;
64	box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
65	position: relative;
66	}
67	
68	.form-signin img {
69	margin-bottom: 20px;
70	}
71	
72	.form-signin h1 {
73	margin-bottom: 10px;
74	font-size: 24px;
75	font-weight: normal;
76	color: #00bfa5;
77	}
78	
79	.form-signin p {
80	margin-bottom: 20px;
81	font-size: 14px;
82	color: #b0bec5;
83	}
84	
85	.form-signin input[type="text"],
86	.form-signin input[type="password"] {
87	width: 100%;
88	padding: 10px;
89	margin-bottom: 10px;
90	border: 1px solid #37474f;
91	border-radius: 4px;
92	background: #263238;
93	color: #ececec;
94	box-sizing: border-box;
95	}

96	
97	.form-signin input[type="text"]::placeholder,
98	.form-signin input[type="password"]::placeholder
	{
99	color: #90a4ae;
100	}
101	
102	.form-signin button {
103	width: 100%;
104	padding: 10px;
105	background-color: #00bfa5;
106	color: white;
107	border: none;
108	border-radius: 4px;
109	cursor: pointer;
110	font-size: 16px;
111	}
112	
113	.form-signin button:hover {
114	background-color: #008e76;
115	}
116	
117	.form-signin .alert {
118	padding: 10px;
119	margin-bottom: 10px;
120	border-radius: 4px;
121	color: #ffab40;
122	background-color: rgba(255, 138, 101, 0.1);
123	border: 1px solid #ffab40;
124	display: none;
125	position: relative;
126	}
127	
128	.form-signin .text-muted {
129	color: #b0bec5;
130	margin-top: 20px;
131	}
132	
133	.btn-register {
134	display: block;
135	width: 93%;
136	text-align: center;
137	padding: 10px;
138	margin-top: 10px;
139	background-color: #37474f;
140	color: #00bfa5;
141	border: 1px solid #00bfa5;

```

142         border-radius: 4px;
143         cursor: pointer;
144         font-size: 14px;
145         text-decoration: none;
146     }
147
148     .btn-register:hover {
149         background-color: #00bfa5;
150         color: #121212;
151     }
152 </style>
153 </head>
154
155 <body>
156
157     <div class="welcome-message">
158         Selamat Datang di Perpustakaan Bersama
159     </div>
160
161     <main class="form-signin">
162         <!-- Alert untuk informasi saat halaman dimuat -
163         ->
164         <div class="alert" id="initial-alert"
165         style="display: block;">
166             Login terlebih dahulu!
167         </div>
168
169         <!-- Alert untuk kesalahan login -->
170         <div class="alert" style="display: <?php echo
171         (session()->getFlashdata('error')) ? 'block' : 'none';
172         ?> ">
173             <?= session()->getFlashdata('error'); ?>
174         </div>
175
176         <form method="post" action="<?= base_url();
177         ?>/login/process">
178             <?= csrf_field(); ?>
179             <h1 class="h3 mb-3 fw-normal">Login</h1>
180             <p>Register dulu apabila belum memiliki
            akun</p>
            <input type="text" name="username"
            id="username" placeholder="Username" required autofocus>
            <input type="password" name="password"
            id="password" placeholder="Password" required>
            <button type="submit">Login</button>

```

181	</form>	
182	<a                class="btn-register"                        href="<?="	
183	base_url('register'); ?>">Register</a>	
184	</main>	
185	</body>	
186		
187	</html>	

*Tabel 11. MODUL 6 Source Code vw\_login.php*

1	<!DOCTYPE html>
2	<html lang="en" class="h-100">
3	
4	<head>
5	<meta charset="utf-8">
6	<meta name="viewport" content="width=device-width, initial-scale=1.0">
7	<title>Register</title>
8	<style>
9	body,
10	html {
11	margin: 0;
12	padding: 0;
13	font-family: Arial, sans-serif;
14	height: 100%;
15	display: flex;
16	justify-content: center;
17	align-items: center;
18	flex-direction: column;
19	background-color: #121212;
20	color: #ececec;
21	}
22	
23	body::before {
24	content: "";
25	position: absolute;
26	top: 0;
27	left: 0;
28	width: 100%;
29	height: 100%;
30	background-image:
	url('https://iispsm.sch.id/new/wp-content/uploads/2021/06/library.jpg');
31	background-size: cover;
32	background-position: center;
33	opacity: 0.1;
34	z-index: 1;
35	}



```
36
37     .welcome-message,
38     .form-register {
39         z-index: 2;
40         position: relative;
41     }
42
43     .welcome-message {
44         width: 100%;
45         max-width: 600px;
46         padding: 20px;
47         text-align: center;
48         margin-bottom: 20px;
49         background: #1f1f1f;
50         border-radius: 10px;
51         box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
52         color: #00bfa5;
53         font-size: 24px;
54     }
55
56     .form-register {
57         width: 100%;
58         max-width: 330px;
59         padding: 20px;
60         margin: auto;
61         background: #1f1f1f;
62         border-radius: 10px;
63         box-shadow: 0 6px 10px rgba(0, 0, 0, 0.15);
64         position: relative;
65     }
66
67     .form-register h1 {
68         margin-bottom: 10px;
69         font-size: 24px;
70         font-weight: normal;
71         color: #00bfa5;
72     }
73
74     .form-register p {
75         margin-bottom: 20px;
76         font-size: 14px;
77         color: #b0bec5;
78     }
79
80     .form-register input[type="text"],
81     .form-register input[type="password"],
82     .form-register input[type="email"] {
```

```

83         width: 100%;
84         padding: 10px;
85         margin-bottom: 10px;
86         border: 1px solid #37474f;
87         border-radius: 4px;
88         background: #263238;
89         color: #ececec;
90         box-sizing: border-box;
91     }
92
93     .form-register input[type="text"]::placeholder,
94     .form-register
95     input[type="password"]::placeholder,
96     .form-register input[type="email"]::placeholder
97     {
98
99         color: #90a4ae;
100     }
101
102     .form-register button {
103         width: 100%;
104         padding: 10px;
105         background-color: #00bfa5;
106         color: white;
107         border: none;
108         border-radius: 4px;
109         cursor: pointer;
110         font-size: 16px;
111     }
112
113     .form-register button:hover {
114         background-color: #008e76;
115     }
116
117     .form-register .alert {
118         padding: 10px;
119         margin-bottom: 10px;
120         border-radius: 4px;
121         color: #ffab40;
122         background-color: rgba(255, 138, 101, 0.1);
123         border: 1px solid #ffab40;
124         display: none;
125         position: relative;
126     }
127
128     .form-register .text-muted {
129         color: #b0bec5;
130         margin-top: 20px;

```

```

128     }
129     </style>
130 </head>
131
132 <body>
133
134     <div class="welcome-message">
135         Form Pendaftaran
136     </div>
137
138     <main class="form-register">
139         <!-- Alert untuk informasi saat halaman dimuat -
140         ->
141         <div class="alert" id="initial-alert"
142         style="display: block;">
143             Isi form untuk mendaftar!
144         </div>
145
146         <!-- Alert untuk kesalahan register -->
147         <div class="alert" style="display: <?php echo
148         (session()->getFlashdata('error')) ? 'block' : 'none';
149         ?>;">
150             <?= session()->getFlashdata('error'); ?>
151         </div>
152
153         <form method="post" action="<?= base_url();
154         ?>/register/process">
155             <?= csrf_field(); ?>
156             <h1 class="h3 mb-3 fw-normal">Register</h1>
157             <p>Isi data di bawah untuk membuat akun
158             baru</p>
159
160             <input type="text" name="username"
161             id="username" placeholder="Username" required autofocus>
162             <input type="password" name="password"
163             id="password" placeholder="Password" required>
164             <input type="password" name="password_conf"
165             id="password_conf" placeholder="Confirm Password"
166             required>
167             <input type="email" name="email" id="email"
168             placeholder="Email" required>
169
170             <button type="submit">Register</button>
171         </form>
172     </main>
173 </body>

```

164	
165	</html>

Tabel 12. MODUL 6 Source Code vw\_register.php

```

1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  class BukuModel extends Model
8  {
9      protected $table = 'buku';
10     protected $allowedFields = ['judul', 'penulis' ,
'penerbit', 'tahun_terbit'];
11     protected $createdField = 'created_at';
12     protected $updatedField = 'updated_at';
13     protected $deletedField = 'deleted_at';
14
15     protected $validationRules = ([
16         'judul' => [
17             'rules' => 'required|string',
18             'errors' => [
19                 'required' => 'Judul harus diisi.',
20                 'string' => 'Judul harus berupa
string.'
21             ]
22         ],
23         'penulis' => [
24             'rules' => 'required|string',
25             'errors' => [
26                 'required' => 'Penulis harus diisi.',
27                 'string' => 'Penulis harus berupa
string.'
28             ]
29         ],
30         'penerbit' => [
31             'rules' => 'required|string',
32             'errors' => [
33                 'required' => 'Penerbit harus diisi.',
34                 'string' => 'Penerbit harus berupa
string.'
35             ]
36         ],
37         'tahun_terbit' => [
38             'rules' => =>
'required|integer|greater_than[1800]|less_than[2024]',
39             'errors' => [

```

40	<code>        'required'          =&gt; 'Tahun terbit harus diisi.',</code>
41	<code>        'integer'           =&gt; 'Tahun terbit harus berupa angka.',</code>
42	<code>        'greater_than'      =&gt; 'Tahun terbit harus lebih besar dari 1800.',</code>
43	<code>        'less_than'         =&gt; 'Tahun terbit harus lebih kecil dari 2024.'</code>
44	<code>    ]</code>
45	<code>    ],</code>
46	<code>    );</code>
47	
48	<code>    protected \$skipValidation = false;</code>
49	
50	<code>}</code>

*Tabel 13. MODUL 6 Source Code BukuModel.php*

1	<code>&lt;?php</code>
2	
3	<code>namespace App\Models;</code>
4	
5	<code>use CodeIgniter\Model;</code>
6	
7	<code>class UsersModel extends Model</code>
8	<code>{</code>
9	<code>    protected \$table = "users";</code>
10	<code>    protected \$returnType = "object";</code>
11	<code>    protected \$useTimestamps = true;</code>
12	<code>    protected \$allowedFields = ['username', 'password', 'email'];</code>
13	<code>}</code>

*Tabel 14. MODUL 6 Source Code UserModel.php*

1	<code>&lt;?php</code>
2	
3	<code>namespace App\Database\Migrations;</code>
4	
5	<code>use CodeIgniter\Database\Migration;</code>
6	
7	<code>class Users extends Migration</code>
8	<code>{</code>
9	<code>    public function up()</code>
10	<code>    {</code>
11	<code>        \$this-&gt;forge-&gt;addField([</code>
12	<code>            'id' =&gt; [</code>
13	<code>                'type' =&gt; 'INT',</code>
14	<code>                'unsigned' =&gt; TRUE,</code>
15	<code>                'auto increment' =&gt; TRUE</code>

```

16         ],
17         'username' => [
18             'type' => 'VARCHAR',
19             'constraint' => 255,
20             'null' => FALSE,
21         ],
22         'email' => [
23             'type' => 'VARCHAR',
24             'constraint' => 255,
25             'null' => FALSE,
26         ],
27         'password' => [
28             'type' => 'TEXT',
29             'null' => FALSE,
30         ],
31         'created_at' => [
32             'type' => 'datetime',
33             'null' => TRUE
34         ],
35         'updated_at' => [
36             'type' => 'datetime',
37             'null' => TRUE
38         ]
39     ];
40
41     $this->forge->addKey('id', TRUE);
42     $this->forge->createTable('users');
43 }
44
45 //-----
46 -----
47
48 public function down()
49 {
50     $this->forge->dropTable('users');
51 }

```

*Tabel 15. MODUL 6 Source Code 2024-06-03-103310\_Users.php*

```

1  <?php
2
3  namespace App\Database\Migrations;
4
5  use CodeIgniter\Database\Migration;
6
7  class Buku extends Migration
8  {
9      public function up()

```

10	{
11	\$this->forge->addField([
12	'id' => [
13	'type' => 'BIGINT',
14	'unsigned' => TRUE,
15	'auto_increment' => TRUE
16	],
17	'judul' => [
18	'type' => 'VARCHAR',
19	'constraint' => 255,
20	'null' => FALSE,
21	],
22	'penulis' => [
23	'type' => 'VARCHAR',
24	'constraint' => 255,
25	'null' => FALSE,
26	],
27	'penerbit' => [
28	'type' => 'VARCHAR',
29	'constraint' => 255,
30	'null' => FALSE,
31	],
32	'tahun_terbit' => [
33	'type' => 'YEAR',
34	'null' => TRUE
35	]
36	]);
37	
38	\$this->forge->addKey('id', TRUE);
39	\$this->forge->createTable('buku');
40	}
41	
42	public function down()
43	{
44	\$this->forge->dropTable('buku');
45	}
46	}

*Tabel 16. MODUL 6 Source Code 2024-06-03-122804\_Buku.php*

1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;
6	
7	class Buku extends Seeder
8	{
9	public function run()

10	{
11	\$this->db->table('buku')->insert([
12	'judul' => 'Laut Bercerita',
13	'penulis' => 'Leila S. Chudori',
14	'penerbit' => 'Gramedia',
15	'tahun_terbit' => '2017'
16	]);
17	
18	\$this->db->table('buku')->insert([
19	'judul' => 'Tentang Kamu',
20	'penulis' => 'Tere Liye',
21	'penerbit' => 'Republika',
22	'tahun_terbit' => '2016'
23	]);
24	}
25	}

*Tabel 17. MODUL 6 Source Code Seeds Buku.php*

1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;
6	
7	class UserSeeder extends Seeder
8	{
9	public function run()
10	{
11	\$this->db->table('users')->insert([
12	'username' => 'noraina',
13	'email' => 'norainaevayantii@gmail.com',
14	'password' => password_hash('123',
15	PASSWORD_DEFAULT),
16	]);
17	}
18	}

*Tabel 18. MODUL 6 Source Code UserSeeder.php*

1	-- phpMyAdmin SQL Dump
2	-- version 5.2.1
3	-- https://www.phpmyadmin.net/
4	--
5	-- Host: 127.0.0.1
6	-- Waktu pembuatan: 11 Jun 2024 pada 19.57
7	-- Versi server: 10.4.32-MariaDB
8	-- Versi PHP: 8.2.12
9	
10	SET SQL MODE = "NO AUTO VALUE ON ZERO";



```

11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET
   @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*!40101 SET
   @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*!40101 SET
   @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Database: `perpustakaan`
22 --
23
24 -- -----
   ---
25
26 --
27 -- Struktur dari tabel `buku`
28 --
29
30 CREATE TABLE `buku` (
31   `id` bigint(20) UNSIGNED NOT NULL,
32   `judul` varchar(255) NOT NULL,
33   `penulis` varchar(255) NOT NULL,
34   `penerbit` varchar(255) NOT NULL,
35   `tahun_terbit` year(4) DEFAULT NULL
36 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
   COLLATE=utf8mb4_general_ci;
37
38 --
39 -- Dumping data untuk tabel `buku`
40 --
41
42 INSERT INTO `buku` (`id`, `judul`, `penulis`,
   `penerbit`, `tahun_terbit`) VALUES
43 (15, 'Laut Bercerita', 'Leila S. Chudori', 'Gramedia',
   '2017'),
44 (16, 'Tentang Kamu', 'Tere Liye', 'Republika', '2016');
45
46 -- -----
   ---
47
48 --
49 -- Struktur dari tabel `migrations`

```

```

50  --
51
52  CREATE TABLE `migrations` (
53      `id` bigint(20) UNSIGNED NOT NULL,
54      `version` varchar(255) NOT NULL,
55      `class` varchar(255) NOT NULL,
56      `group` varchar(255) NOT NULL,
57      `namespace` varchar(255) NOT NULL,
58      `time` int(11) NOT NULL,
59      `batch` int(11) UNSIGNED NOT NULL
60  )          ENGINE=InnoDB          DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
61
62  --
63  -- Dumping data untuk tabel `migrations`
64  --
65
66  INSERT INTO `migrations` (`id`, `version`, `class`,
67      `group`, `namespace`, `time`, `batch`) VALUES
68      (1, '2024-06-03-103310',
69      'App\\Database\\Migrations\\Users', 'default', 'App',
1718081782, 1),
70      (2, '2024-06-03-122804',
71      'App\\Database\\Migrations\\Buku', 'default', 'App',
1718081782, 1),
72      (4, '2024-06-11-173406',
73      'App\\Database\\Migrations\\Users', 'default', 'App',
1718127395, 3);
74
75  -- -----
76  ---
77
78  --
79  -- Struktur dari tabel `users`
80  --
81
82  CREATE TABLE `users` (
83      `id` int(10) UNSIGNED NOT NULL,
84      `username` varchar(255) NOT NULL,
85      `email` varchar(255) NOT NULL,
86      `password` text NOT NULL,
87      `created_at` datetime DEFAULT NULL,
88      `updated_at` datetime DEFAULT NULL
89  )          ENGINE=InnoDB          DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
90
91  --

```

```

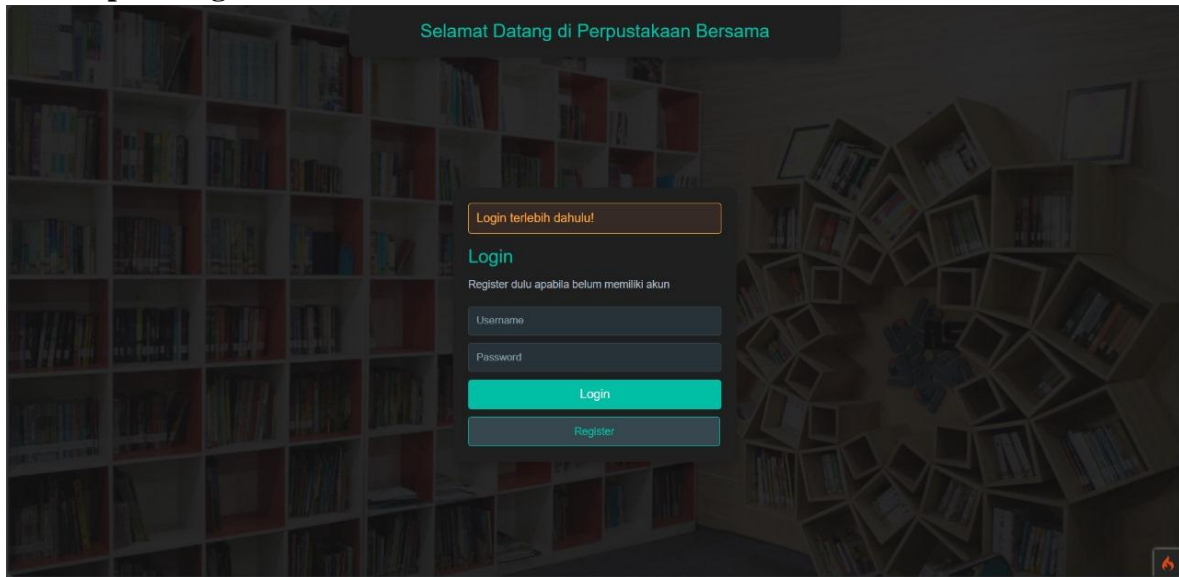
87  -- Dumping data untuk tabel `users`
88  --
89
90  INSERT INTO `users` (`id`, `username`, `email`,
91  `password`, `created_at`, `updated_at`) VALUES
92  (4, 'noraina', 'norainaevayantii@gmail.com',
93  '$2y$10$xYhsftGqil32/1Y8Qrypge2VGyR2bS0TEBFpXTa4sXUKI4Gw
94  nEoh2', NULL, NULL);
95
96  --
97  -- Indexes for dumped tables
98  --
99  -- Indeks untuk tabel `buku`
100 --
101 ALTER TABLE `buku`
102     ADD PRIMARY KEY (`id`);
103
104 --
105 -- Indeks untuk tabel `migrations`
106 --
107 ALTER TABLE `migrations`
108     ADD PRIMARY KEY (`id`);
109
110 --
111 -- Indeks untuk tabel `users`
112 --
113 ALTER TABLE `users`
114     ADD PRIMARY KEY (`id`),
115     ADD UNIQUE KEY `email` (`email`);
116
117 --
118 -- AUTO_INCREMENT untuk tabel yang dibuang
119 --
120 --
121 -- AUTO_INCREMENT untuk tabel `buku`
122 --
123 ALTER TABLE `buku`
124     MODIFY `id` bigint(20) UNSIGNED NOT NULL
125     AUTO_INCREMENT, AUTO_INCREMENT=19;
126
127 --
128 -- AUTO_INCREMENT untuk tabel `migrations`
129 --
130 ALTER TABLE `migrations`

```

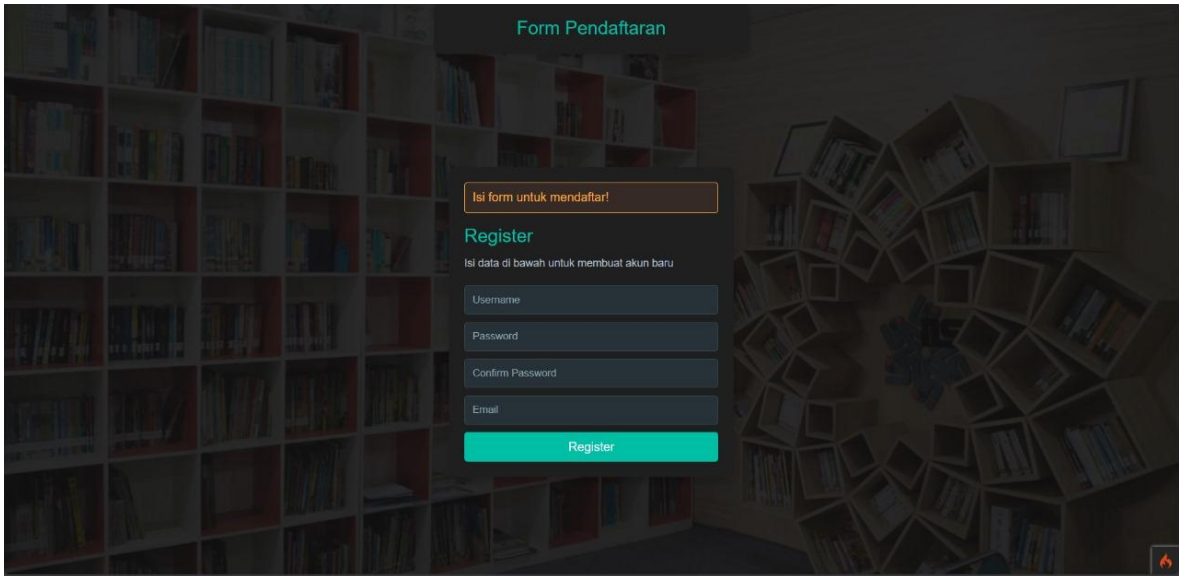
130	MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;	
131		
132	--	
133	-- AUTO_INCREMENT untuk tabel `users`	
134	--	
135	ALTER TABLE `users`	
136	MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;	
137	COMMIT;	
138		
139	/*!40101	SET
	CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;	
140	/*!40101	SET
	CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;	
141	/*!40101	SET
	COLLATION CONNECTION=@OLD_COLLATION CONNECTION */;	

*Tabel 19. MODUL 6 Source Code perpustakaan.sql*

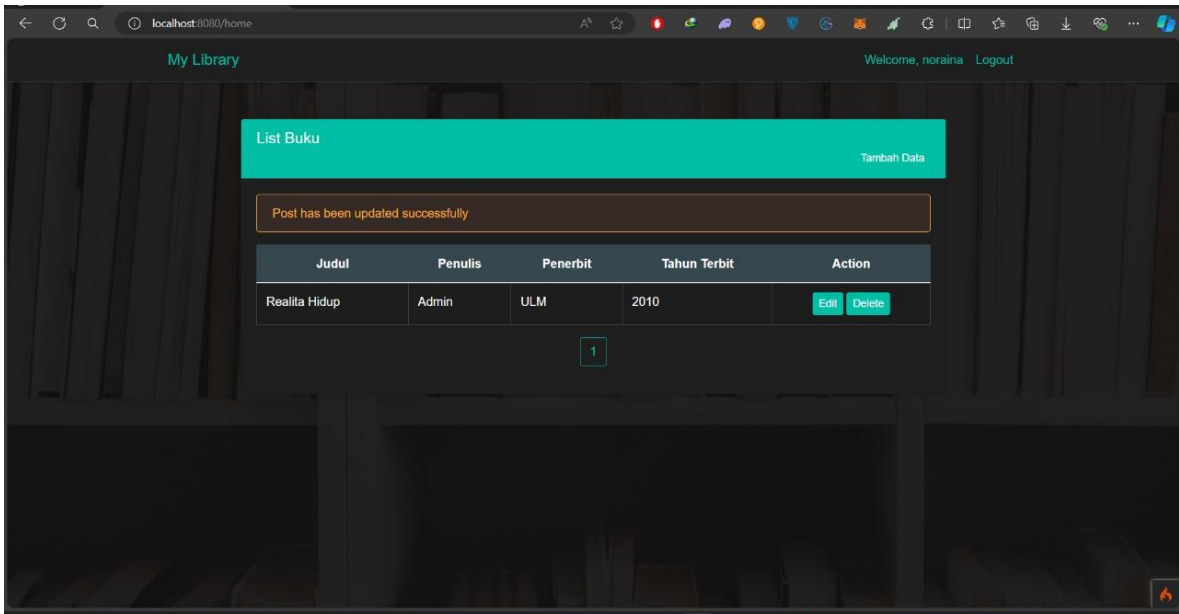
## B. Output Program



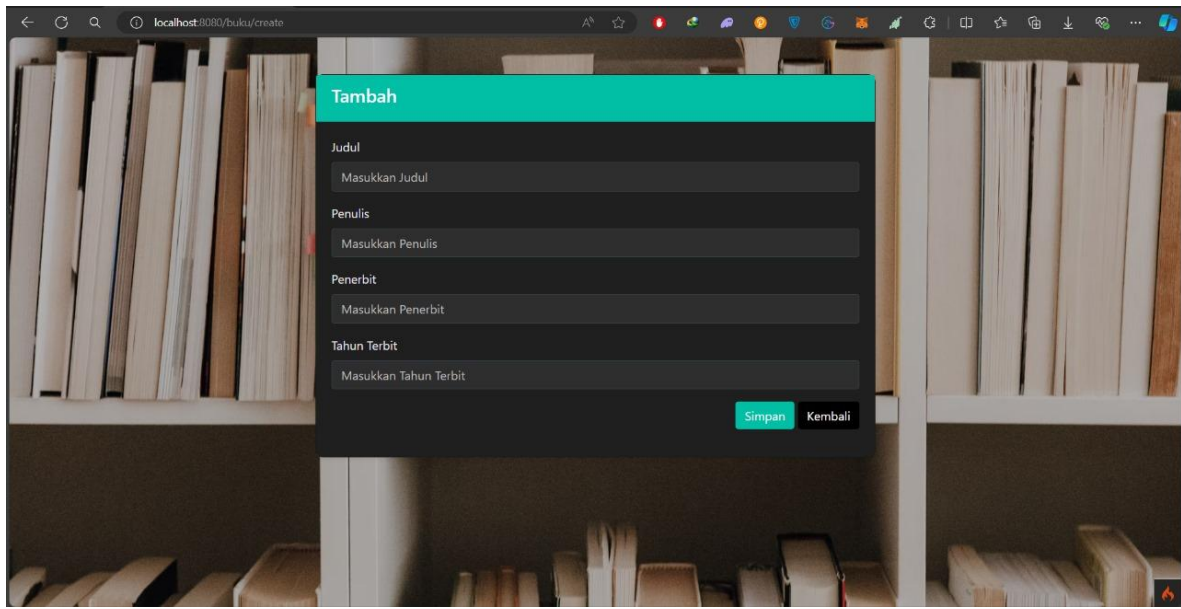
*Gambar 1. MODUL 6 Screenshot Hasil Jawaban Output 1*



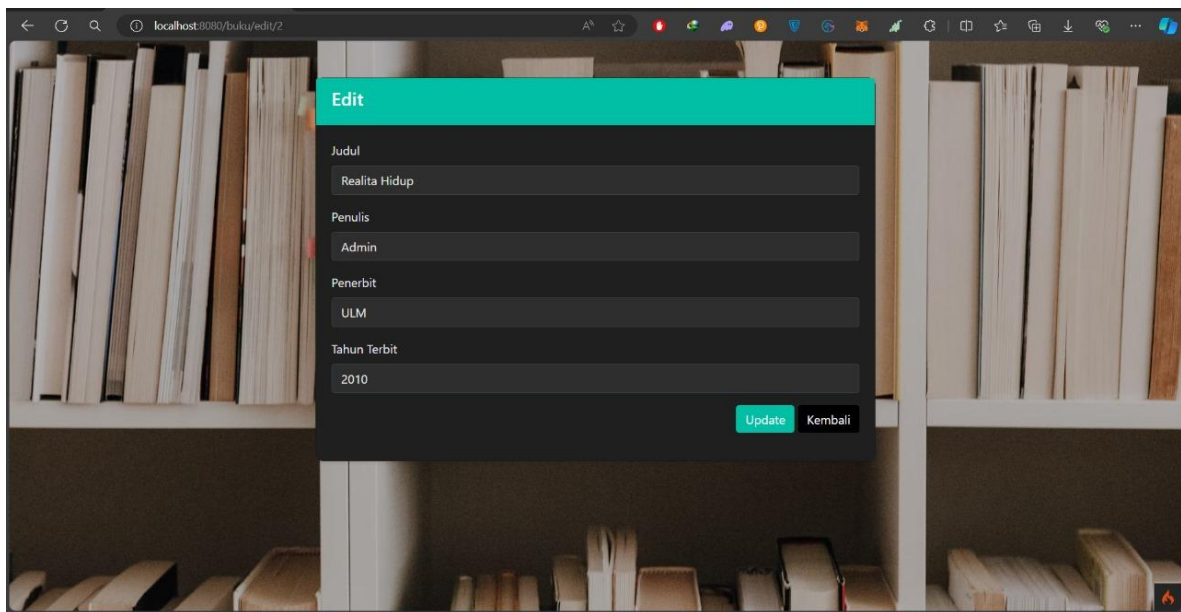
Gambar 2. MODUL 6 Screenshot Hasil Jawaban Output 2



Gambar 3. MODUL 6 Screenshot Hasil Jawaban Output 3



Gambar 4. MODUL 6 Screenshot Hasil Jawaban Output 4



Gambar 5. MODUL 6 Screenshot Hasil Jawaban Output 5

## C. Pembahasan

### File Routes.php

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `use CodeIgniter\Router\RouteCollection;` yang digunakan untuk mengimpor kelas `RouteCollection` dari namespace `CodeIgniter\Router`.

Pada baris [6] terdapat komentar `/** @var RouteCollection $routes */` yang memberikan informasi kepada IDE atau editor kode bahwa variabel `$routes` adalah instance dari kelas `RouteCollection`.

Pada baris [8] terdapat `$routes->setAutoRoute(true);` yang mengaktifkan fitur auto-routing pada CodeIgniter, sehingga rute dapat dibuat secara otomatis berdasarkan struktur folder dan nama file controller.

Pada baris [9] terdapat `$routes->get('/', 'Buku::login');` yang mendefinisikan rute untuk URL dasar (/) yang akan diarahkan ke metode login dari controller Buku.

Pada baris [10] terdapat `$routes->get('home', 'Buku::index');` yang mendefinisikan rute untuk URL home yang akan diarahkan ke metode index dari controller Buku.

Pada baris [11] terdapat `$routes->get('/register', 'Register::index');` yang mendefinisikan rute untuk URL register yang akan diarahkan ke metode index dari controller Register.

Pada baris [12] terdapat `$routes->post('/register/process', 'Register::process');` yang mendefinisikan rute POST untuk URL register/process yang akan diarahkan ke metode process dari controller Register.

Pada baris [13] terdapat `$routes->get('/login', 'Login::index');` yang mendefinisikan rute untuk URL login yang akan diarahkan ke metode index dari controller Login.

Pada baris [14] terdapat `$routes->post('/login/process', 'Login::process');` yang mendefinisikan rute POST untuk URL login/process yang akan diarahkan ke metode process dari controller Login.

Pada baris [15] terdapat `$routes->get('/logout', 'Login::logout');` yang mendefinisikan rute untuk URL logout yang akan diarahkan ke metode logout dari controller Login.

Pada baris [16] terdapat `$routes->get('/buku/create', 'Buku::create');` yang mendefinisikan rute untuk URL buku/create yang akan diarahkan ke metode create dari controller Buku.

Pada baris [17] terdapat `$routes->post('buku/store', 'Buku::store');` yang mendefinisikan rute POST untuk URL buku/store yang akan diarahkan ke metode store dari controller Buku.

Pada baris [18] terdapat `$routes->post('buku/update/', 'Buku::update', ['filter' => 'auth']);` yang mendefinisikan rute POST untuk URL buku/update/ yang akan diarahkan ke metode update dari controller Buku, dengan filter auth yang memastikan bahwa hanya pengguna yang terautentikasi yang dapat mengakses rute ini.

Pada baris [19] terdapat `$routes->get('buku/edit/', 'Buku::edit', ['filter' => 'auth'])`; yang mendefinisikan rute untuk URL buku/edit/ yang akan diarahkan ke metode edit dari controller Buku, dengan filter auth.

Pada baris [20] terdapat `$routes->get('buku/delete/', 'Buku::delete', ['filter' => 'auth'])`; yang mendefinisikan rute untuk URL buku/delete/ yang akan diarahkan ke metode delete dari controller Buku, dengan filter auth.

### **File BaseController.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Controllers;` yang mendefinisikan namespace untuk kelas ini sebagai App\Controllers, yang berarti kelas ini berada dalam folder App/Controllers.

Pada baris [5] terdapat `use CodeIgniter\Controller;` yang mengimpor kelas Controller dari framework CodeIgniter.

Pada baris [6-10] terdapat `use CodeIgniter\HTTP\CLIRequest;; use CodeIgniter\HTTP\IncomingRequest;; use CodeIgniter\HTTP\RequestInterface;; use CodeIgniter\HTTP\ResponseInterface;; dan use Psr\Log\LoggerInterface;` yang masing-masing mengimpor kelas yang dibutuhkan untuk menangani request, response, dan logging.

Pada baris [12-23] terdapat blok komentar `/** ... */` yang memberikan penjelasan tentang kelas BaseController. Blok komentar ini menjelaskan bahwa BaseController adalah tempat yang nyaman untuk memuat komponen dan melakukan fungsi yang diperlukan oleh semua controller, dan bahwa kelas ini harus diperluas oleh controller lain.

Pada baris [25] terdapat `abstract class BaseController extends Controller` yang mendefinisikan kelas BaseController sebagai kelas abstrak yang mewarisi kelas Controller.

Pada baris [28-33] terdapat properti `$request` yang merupakan instance dari objek Request. Properti ini digunakan untuk menyimpan request yang diterima oleh aplikasi. Tipe datanya dapat berupa CLIRequest atau IncomingRequest.

Pada baris [35-41] terdapat properti `$helpers` yang merupakan array yang berisi daftar nama helper yang akan dimuat secara otomatis ketika kelas ini diinstansiasi. Helper-helper ini akan tersedia untuk semua controller yang memperluas BaseController.



Pada baris [43-48] terdapat komentar mengenai properti yang harus dideklarasikan secara eksplisit untuk mencegah pembuatan properti dinamis, yang dideprekasikan di PHP 8.2. Contohnya adalah properti `$session` yang dikomentari.

Pada baris [50-58] terdapat metode `initController` yang digunakan untuk menginisialisasi controller. Metode ini menerima tiga parameter: `RequestInterface $request`, `ResponseInterface $response`, dan `LoggerInterface $logger`.

Pada baris [55] terdapat `parent::initController($request, $response, $logger);` yang memanggil metode `initController` dari kelas induk (`Controller`).

Pada baris [58] terdapat contoh penggunaan `preload` untuk model atau library seperti `$this->session = \Config\Services::session();`, yang dikomentari.

### **File Buku.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Controllers;` yang mendefinisikan namespace untuk kelas ini sebagai `App\Controllers`, yang berarti kelas ini berada dalam folder `App/Controllers`.

Pada baris [5] terdapat `use App\Models\BukuModel;` yang mengimpor kelas `BukuModel` dari namespace `App\Models`.

Pada baris [6] terdapat `use CodeIgniter\Controller;` yang mengimpor kelas `Controller` dari framework `CodeIgniter`.

Pada baris [7] terdapat `use App\Models\PostModel;` yang mengimpor kelas `PostModel` dari namespace `App\Models`.

Pada baris [9] terdapat deklarasi kelas `Buku` yang memperluas kelas `Controller`.

Pada baris [11-17] terdapat definisi konstruktor `__construct()`. Di dalamnya, `$this->model` diinisialisasi dengan instance `BukuModel` dan helper form dan url dimuat secara otomatis.

Pada baris [19-26] terdapat metode `login()` yang mengambil data dari model, mempaginasi hasilnya, dan mengirimkan data tersebut ke view `vw_login`.

Pada baris [28-36] terdapat metode `index()` yang pertama-tama memeriksa apakah pengguna telah masuk (terautentikasi) atau tidak. Jika tidak, pengguna akan diarahkan ke halaman `login`. Jika ya, data diambil dari model, dipaginasi, dan dikirimkan ke view `vw_buku`.

Pada baris [38-44] terdapat metode `create()` yang memeriksa apakah pengguna telah masuk. Jika tidak, pengguna akan diarahkan ke halaman login. Jika ya, halaman view `vw_add` akan ditampilkan dengan judul 'Create new post'.

Pada baris [46-60] terdapat metode `store()` yang mengambil data dari request POST, memvalidasinya, dan menyimpannya ke database menggunakan model. Jika data valid, pengguna akan diarahkan ke halaman home dengan pesan sukses. Jika tidak, halaman `create` akan ditampilkan lagi.

Pada baris [62-74] terdapat metode `edit($id)` yang memeriksa apakah pengguna telah masuk. Jika tidak, pengguna akan diarahkan ke halaman login. Jika ya, data dari model diambil berdasarkan ID, dan jika data ditemukan, halaman view `vw_edit` akan ditampilkan dengan data tersebut. Jika tidak, pengguna akan diarahkan kembali dengan pesan error.

Pada baris [76-97] terdapat metode `update($id)` yang mirip dengan metode `edit`, tetapi digunakan untuk memperbarui data. Data dari request POST divalidasi dan diperbarui di database. Jika berhasil, pengguna akan diarahkan ke halaman home dengan pesan sukses. Jika tidak, pengguna akan diarahkan kembali dengan pesan error.

Pada baris [99-114] terdapat metode `delete($id)` yang memeriksa apakah pengguna telah masuk. Jika tidak, pengguna akan diarahkan ke halaman home. Jika ya, data dihapus dari database berdasarkan ID. Jika berhasil, pengguna akan diarahkan ke halaman home dengan pesan sukses. Jika tidak, pengguna akan diarahkan kembali dengan pesan error.

### **File Home.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Controllers;` yang mendefinisikan namespace untuk kelas ini sebagai `App\Controllers`, yang berarti kelas ini berada dalam folder `App/Controllers`.

Pada baris [5] terdapat deklarasi kelas `Home` yang memperluas kelas `BaseController`. Dengan memperluas `BaseController`, kelas `Home` dapat mewarisi semua properti dan metode yang didefinisikan dalam `BaseController`.

Pada baris [7-10] terdapat metode `index()`. Metode ini:

Pada baris [8] mendefinisikan bahwa metode `index` mengembalikan tipe data string dengan menggunakan `: string`.

Pada baris [9] terdapat `return view('vw_buku');` yang mengembalikan hasil dari fungsi `view`. Fungsi `view` digunakan untuk memuat dan menampilkan view dengan nama

`vw_buku`. View ini adalah template yang akan ditampilkan ketika pengguna mengakses metode `index` dari kelas `Home`.

### **File Login.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Controllers;` yang mendefinisikan namespace untuk kelas ini sebagai `App\Controllers`, yang berarti kelas ini berada dalam folder `App/Controllers`.

Pada baris [5] terdapat `use App\Models\UsersModel;` yang mengimpor kelas `UsersModel` dari namespace `App\Models`.

Pada baris [7] terdapat deklarasi kelas `Login` yang memperluas kelas `BaseController`. Dengan memperluas `BaseController`, kelas `Login` dapat mewarisi semua properti dan metode yang didefinisikan dalam `BaseController`.

Pada baris [9-11] terdapat metode `index()` yang mengembalikan view `vw_login`. Metode ini digunakan untuk menampilkan halaman login ketika pengguna mengakses URL yang mengarah ke controller `Login`.

Pada baris [13-31] terdapat metode `process()` yang digunakan untuk memproses data login.

Pada baris [14] dibuat instance baru dari `UsersModel`.

Pada baris [15-16] nilai `username` dan `password` diambil dari request menggunakan metode `getVar`.

Pada baris [17] dilakukan pencarian data user berdasarkan `username`.

Pada baris [19] dilakukan pengecekan apakah data user ditemukan. Jika ditemukan, dilanjutkan ke langkah 5. Jika tidak, diberikan pesan error dan diarahkan kembali ke halaman login (baris [29-31]).

Pada baris [21] dilakukan verifikasi password menggunakan `password_verify`. Jika password valid, session diatur dengan `username` dan status `logged_in` dan pengguna diarahkan ke halaman home (baris [23-25]). Jika tidak valid, diberikan pesan error dan diarahkan kembali ke halaman login (baris [26-28]).

Pada baris [33-36] terdapat metode `logout()` yang digunakan untuk mengakhiri session pengguna. Metode ini:

Pada baris [34] memanggil `session()->destroy()` untuk menghancurkan session.

Pada baris [35] mengarahkan pengguna kembali ke halaman login.

### **File Register.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Controllers;` yang mendefinisikan namespace untuk kelas ini sebagai `App\Controllers`, yang berarti kelas ini berada dalam folder `App/Controllers`.

Pada baris [5] terdapat `use App\Models\UsersModel;` yang mengimpor kelas `UsersModel` dari namespace `App\Models`.

Pada baris [7] terdapat deklarasi kelas `Register` yang memperluas kelas `BaseController`. Dengan memperluas `BaseController`, kelas `Register` dapat mewarisi semua properti dan metode yang didefinisikan dalam `BaseController`.

Pada baris [9-11] terdapat metode `index()` yang mengembalikan view `vw_register`. Metode ini digunakan untuk menampilkan halaman registrasi ketika pengguna mengakses URL yang mengarah ke controller `Register`.

Pada baris [13-52] terdapat metode `process()` yang digunakan untuk memproses data registrasi.

Pada baris [15-46] terdapat blok validasi input menggunakan metode `validate()`. Setiap field (`username`, `password`, `password_conf`, `email`) memiliki aturan validasi dan pesan error khusus yang akan ditampilkan jika validasi gagal. Jika validasi gagal, pesan error disimpan dalam session dan pengguna diarahkan kembali ke halaman sebelumnya dengan input yang telah dimasukkan sebelumnya.

Pada baris [48] dibuat instance baru dari `UsersModel`.

Pada baris [51] dilakukan hashing pada `password` yang diterima dari input menggunakan `password_hash()` dengan algoritma `PASSWORD_DEFAULT`.

Pada baris [53-56] data pengguna baru (`username`, `password` yang sudah di-hash, dan `email`) disimpan ke dalam database menggunakan metode `insert()` dari model `UsersModel`.

Pada baris [58] pengguna diarahkan ke halaman login setelah berhasil mendaftar.

### **File AuthFilter.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat komentar `// app/Filters/AuthFilter.php` yang menunjukkan lokasi file filter ini dalam struktur proyek.

Pada baris [5] terdapat namespace `App\Filters;` yang mendefinisikan namespace untuk kelas ini sebagai `App\Filters`, yang berarti kelas ini berada dalam folder `App/Filters`.

Pada baris [7-10] terdapat `use CodeIgniter\HTTP\RequestInterface;`, `use CodeIgniter\HTTP\ResponseInterface;`, dan `use CodeIgniter\Filters\FilterInterface;` yang mengimpor kelas-kelas yang diperlukan untuk menangani request, response, dan filter.

Pada baris [12] terdapat deklarasi kelas `AuthFilter` yang mengimplementasikan `FilterInterface`. Dengan mengimplementasikan `FilterInterface`, kelas ini harus mendefinisikan metode `before` dan `after`.

Pada baris [14-21] terdapat metode `before()` yang akan dijalankan sebelum request diproses lebih lanjut.

Pada baris [16] terdapat komentar `// Check if the user is logged in` yang menjelaskan bahwa kode di bawahnya akan memeriksa apakah pengguna sudah masuk atau belum.

Pada baris [17] dilakukan pengecekan apakah session memiliki `user_id`. Jika tidak, berarti pengguna belum masuk.

Pada baris [19] jika pengguna belum masuk, pengguna akan diarahkan ke halaman login menggunakan `redirect()->to('/login')`.

Pada baris [23-25] terdapat metode `after()` yang akan dijalankan setelah request diproses.

Pada baris [24] terdapat komentar `// Do nothing after the request` yang menjelaskan bahwa tidak ada tindakan yang dilakukan setelah request diproses.

### **File `vw_add.php`**

Pada baris [1] terdapat `<?= $this->extend('layouts/main') ?>` yang digunakan untuk memperluas (extend) template utama `layouts/main`. Artinya, template `layouts/main` akan digunakan sebagai kerangka dasar dari view ini.

Pada baris [3] terdapat `<?= $this->section('content') ?>` yang mendefinisikan bagian `content` dari template. Semua konten di dalam bagian ini akan dimasukkan ke dalam placeholder `content` yang ada di template `layouts/main`.

Pada baris [4-47] terdapat elemen `<style>` yang berisi gaya CSS yang digunakan untuk mengatur tampilan halaman.

Pada baris [5-11] terdapat aturan CSS untuk elemen `body`, yang menetapkan gambar latar belakang, pengulangan, posisi, dan ukuran.

Pada baris [13-16] terdapat aturan CSS untuk elemen `.card`, yang menetapkan latar belakang, warna teks, bayangan, dan radius border.

Pada baris [18-20] terdapat aturan CSS untuk elemen `.card-header`, yang menetapkan warna latar belakang dan border bawah.

Pada baris [22-24] terdapat aturan CSS untuk elemen `.btn-success`, yang menetapkan warna latar belakang dan border.

Pada baris [26-28] terdapat aturan CSS untuk elemen `.btn-success:hover`, yang menetapkan warna latar belakang saat tombol di-hover.

Pada baris [30-32] terdapat aturan CSS untuk elemen `.btn-secondary`, yang menetapkan warna latar belakang dan border.

Pada baris [34-36] terdapat aturan CSS untuk elemen `.btn-secondary:hover`, yang menetapkan warna latar belakang saat tombol di-hover.

Pada baris [38-41] terdapat aturan CSS untuk elemen `.alert`, yang menetapkan warna latar belakang, warna teks, dan border.

Pada baris [43-46] terdapat aturan CSS untuk elemen `.form-control`, yang menetapkan warna latar belakang, warna teks, dan border.

Pada baris [48] terdapat aturan CSS untuk elemen `.form-control::placeholder`, yang menetapkan warna placeholder.

Pada baris [50] terdapat aturan CSS untuk elemen `label`, yang menetapkan warna teks.

Pada baris [49-71] terdapat markup HTML yang mendefinisikan struktur halaman.

Pada baris [49] terdapat `<div class="container mt-5">` yang membuat container dengan margin atas 5.

Pada baris [50] terdapat `<div class="row justify-content-center">` yang membuat baris dengan konten yang diatur ke tengah secara horizontal.

Pada baris [51] terdapat `<div class="col-md-8">` yang membuat kolom dengan lebar 8 unit pada breakpoint medium.

Pada baris [52] terdapat `<div class="card shadow-sm">` yang membuat kartu dengan bayangan kecil.

Pada baris [53-55] terdapat `<div class="card-header text-white">` yang membuat header kartu dengan teks berwarna putih dan judul "Tambah".

Pada baris [56-70] terdapat `<div class="card-body">` yang berisi konten utama kartu.

Pada baris [57-60] terdapat blok PHP yang menampilkan pesan error jika ada.

Pada baris [62] terdapat `<?= validation_list_errors() ?>` yang menampilkan daftar error validasi jika ada.

Pada baris [64] terdapat `<?= form_open('buku/store'); ?>` yang membuka form dengan action buku/store.

Pada baris [65-68] terdapat grup form untuk input judul, penulis, penerbit, dan tahun\_terbit, masing-masing dengan label dan input teks/angka.

Pada baris [69-70] terdapat grup tombol untuk menyimpan form dan kembali ke halaman sebelumnya.

Pada baris [73] terdapat `<?= $this->endSection() ?>` yang menutup bagian content.

Pada baris [75] terdapat `<?= $this->section('extra-js') ?>` yang mendefinisikan bagian extra-js untuk menyertakan skrip JavaScript tambahan.

Pada baris [76-81] terdapat `<link>` dan `<script>` untuk menyertakan CSS dan JS dari Summernote, editor teks WYSIWYG.

Pada baris [82-87] terdapat skrip jQuery yang menginisialisasi Summernote pada elemen dengan ID `post_content`.

Pada baris [88] terdapat `<?= $this->endSection() ?>` yang menutup bagian extra-js.

### **File vw\_buku.php**

Pada baris [1] terdapat `<?= $this->extend('layouts/main') ?>` yang digunakan untuk memperluas (extend) template utama layouts/main. Artinya, template layouts/main akan digunakan sebagai kerangka dasar dari view ini.

Pada baris [3] terdapat `<?= $this->section('content') ?>` yang mendefinisikan bagian content dari template. Semua konten di dalam bagian ini akan dimasukkan ke dalam placeholder content yang ada di template layouts/main.

Pada baris [4-69] terdapat elemen `<style>` yang berisi gaya CSS yang digunakan untuk mengatur tampilan halaman.

Pada baris [5-18] terdapat aturan CSS untuk elemen `body` dan `html`, yang mengatur `margin`, `padding`, `font`, `tinggi`, `tampilan flex`, `warna latar belakang`, `warna teks`, `posisi`, dan `overflow`.

Pada baris [19-25] terdapat aturan CSS untuk elemen `body::before`, yang menetapkan `gambar latar belakang`, `ukuran`, `posisi`, `opasitas`, dan `z-index`.

Pada baris [26-30] terdapat aturan CSS untuk elemen `.navbar`, yang mengatur `z-index`, `posisi`, `warna latar belakang`, dan `border bawah`.

Pada baris [31-35] terdapat aturan CSS untuk elemen `.container` dan `.card`, yang mengatur `z-index` dan `posisi`.

Pada baris [36-43] terdapat aturan CSS untuk elemen `.navbar-brand` dan `.nav-link`, yang mengatur `warna teks`.

Pada baris [44-49] terdapat aturan CSS untuk elemen `.card`, yang mengatur `warna latar belakang`, `warna teks`, `bayangan`, dan `radius border`.

Pada baris [50-52] terdapat aturan CSS untuk elemen `.card-header`, yang mengatur `warna latar belakang` dan `border bawah`.

Pada baris [53-56] terdapat aturan CSS untuk elemen `.btn`, yang mengatur `warna latar belakang`, `warna teks`, dan `border`.

Pada baris [57-59] terdapat aturan CSS untuk elemen `.btn:hover`, yang mengatur `warna latar belakang` saat tombol di-hover.

Pada baris [60-62] terdapat aturan CSS untuk elemen `.table thead`, yang mengatur `warna latar belakang`.

Pada baris [63-67] terdapat aturan CSS untuk elemen `.table th` dan `.table td`, yang mengatur `border` dan `warna teks`.

Pada baris [68-73] terdapat aturan CSS untuk elemen `.pagination li a`, yang mengatur `warna latar belakang`, `warna teks`, dan `border`.

Pada baris [74-77] terdapat aturan CSS untuk elemen `.pagination li a:hover`, yang mengatur `warna latar belakang` dan `warna teks` saat di-hover.

Pada baris [78-81] terdapat aturan CSS untuk elemen `.alert`, yang mengatur `warna latar belakang`, `warna teks`, dan `border`.

Pada baris [83-105] terdapat markup HTML yang mendefinisikan struktur halaman.



Pada baris [83-102] terdapat elemen `<nav>` yang mendefinisikan navbar.

Pada baris [85-86] terdapat elemen container yang membungkus konten navbar.

Pada baris [87] terdapat elemen `navbar-brand` yang menampilkan tautan ke halaman utama.

Pada baris [88-90] terdapat elemen `navbar-toggler` yang berfungsi sebagai tombol untuk menampilkan atau menyembunyikan menu navbar pada layar kecil.

Pada baris [91-96] terdapat elemen `navbar-collapse` yang menampilkan menu navigasi, yang berisi item navigasi untuk login, register, dan logout tergantung pada status login pengguna.

Pada baris [106-165] terdapat markup HTML yang mendefinisikan konten utama halaman.

Pada baris [106-108] terdapat elemen container `mt-5` yang membungkus konten dengan margin atas 5.

Pada baris [109-111] terdapat elemen `row justify-content-center` yang mengatur konten ke tengah secara horizontal.

Pada baris [112-114] terdapat elemen `col-md-10` yang mengatur kolom dengan lebar 10 unit pada breakpoint medium.

Pada baris [115-121] terdapat elemen `card shadow-sm` yang mendefinisikan kartu dengan bayangan kecil, dan di dalamnya terdapat `card-header` dan tombol Tambah Data.

Pada baris [122-159] terdapat elemen `card-body` yang berisi konten utama kartu.

Pada baris [123-126] terdapat elemen `alert` untuk menampilkan pesan sukses jika ada.

Pada baris [127-130] terdapat elemen `alert` untuk menampilkan pesan error jika ada.

Pada baris [131-156] terdapat elemen `table` yang menampilkan daftar buku dengan kolom Judul, Penulis, Penerbit, Tahun Terbit, dan Action.

Pada baris [132-136] terdapat elemen `thead` yang mendefinisikan header tabel.

Pada baris [137-155] terdapat elemen `tbody` yang mendefinisikan konten tabel dengan data buku yang di-loop.

Pada baris [157-159] terdapat elemen `d-flex justify-content-center` yang menampilkan pagination.

Pada baris [167] terdapat `<?= $this->endSection() ?>` yang menutup bagian content.

Pada baris [169] terdapat `<?= $this->section('extra-js') ?>` yang mendefinisikan bagian `extra-js` untuk menyertakan skrip JavaScript tambahan.

Pada baris [170-175] terdapat skrip jQuery yang menambahkan kelas `page-item` dan `page-link` ke elemen pagination.

Pada baris [176] terdapat `<?= $this->endSection() ?>` yang menutup bagian `extra-js`.

### **File `vw_edit.php`**

Pada baris [1] terdapat `<?= $this->extend('layouts/main') ?>` yang digunakan untuk memperluas (`extend`) template utama `layouts/main`. Artinya, template `layouts/main` akan digunakan sebagai kerangka dasar dari view ini.

Pada baris [3] terdapat `<?= $this->section('content') ?>` yang mendefinisikan bagian content dari template. Semua konten di dalam bagian ini akan dimasukkan ke dalam placeholder content yang ada di template `layouts/main`.

Pada baris [4-48] terdapat elemen `<style>` yang berisi gaya CSS yang digunakan untuk mengatur tampilan halaman.

Pada baris [5-11] terdapat aturan CSS untuk elemen `body`, yang menetapkan gambar latar belakang, pengulangan, posisi, ukuran, dan warna teks.

Pada baris [13-17] terdapat aturan CSS untuk elemen `.card`, yang menetapkan warna latar belakang, warna teks, bayangan, dan radius border.

Pada baris [19-21] terdapat aturan CSS untuk elemen `.card-header`, yang menetapkan warna latar belakang dan border bawah.

Pada baris [23-25] terdapat aturan CSS untuk elemen `.btn-success`, yang menetapkan warna latar belakang dan border.

Pada baris [27-29] terdapat aturan CSS untuk elemen `.btn-success:hover`, yang menetapkan warna latar belakang saat tombol di-hover.

Pada baris [31-33] terdapat aturan CSS untuk elemen `.btn-secondary`, yang menetapkan warna latar belakang dan border.

Pada baris [35-37] terdapat aturan CSS untuk elemen `.btn-secondary:hover`, yang menetapkan warna latar belakang saat tombol di-hover.

Pada baris [39-42] terdapat aturan CSS untuk elemen `.alert`, yang menetapkan warna latar belakang, warna teks, dan border.

Pada baris [44-47] terdapat aturan CSS untuk elemen `.form-control`, yang menetapkan warna latar belakang, warna teks, dan border.

Pada baris [48] terdapat aturan CSS untuk elemen `.form-control::placeholder`, yang menetapkan warna placeholder.

Pada baris [50] terdapat aturan CSS untuk elemen `label`, yang menetapkan warna teks.

Pada baris [49-89] terdapat markup HTML yang mendefinisikan struktur halaman.

Pada baris [49] terdapat `<div class="container mt-5">` yang membuat container dengan margin atas 5.

Pada baris [50] terdapat `<div class="row justify-content-center">` yang membuat baris dengan konten yang diatur ke tengah secara horizontal.

Pada baris [51] terdapat `<div class="col-md-8">` yang membuat kolom dengan lebar 8 unit pada breakpoint medium.

Pada baris [52] terdapat `<div class="card shadow-sm">` yang membuat kartu dengan bayangan kecil.

Pada baris [53-55] terdapat `<div class="card-header text-white">` yang membuat header kartu dengan teks berwarna putih dan judul "Edit".

Pada baris [56-87] terdapat `<div class="card-body">` yang berisi konten utama kartu.

Pada baris [57-60] terdapat blok PHP yang menampilkan pesan error jika ada.

Pada baris [62] terdapat `<?= validation_list_errors() ?>` yang menampilkan daftar error validasi jika ada.

Pada baris [64] terdapat `<?= form_open('buku/update/' . $post['id']); ?>` yang membuka form dengan action `buku/update/{id}`.

Pada baris [65] terdapat input `hidden _method` dengan nilai `PUT` untuk metode `update`.

Pada baris [67-80] terdapat grup form untuk input judul, penulis, penerbit, dan tahun terbit, masing-masing dengan label, input teks/angka, dan nilai default dari data yang akan diedit.

Pada baris [81-82] terdapat grup tombol untuk menyimpan form dan kembali ke halaman sebelumnya.

Pada baris [91] terdapat `<?= $this->endSection() ?>` yang menutup bagian content.

Pada baris [93] terdapat `<?= $this->section('extra-js') ?>` yang mendefinisikan bagian extra-js untuk menyertakan skrip JavaScript tambahan.

Pada baris [94-96] terdapat `<link>` dan `<script>` untuk menyertakan CSS dan JS dari Summernote, editor teks WYSIWYG.

Pada baris [97-102] terdapat skrip jQuery yang menginisialisasi Summernote pada elemen dengan ID `post_content`.

Pada baris [103] terdapat `<?= $this->endSection() ?>` yang menutup bagian extra-js.

### **File vw\_login.php**

Pada baris [1] terdapat `<!DOCTYPE html>` yang mendefinisikan dokumen ini sebagai dokumen HTML5.

Pada baris [2] terdapat `<html lang="en">` yang mendefinisikan bahasa yang digunakan dalam dokumen HTML ini sebagai bahasa Inggris.

Pada baris [4] terdapat `<meta charset="UTF-8">` yang mendefinisikan karakter encoding yang digunakan pada halaman web ini sebagai UTF-8.

Pada baris [5] terdapat `<meta name="viewport" content="width=device-width, initial-scale=1.0">` yang mengatur tampilan dan skalabilitas halaman pada berbagai perangkat, seperti mobile dan desktop.

Pada baris [6] terdapat `<title>Login</title>` yang mendefinisikan judul halaman yang akan ditampilkan pada tab browser.

Pada baris [7-75] terdapat elemen `<style>` yang berisi gaya CSS yang digunakan untuk mengatur tampilan halaman login.

Pada baris [8-16] terdapat aturan CSS untuk elemen `body`, yang menetapkan tampilan flex, pengaturan margin dan padding, font, warna latar belakang, warna teks, posisi relatif, dan overflow.

Pada baris [18-24] terdapat aturan CSS untuk elemen `body::before`, yang menetapkan gambar latar belakang, ukuran, posisi, opasitas, dan z-index.

Pada baris [26-30] terdapat aturan CSS untuk elemen `.welcome-message` dan `.form-signin`, yang menetapkan z-index dan posisi relatif.

Pada baris [32-38] terdapat aturan CSS untuk elemen `.welcome-message`, yang menetapkan lebar maksimal, padding, teks align, margin bawah, warna

latar belakang, radius border, bayangan, warna teks, dan ukuran font.

Pada baris [40-47] terdapat aturan CSS untuk elemen `.form-signin`, yang menetapkan lebar maksimal, padding, margin, warna latar belakang, radius border, bayangan, dan posisi relatif.

Pada baris [49-51] terdapat aturan CSS untuk elemen `.form-signin img`, yang menetapkan margin bawah.

Pada baris [53-57] terdapat aturan CSS untuk elemen `.form-signin h1`, yang menetapkan margin bawah, ukuran font, font-weight, dan warna teks.

Pada baris [59-63] terdapat aturan CSS untuk elemen `.form-signin p`, yang menetapkan margin bawah, ukuran font, dan warna teks.

Pada baris [65-71] terdapat aturan CSS untuk elemen `input` (`input[type="text"]`, `input[type="password"]`), yang menetapkan lebar, padding, margin bawah, border, radius border, warna latar belakang, warna teks, dan box-sizing.

Pada baris [73-75] terdapat aturan CSS untuk elemen placeholder pada `input` (`::placeholder`), yang menetapkan warna teks placeholder.

Pada baris [77-82] terdapat aturan CSS untuk elemen tombol (`button`), yang menetapkan lebar, padding, warna latar belakang, warna teks, border, radius border, cursor, dan ukuran font.

Pada baris [84-86] terdapat aturan CSS untuk elemen tombol saat di-hover (`button:hover`), yang menetapkan warna latar belakang.

Pada baris [88-94] terdapat aturan CSS untuk elemen alert (`.alert`), yang menetapkan padding, margin bawah, radius border, warna teks, warna latar belakang, border, display, dan posisi relatif.

Pada baris [96-98] terdapat aturan CSS untuk elemen teks muted (`.text-muted`), yang menetapkan warna teks dan margin atas.

Pada baris [100-106] terdapat aturan CSS untuk elemen tombol register (`.btn-register`), yang menetapkan display, lebar, teks align, padding, margin atas, warna latar belakang, warna teks, border, radius border, cursor, ukuran font, dan text-decoration.

Pada baris [108-110] terdapat aturan CSS untuk elemen tombol register saat di-hover (`.btn-register:hover`), yang menetapkan warna latar belakang dan warna teks.

Pada baris [112-114] terdapat elemen `<div class="welcome-message">` yang menampilkan pesan selamat datang di perpustakaan bersama.

Pada baris [116-136] terdapat elemen `<main class="form-signin">` yang berisi form login.

Pada baris [117-120] terdapat elemen alert untuk informasi saat halaman dimuat.

Pada baris [122-125] terdapat elemen alert untuk kesalahan login.

Pada baris [127] terdapat elemen form dengan method POST dan action yang mengarah ke `login/process`.

Pada baris [128] terdapat elemen `<?= csrf_field(); ?>` untuk melindungi form dari CSRF.

Pada baris [129-131] terdapat elemen `<h1>` dan `<p>` yang menampilkan judul dan deskripsi form.

Pada baris [133-134] terdapat elemen `input` untuk username dan password dengan placeholder, atribut `required`, dan `autofocus` untuk input username.

Pada baris [136] terdapat elemen `tombol` untuk submit form.

Pada baris [137] terdapat elemen `anchor` untuk tombol register yang mengarah ke halaman register.

Pada baris [138-140] terdapat penutupan elemen `<body>` dan `<html>`.

### **File `vw_register.php`**

Pada baris [1] terdapat `<!DOCTYPE html>` yang mendefinisikan dokumen ini sebagai dokumen HTML5.

Pada baris [2] terdapat `<html lang="en" class="h-100">` yang mendefinisikan bahasa yang digunakan dalam dokumen HTML ini sebagai bahasa Inggris dan menambahkan kelas `h-100` untuk mengatur tinggi elemen HTML.

Pada baris [4] terdapat `<meta charset="utf-8">` yang mendefinisikan karakter encoding yang digunakan pada halaman web ini sebagai UTF-8.

Pada baris [5] terdapat `<meta name="viewport" content="width=device-width, initial-scale=1.0">` yang mengatur tampilan dan skalabilitas halaman pada berbagai perangkat, seperti mobile dan desktop.

Pada baris [6] terdapat `<title>Register</title>` yang mendefinisikan judul halaman yang akan ditampilkan pada tab browser.

Pada baris [7-79] terdapat elemen `<style>` yang berisi gaya CSS yang digunakan untuk mengatur tampilan halaman register.

Pada baris [8-16] terdapat aturan CSS untuk elemen `body` dan `html`, yang menetapkan `margin`, `padding`, `font`, `tinggi`, tampilan `flex`, pengaturan `justify-content` dan `align-items`, warna latar belakang, dan warna teks.

Pada baris [18-24] terdapat aturan CSS untuk elemen `body::before`, yang menetapkan gambar latar belakang, ukuran, posisi, opasitas, dan `z-index`.

Pada baris [26-30] terdapat aturan CSS untuk elemen `.welcome-message` dan `.form-register`, yang menetapkan `z-index` dan posisi relatif.

Pada baris [32-38] terdapat aturan CSS untuk elemen `.welcome-message`, yang menetapkan lebar maksimal, `padding`, teks `align`, margin bawah, warna latar belakang, radius border, bayangan, warna teks, dan ukuran font.

Pada baris [40-47] terdapat aturan CSS untuk elemen `.form-register`, yang menetapkan lebar maksimal, `padding`, margin, warna latar belakang, radius border, bayangan, dan posisi relatif.

Pada baris [49-51] terdapat aturan CSS untuk elemen `.form-register h1`, yang menetapkan margin bawah, ukuran font, `font-weight`, dan warna teks.

Pada baris [53-57] terdapat aturan CSS untuk elemen `.form-register p`, yang menetapkan margin bawah, ukuran font, dan warna teks.

Pada baris [59-67] terdapat aturan CSS untuk elemen `input` (`input[type="text"]`, `input[type="password"]`, `input[type="email"]`), yang menetapkan lebar, `padding`, margin bawah, border, radius border, warna latar belakang, warna teks, dan `box-sizing`.

Pada baris [69-71] terdapat aturan CSS untuk elemen placeholder pada `input` (`::placeholder`), yang menetapkan warna teks placeholder.

Pada baris [73-78] terdapat aturan CSS untuk elemen tombol (`button`), yang menetapkan lebar, `padding`, warna latar belakang, warna teks, border, radius border, cursor, dan ukuran font.

Pada baris [80-82] terdapat aturan CSS untuk elemen tombol saat di-hover (`button:hover`), yang menetapkan warna latar belakang.

Pada baris [84-90] terdapat aturan CSS untuk elemen alert (`.alert`), yang menetapkan `padding`, margin bawah, radius border, warna teks, warna latar belakang, border, display, dan posisi relatif.

Pada baris [92-94] terdapat aturan CSS untuk elemen teks muted (`.text-muted`), yang menetapkan warna teks dan margin atas.

Pada baris [96-98] terdapat elemen `<div class="welcome-message">` yang menampilkan pesan "Form Pendaftaran".

Pada baris [100-121] terdapat elemen `<main class="form-register">` yang berisi form register.

Pada baris [101-104] terdapat elemen alert untuk informasi saat halaman dimuat.

Pada baris [106-109] terdapat elemen alert untuk kesalahan register.

Pada baris [111] terdapat elemen form dengan method POST dan action yang mengarah ke `register/process`.

Pada baris [112] terdapat elemen `<?= csrf_field(); ?>` untuk melindungi form dari CSRF.

Pada baris [113-115] terdapat elemen `<h1>` dan `<p>` yang menampilkan judul dan deskripsi form.

Pada baris [117-120] terdapat elemen input untuk username, password, konfirmasi password, dan email dengan placeholder, atribut required, dan autofocus untuk input username.

Pada baris [122] terdapat elemen tombol untuk submit form.

Pada baris [124-126] terdapat penutupan elemen `<body>` dan `<html>`.

### **File BukuModel.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Models;` yang mendefinisikan namespace untuk kelas ini sebagai `App\Models`, yang berarti kelas ini berada dalam folder `App/Models`.

Pada baris [5] terdapat `use CodeIgniter\Model;` yang mengimpor kelas `Model` dari framework `CodeIgniter`.

Pada baris [7] terdapat deklarasi kelas `BukuModel` yang memperluas kelas `Model`. Dengan memperluas kelas `Model`, kelas `BukuModel` dapat menggunakan semua fitur dan metode yang disediakan oleh `CodeIgniter` untuk berinteraksi dengan database.



Pada baris [9] terdapat `protected $table = 'buku';` yang mendefinisikan nama tabel yang akan digunakan oleh model ini, yaitu tabel buku.

Pada baris [10] terdapat `protected $allowedFields = ['judul', 'penulis', 'penerbit', 'tahun_terbit'];` yang mendefinisikan daftar field yang diizinkan untuk diisi atau diubah melalui model ini. Field yang disebutkan adalah judul, penulis, penerbit, dan tahun\_terbit.

Pada baris [11-13] terdapat `protected $createdField`, `protected $updatedField`, dan `protected $deletedField` yang masing-masing mendefinisikan field yang akan digunakan untuk menyimpan timestamp saat record dibuat, diupdate, dan dihapus. Field tersebut adalah `created_at`, `updated_at`, dan `deleted_at`.

Pada baris [15-41] terdapat `protected $validationRules` yang mendefinisikan aturan validasi untuk setiap field.

Pada baris [17-21] terdapat aturan validasi untuk field `judul`, yang harus diisi (`required`) dan harus berupa string (`string`). Pesan error juga disediakan untuk setiap aturan.

Pada baris [23-27] terdapat aturan validasi untuk field `penulis`, yang harus diisi (`required`) dan harus berupa string (`string`). Pesan error juga disediakan untuk setiap aturan.

Pada baris [29-33] terdapat aturan validasi untuk field `penerbit`, yang harus diisi (`required`) dan harus berupa string (`string`). Pesan error juga disediakan untuk setiap aturan.

Pada baris [35-41] terdapat aturan validasi untuk field `tahun_terbit`, yang harus diisi (`required`), harus berupa integer (`integer`), harus lebih besar dari 1800 (`greater_than[1800]`), dan harus lebih kecil dari 2024 (`less_than[2024]`). Pesan error juga disediakan untuk setiap aturan.

Pada baris [43] terdapat `protected $skipValidation = false;` yang berarti validasi tidak akan dilewati. Jika nilai ini diatur ke `true`, maka validasi akan dilewati.

### **File UserModel.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Models;` yang mendefinisikan namespace untuk kelas ini sebagai `App\Models`, yang berarti kelas ini berada dalam folder `App/Models`.

Pada baris [5] terdapat `use CodeIgniter\Model;` yang mengimpor kelas Model dari framework CodeIgniter.

Pada baris [7] terdapat deklarasi kelas `UsersModel` yang memperluas kelas Model. Dengan memperluas kelas Model, kelas `UsersModel` dapat menggunakan semua fitur dan metode yang disediakan oleh CodeIgniter untuk berinteraksi dengan database.

Pada baris [9] terdapat `protected $table = "users";` yang mendefinisikan nama tabel yang akan digunakan oleh model ini, yaitu tabel `users`.

Pada baris [10] terdapat `protected $returnType = "object";` yang menentukan bahwa hasil query yang dilakukan oleh model ini akan dikembalikan sebagai objek.

Pada baris [11] terdapat `protected $useTimestamps = true;` yang menentukan bahwa model ini akan secara otomatis mengelola field `created_at`, `updated_at`, dan `deleted_at` (jika field tersebut ada di tabel).

Pada baris [12] terdapat `protected $allowedFields = ['username', 'password', 'email'];` yang mendefinisikan daftar field yang diizinkan untuk diisi atau diubah melalui model ini. Field yang disebutkan adalah `username`, `password`, dan `email`.

### **File 2024-06-03-103310\_Users.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Database\Migrations;` yang mendefinisikan namespace untuk kelas ini sebagai `App\Database\Migrations`, yang berarti kelas ini berada dalam folder `App/Database/Migrations`.

Pada baris [5] terdapat `use CodeIgniter\Database\Migration;` yang mengimpor kelas Migration dari framework CodeIgniter.

Pada baris [7] terdapat deklarasi kelas `Users` yang memperluas kelas Migration. Dengan memperluas kelas Migration, kelas `Users` dapat menggunakan semua fitur dan metode yang disediakan oleh CodeIgniter untuk mengelola migrasi database.

Pada baris [9-34] terdapat metode `up()` yang digunakan untuk mendefinisikan skema tabel `users` dan membuat tabel tersebut di database.

Pada baris [10-32] terdapat `$this->forge->addField([...])` yang mendefinisikan field-field yang akan ada di tabel `users`.

Pada baris [11-14] terdapat definisi untuk field `id` yang merupakan integer, unsigned, dan auto increment.

Pada baris [15-18] terdapat definisi untuk field `username` yang merupakan VARCHAR dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [19-22] terdapat definisi untuk field `email` yang merupakan VARCHAR dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [23-26] terdapat definisi untuk field `password` yang merupakan TEXT dan tidak boleh null.

Pada baris [27-29] terdapat definisi untuk field `created_at` yang merupakan datetime dan boleh null.

Pada baris [30-32] terdapat definisi untuk field `updated_at` yang merupakan datetime dan boleh null.

Pada baris [34] terdapat `$this->forge->addKey('id', TRUE);` yang menetapkan field `id` sebagai primary key.

Pada baris [35] terdapat `$this->forge->createTable('users');` yang membuat tabel `users` di database dengan skema yang telah didefinisikan.

Pada baris [39-42] terdapat metode `down()` yang digunakan untuk menghapus tabel `users` dari database.

Pada baris [41] terdapat `$this->forge->dropTable('users');` yang menghapus tabel `users`.

### **File 2024-06-03-122804\_Buku.php**

Pada baris [1], `<?php` adalah tag pembuka untuk kode PHP.

Pada baris [3], `namespace App\Database\Migrations;` mendefinisikan namespace untuk kelas ini sebagai `App\Database\Migrations`, yang berarti kelas ini berada dalam folder `App/Database/Migrations`.

Pada baris [5], `use CodeIgniter\Database\Migration;` mengimpor kelas `Migration` dari framework `CodeIgniter`.

Pada baris [7], deklarasi kelas `Buku` yang memperluas kelas `Migration`. Dengan memperluas kelas `Migration`, kelas `Buku` dapat menggunakan semua fitur dan metode yang disediakan oleh `CodeIgniter` untuk mengelola migrasi database.

Pada baris [9], deklarasi metode `up()` yang digunakan untuk mendefinisikan skema tabel buku dan membuat tabel tersebut di database.

Pada baris [10], `$this->forge->addField()` mulai mendefinisikan field-field yang akan ada di tabel buku.

Pada baris [11-14], definisi untuk field `id` yang merupakan `BIGINT`, `unsigned`, dan `auto increment`. `BIGINT` digunakan untuk menyimpan nilai integer yang besar, `unsigned` berarti nilai tidak negatif, dan `auto_increment` akan otomatis menambahkan nilai unik untuk setiap record baru.

Pada baris [15-18], definisi untuk field `judul` yang merupakan `VARCHAR` dengan panjang maksimal 255 karakter dan tidak boleh `null` (`null => FALSE`).

Pada baris [19-22], definisi untuk field `penulis` yang merupakan `VARCHAR` dengan panjang maksimal 255 karakter dan tidak boleh `null` (`null => FALSE`).

Pada baris [23-26], definisi untuk field `penerbit` yang merupakan `VARCHAR` dengan panjang maksimal 255 karakter dan tidak boleh `null` (`null => FALSE`).

Pada baris [27-29], definisi untuk field `tahun_terbit` yang merupakan `YEAR` dan boleh `null` (`null => TRUE`).

Pada baris [30], penutupan array yang mendefinisikan field-field tabel buku.

Pada baris [32], `$this->forge->addKey('id', TRUE)`; menetapkan field `id` sebagai `primary key`. `TRUE` menunjukkan bahwa ini adalah `primary key`.

Pada baris [33], `$this->forge->createTable('buku')`; membuat tabel buku di database dengan skema yang telah didefinisikan.

Pada baris [36], deklarasi metode `down()` yang digunakan untuk menghapus tabel buku dari database.

Pada baris [38], `$this->forge->dropTable('buku')`; menghapus tabel buku.

Pada baris [40], penutupan metode `down()`.

Pada baris [42], penutupan kelas `Buku`.

### **File Seeds Buku.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Database\Seeds;` yang digunakan untuk mendefinisikan namespace dari kelas ini sebagai `App\Database\Seeds`.

Pada baris [5] terdapat `use CodeIgniter\Database\Seeder;` yang digunakan untuk mengimpor kelas Seeder dari framework CodeIgniter.

Pada baris [7] terdapat `class Buku extends Seeder` yang mendeklarasikan kelas Buku yang memperluas kelas Seeder.

Pada baris [9] terdapat `public function run()` yang mendeklarasikan metode run yang akan dijalankan ketika seeder ini dieksekusi.

Pada baris [11] terdapat `$this->db->table('buku')->insert([` yang digunakan untuk memasukkan data baru ke dalam tabel buku.

Pada baris [12] terdapat `'judul' => 'Laut Bercerita',` yang mengatur nilai field judul untuk record pertama.

Pada baris [13] terdapat `'penulis' => 'Leila S. Chudori',` yang mengatur nilai field penulis untuk record pertama.

Pada baris [14] terdapat `'penerbit' => 'Gramedia',` yang mengatur nilai field penerbit untuk record pertama.

Pada baris [15] terdapat `'tahun_terbit' => '2017'` yang mengatur nilai field tahun\_terbit untuk record pertama.

Pada baris [16] terdapat `]);` yang menutup array data dan perintah insert untuk record pertama.

Pada baris [18] terdapat `$this->db->table('buku')->insert([` yang digunakan untuk memasukkan data baru ke dalam tabel buku.

Pada baris [19] terdapat `'judul' => 'Tentang Kamu',` yang mengatur nilai field judul untuk record kedua.

Pada baris [20] terdapat `'penulis' => 'Tere Liye',` yang mengatur nilai field penulis untuk record kedua.

Pada baris [21] terdapat `'penerbit' => 'Republika',` yang mengatur nilai field penerbit untuk record kedua.

Pada baris [22] terdapat `'tahun_terbit' => '2016'` yang mengatur nilai field tahun\_terbit untuk record kedua.

Pada baris [23] terdapat `]);` yang menutup array data dan perintah insert untuk record kedua.

Pada baris [25] terdapat `}` yang menutup metode run.

Pada baris [26] terdapat `}` yang menutup kelas Buku.

### **File UserSeeder.php**

Pada baris [1] terdapat `<?php` yang merupakan tag pembuka untuk kode PHP.

Pada baris [3] terdapat `namespace App\Database\Seeds;` yang digunakan untuk mendefinisikan namespace dari kelas ini sebagai `App\Database\Seeds`.

Pada baris [5] terdapat `use CodeIgniter\Database\Seeder;` yang digunakan untuk mengimpor kelas Seeder dari framework CodeIgniter.

Pada baris [7] terdapat `class UserSeeder extends Seeder` yang mendeklarasikan kelas UserSeeder yang memperluas kelas Seeder.

Pada baris [9] terdapat `public function run()` yang mendeklarasikan metode run yang akan dijalankan ketika seeder ini dieksekusi.

Pada baris [11] terdapat `$this->db->table('users')->insert([` yang digunakan untuk memasukkan data baru ke dalam tabel users.

Pada baris [12] terdapat `'username' => noraina,` yang mengatur nilai field username untuk record yang dimasukkan.

Pada baris [13] terdapat `'email' => 'norainaevayantii@gmail.com',` yang mengatur nilai field email untuk record yang dimasukkan.

Pada baris [14] terdapat `'password' => password_hash('123', PASSWORD_DEFAULT),` yang mengatur nilai field password untuk record yang dimasukkan. Fungsi `password_hash` digunakan untuk mengenkripsi kata sandi 123 dengan algoritma hashing default.

Pada baris [15] terdapat `]);` yang menutup array data dan perintah insert untuk record tersebut.

Pada baris [17] terdapat `}` yang menutup metode run.

Pada baris [18] terdapat `}` yang menutup kelas UserSeeder.

### **File perpustakaan.sql**

Pada baris [1] terdapat `-- phpMyAdmin SQL Dump` yang menunjukkan ini adalah dump SQL dari phpMyAdmin.

Pada baris [2] terdapat `-- version 5.2.1` yang menunjukkan versi phpMyAdmin yang digunakan.

Pada baris [3] terdapat `-- https://www.phpmyadmin.net/` yang menunjukkan situs web phpMyAdmin.

Pada baris [5] terdapat `-- Host: 127.0.0.1` yang menunjukkan host database.

Pada baris [6] terdapat `-- Waktu pembuatan: 11 Jun 2024 pada 19.57` yang menunjukkan waktu pembuatan dump.

Pada baris [7] terdapat `-- Versi server: 10.4.32-MariaDB` yang menunjukkan versi server database yang digunakan.

Pada baris [8] terdapat `-- Versi PHP: 8.2.12` yang menunjukkan versi PHP yang digunakan.

Pada baris [10] terdapat `SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";` yang digunakan untuk mengatur mode SQL.

Pada baris [11] terdapat `START TRANSACTION;` yang memulai transaksi SQL.

Pada baris [12] terdapat `SET time_zone = "+00:00";` yang mengatur zona waktu menjadi UTC.

Pada baris [15] terdapat `/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;` yang menyimpan pengaturan karakter set client lama.

Pada baris [16] terdapat `/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;` yang menyimpan pengaturan karakter set hasil lama.

Pada baris [17] terdapat `/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;` yang menyimpan pengaturan collation connection lama.

Pada baris [18] terdapat `/*!40101 SET NAMES utf8mb4 */;` yang mengatur encoding karakter menjadi UTF-8 multibyte.

Pada baris [22] terdapat `-- Database: 'perpustakaan'` yang menunjukkan database yang digunakan adalah perpustakaan.

Pada baris [28] terdapat `-- Struktur dari tabel 'buku'` yang menunjukkan akan mendefinisikan struktur tabel buku.

Pada baris [30] terdapat `CREATE TABLE 'buku'` yang mendefinisikan tabel buku.

Pada baris [31] terdapat `id bigint(20) UNSIGNED NOT NULL` yang mendefinisikan field id sebagai BIGINT, unsigned, dan tidak boleh null.

Pada baris [32] terdapat `judul varchar(255) NOT NULL` yang mendefinisikan field judul sebagai VARCHAR dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [33] terdapat `penulis varchar(255) NOT NULL` yang mendefinisikan field penulis sebagai VARCHAR dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [34] terdapat `penerbit varchar(255) NOT NULL` yang mendefinisikan field penerbit sebagai VARCHAR dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [35] terdapat `tahun_terbit year(4) DEFAULT NULL` yang mendefinisikan field tahun\_terbit sebagai YEAR yang boleh null.

Pada baris [37] terdapat `ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci` yang mengatur mesin tabel menggunakan InnoDB dan charset serta collation menggunakan utf8mb4.

Pada baris [41] terdapat `-- Dumping data untuk tabel 'buku'` yang menunjukkan akan memasukkan data ke dalam tabel buku.

Pada baris [43] terdapat `INSERT INTO 'buku' ('id', 'judul', 'penulis', 'penerbit', 'tahun_terbit') VALUES` yang digunakan untuk memasukkan data ke dalam tabel buku.

Pada baris [44] terdapat `(15, 'Laut Bercerita', 'Leila S. Chudori', 'Gramedia', '2017')` yang memasukkan data buku dengan id 15.

Pada baris [45] terdapat `(16, 'Tentang Kamu', 'Tere Liye', 'Republika', '2016')` yang memasukkan data buku dengan id 16.

Pada baris [50] terdapat `-- Struktur dari tabel 'migrations'` yang menunjukkan akan mendefinisikan struktur tabel migrations.

Pada baris [52] terdapat `CREATE TABLE 'migrations'` yang mendefinisikan tabel migrations.

Pada baris [53] terdapat `id bigint(20) UNSIGNED NOT NULL` yang mendefinisikan field id sebagai BIGINT, unsigned, dan tidak boleh null.

Pada baris [54] terdapat `version varchar(255) NOT NULL` yang mendefinisikan field version sebagai VARCHAR dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [55] terdapat `class varchar(255) NOT NULL` yang mendefinisikan field class sebagai VARCHAR dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [56] terdapat `group varchar(255) NOT NULL` yang mendefinisikan field group sebagai VARCHAR dengan panjang maksimal 255 karakter dan tidak boleh null.



Pada baris [57] terdapat `namespace varchar(255) NOT NULL` yang mendefinisikan field `namespace` sebagai `VARCHAR` dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [58] terdapat `time int(11) NOT NULL` yang mendefinisikan field `time` sebagai `INT` yang tidak boleh null.

Pada baris [59] terdapat `batch int(11) UNSIGNED NOT NULL` yang mendefinisikan field `batch` sebagai `INT unsigned` yang tidak boleh null.

Pada baris [61] terdapat `ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci` yang mengatur mesin tabel menggunakan `InnoDB` dan charset serta collation menggunakan `utf8mb4`.

Pada baris [65] terdapat `-- Dumping data untuk tabel 'migrations'` yang menunjukkan akan memasukkan data ke dalam tabel `migrations`.

Pada baris [67] terdapat `INSERT INTO 'migrations' ('id', 'version', 'class', 'group', 'namespace', 'time', 'batch') VALUES` yang digunakan untuk memasukkan data ke dalam tabel `migrations`.

Pada baris [68] terdapat `(1, '2024-06-03-103310', 'App\\Database\\Migrations\\Users', 'default', 'App', 1718081782, 1)` yang memasukkan data migrasi dengan id 1.

Pada baris [69] terdapat `(2, '2024-06-03-122804', 'App\\Database\\Migrations\\Buku', 'default', 'App', 1718081782, 1)` yang memasukkan data migrasi dengan id 2.

Pada baris [70] terdapat `(4, '2024-06-11-173406', 'App\\Database\\Migrations\\Users', 'default', 'App', 1718127395, 3)` yang memasukkan data migrasi dengan id 4.

Pada baris [75] terdapat `-- Struktur dari tabel 'users'` yang menunjukkan akan mendefinisikan struktur tabel `users`.

Pada baris [77] terdapat `CREATE TABLE 'users'` yang mendefinisikan tabel `users`.

Pada baris [78] terdapat `id int(10) UNSIGNED NOT NULL` yang mendefinisikan field `id` sebagai `INT`, unsigned, dan tidak boleh null.

Pada baris [79] terdapat `username varchar(255) NOT NULL` yang mendefinisikan field `username` sebagai `VARCHAR` dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [80] terdapat `email varchar(255) NOT NULL` yang mendefinisikan field `email` sebagai `VARCHAR` dengan panjang maksimal 255 karakter dan tidak boleh null.

Pada baris [81] terdapat `password text NOT NULL` yang mendefinisikan field password sebagai TEXT yang tidak boleh null.

Pada baris [82] terdapat `created_at datetime DEFAULT NULL` yang mendefinisikan field created\_at sebagai datetime yang boleh null.

Pada baris [83] terdapat `updated_at datetime DEFAULT NULL` yang mendefinisikan field updated\_at sebagai datetime yang boleh null.

Pada baris [85] terdapat `ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci` yang mengatur mesin tabel menggunakan InnoDB dan charset serta collation menggunakan utf8mb4.

Pada baris [89] terdapat `-- Dumping data untuk tabel 'users'` yang menunjukkan akan memasukkan data ke dalam tabel users.

Pada baris [91] terdapat `INSERT INTO 'users' ('id', 'username', 'email', 'password', 'created_at', 'updated_at') VALUES` yang digunakan untuk memasukkan data ke dalam tabel users.

Pada baris [92] terdapat `(4, 'noraina', 'norainaevayantii@gmail.com', '$2y$10$xYhsftGqi132/1Y8Qrypge2VGyR2bS0TEBFpXTa4sXUKI4GwnEoh2', NULL, NULL)` yang memasukkan data pengguna dengan id 4.

Pada baris [97] terdapat `-- Indeks untuk tabel 'buku'` yang menunjukkan akan menambahkan indeks untuk tabel buku.

Pada baris [99] terdapat `ALTER TABLE 'buku' ADD PRIMARY KEY ('id');` yang menambahkan primary key pada field id di tabel.

#### **D. Tautan Git**

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/norainaevayanti/Praktikum-Pemrograman-Web-II.git>