

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD dan Login pada Codeigniter

Oleh:

Muhammad Hafiz Ansari

NIM. 2210817210003

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN II
MODUL 6

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

Nama Praktikan : Muhammad Hafiz Ansari
NIM : 2210817210003

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

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

DAFTAR ISI

| | |
|---------------------------|----|
| LEMBAR PENGESAHAN | 2 |
| DAFTAR ISI | 3 |
| DAFTAR GAMBAR..... | 4 |
| DAFTAR TABEL | 5 |
| SOAL 1 | 6 |
| A. Source Code | 6 |
| B. Output Program..... | 30 |
| C. Pembahasan..... | 32 |
| D. Tautan Git | 40 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1. Soal Praktikum 6..... | 6 |
| Gambar 2. Screenshot 1 Hasil Jawaban Praktikum 6..... | 30 |
| Gambar 3. Screenshot 2 Hasil Jawaban Praktikum 6..... | 31 |
| Gambar 4. Screenshot 3 Hasil Jawaban Praktikum 6..... | 31 |
| Gambar 5. Screenshot 4 Hasil Jawaban Praktikum 6..... | 32 |

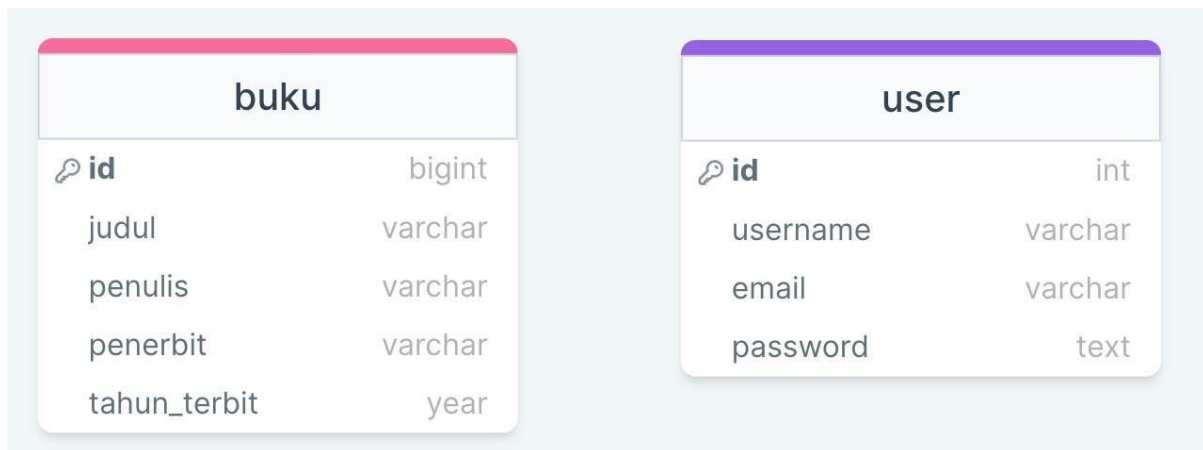
DAFTAR TABEL

| | |
|---|----|
| Tabel 1. Source Code Routes.php | 7 |
| Tabel 2. Source Code BukuController.php | 9 |
| Tabel 3. Source Code UserController.php..... | 11 |
| Tabel 4. Source Code AuthFilter.php | 12 |
| Tabel 5. Source Code Create.php | 15 |
| Tabel 6. Source Code Edit.php | 19 |
| Tabel 7. Source Code Index.php | 22 |
| Tabel 8. Source Code Login.php | 26 |
| Tabel 9. Source Code BukuModel.php | 27 |
| Tabel 10. Source Code UserModel.php..... | 28 |
| Tabel 11. Source Code UserSeeder.php | 28 |
| Tabel 12. Source Code Buku.php | 29 |
| Tabel 13. Source Code User.php | 30 |

SOAL 1

Buatlah sebuah website yang memiliki fitur login dan logout. Setelah login pengguna website dapat melakukan CRUD yang terhubung langsung dengan database. Beberapa syarat yang harus ada pada website:

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



Gambar 1. Soal Praktikum 6

Simpan dalam folder github PRAK 601

A. Source Code

Routes.php

```
1 <?php
2
3 namespace Config;
4
5 $routes = Services::routes();
6
7 $routes->setDefaultNamespace('App\Controllers');
8 $routes->setDefaultController('Home');
9 $routes->setDefaultMethod('index');
10 $routes->setTranslateURIDashes(false);
11 $routes->set404Override();
12
```

```

13 $routes->group('', ["filter" => 'auth'], static function ($routes)
14 {
15     $routes->get('/', 'BukuController::index');
16     $routes->get('/buku', 'BukuController::index');
17     $routes->get('/buku/create', 'BukuController::create');
18     $routes->post('/buku/create', 'BukuController::store');
19     $routes->get('/buku/(:num)/edit', 'BukuController::edit/$1');
20     $routes->post('/buku/(:num)/edit', 'BukuController::update/$1');
21     $routes->delete('/buku/(:num)/delete', 'BukuController::delete/$1');
22     $routes->get('/logout', 'UserController::logout');
23 });
24 $routes->get('/login', 'UserController::index');
25 $routes->post('/login', 'UserController::login');
26
27 $routes->get('/signup', 'UserController::signUp', ['as'=>'signup']);
28 $routes->post('/signup', 'UserController::signUpProcess');
29
30 $routes->get('/forgot-password', 'UserController::forgotPassword');
31 $routes->post('/forgot-password',
32     'UserController::forgotPasswordProcess');
33
34 if (is_file(APPPATH . 'Config/' . ENVIRONMENT . '/Routes.php')) {
35     require APPPATH . 'Config/' . ENVIRONMENT . '/Routes.php';
36 }

```

Tabel 1. Source Code Routes.php

BukuController.php

```

1 <?php
2
3 namespace App\Controllers;
4
5 use App\Controllers\BaseController;
6 use App\Models\BukuModel;
7
8 class BukuController extends BaseController
9 {
10     public function index()
11     {
12         return view('buku/index', [
13             'buku' => (new BukuModel())->findAll()
14         ]);
15     }
16
17     public function create()
18     {
19         return view('buku/create');
20     }
21
22     public function store()

```

```

23     {
24         $validation = \Config\Services::validation();
25         $validation->setRules([
26             'judul' => 'required|string',
27             'penulis' => 'required|string',
28             'penerbit' => 'required|string',
29             'tahun_terbit' =>
'required|integer|greater_than[1800]|less_than[2024]',
30         ]);
31
32         if ($validation->withRequest($this->request)->run() ===
FALSE) {
33             return redirect()->back()->withInput()-
>with('errors', $validation->getErrors());
34         }
35
36         $buku = new BukuModel();
37         $buku->insert([
38             'judul' => $this->request->getPost('judul'),
39             'penulis' => $this->request->getPost('penulis'),
40             'penerbit' => $this->request->getPost('penerbit'),
41             'tahun_terbit' => $this->request-
>getPost('tahun_terbit'),
42         ]);
43
44         return redirect()->to('/buku');
45     }
46
47     public function edit($id)
48     {
49         return view('buku/edit', [
50             'buku' => (new BukuModel())->find($id)
51         ]);
52     }
53
54     public function update($id)
55     {
56         $validation = \Config\Services::validation();
57         $validation->setRules([
58             'judul' => 'required|string',
59             'penulis' => 'required|string',
60             'penerbit' => 'required|string',
61             'tahun_terbit' =>
'required|integer|greater_than[1800]|less_than[2024]',
62         ]);
63
64         if ($validation->withRequest($this->request)->run() ===
FALSE) {
65             return redirect()->back()->withInput()->with('errors',
$validation->getErrors());
66         }
67     }

```


| | |
|----|---|
| 68 | \$buku = new BukuModel(); |
| 69 | \$buku->update(\$id, [|
| 70 | 'judul' => \$this->request->getPost('judul'), |
| 71 | 'penulis' => \$this->request->getPost('penulis'), |
| 72 | 'penerbit' => \$this->request->getPost('penerbit'), |
| 73 | 'tahun_terbit' => \$this->request-> |
| | >getPost('tahun_terbit'), |
| 74 |]); |
| 75 | |
| 76 | return redirect()->to('/buku'); |
| 77 | } |
| 78 | |
| 79 | public function delete(\$id) |
| 80 | { |
| 81 | \$buku = new BukuModel(); |
| 82 | \$buku->delete(\$id); |
| 83 | return redirect()->to('/buku'); |
| 84 | } |
| 85 | } |

Tabel 2. Source Code BukuController.php

UserController.php

| | |
|----|--|
| 1 | <?php |
| 2 | |
| 3 | namespace App\Controllers; |
| 4 | |
| 5 | use App\Controllers\BaseController; |
| 6 | use App\Models\UserModel; |
| 7 | |
| 8 | class UserController extends BaseController |
| 9 | { |
| 10 | public function index() |
| 11 | { |
| 12 | return view('login'); |
| 13 | } |
| 14 | |
| 15 | public function login() |
| 16 | { |
| 17 | \$user = new UserModel(); |
| 18 | \$username = \$this->request->getPost('username'); |
| 19 | \$password = \$this->request->getPost('password'); |
| 20 | |
| 21 | if (empty(\$username) empty(\$password)) { |
| 22 | session()->setFlashdata('pesan', 'Username dan |
| | password harus diisi'); |
| 23 | return redirect()->to(base_url('login')); |
| 24 | } |
| 25 | |
| 26 | \$dataUser = \$user->where('username', \$username)-> |
| | orWhere('email', \$username)->first(); |

```

27         if (!$dataUser) {
28             session()->setFlashdata('pesan', 'Username atau email
tidak ditemukan');
                return redirect()->to(base_url('login'));
29         }
30
31         if (!password_verify($pw, $dataUser['password'])) {
32             session()->setFlashdata('pesan', 'Password salah');
33             return redirect()->to(base_url('login'));
34         }
35
36         session()->set([
37             'username' => $dataUser['username'],
38             'email' => $dataUser['email'],
39             'logged_in' => true
40         ]);
41         return redirect()->to(base_url('buku'));
42     }
43
44     public function logout()
45     {
46         session()->destroy();
47         return redirect()->to(base_url('login'));
48     }
49
50     public function signUp()
51     {
52         if (session()->get('logged_in')) {
53             return redirect()->to(base_url('buku'));
54         }
55         return view('signup');
56     }
57
58     public function signUpProcess()
59     {
60         $rules = [
61             'username' =>
62             'required|min_length[5]|max_length[20]|is_unique[user.username]',
63             'email' =>
64             'required|valid_email|is_unique[user.email]',
65             'password' => 'required|min_length[8]',
66             'confirm_password' => 'matches[password]'
67         ];
68
69         if (!$this->validate($rules)) {
70             return redirect()->to('/signup')->withInput()-
71             >with('errors', $this->validator->getErrors());
72         }
73
74         $user = new UserModel();
75         $newData = [
76             'username' => $this->request->getPost('username'),

```

| | |
|-----|---|
| 74 | 'email' => \$this->request->getPost('email'), |
| 75 | 'password' => password_hash(\$this->request- |
| 76 | >getPost('password'), PASSWORD_DEFAULT) |
| |]; |
| 77 | |
| 78 | \$user->save(\$newData); |
| 79 | |
| 80 | session()->set([|
| 81 | 'username' => \$newData['username'], |
| 82 | 'email' => \$newData['email'], |
| 83 | 'logged_in' => true |
| 84 |]); |
| 85 | |
| 86 | session()->setFlashdata('success', 'Pendaftaran berhasil. |
| 87 | Anda sekarang masuk.'); |
| | return redirect()->to('/buku'); |
| 88 | } |
| 89 | |
| 90 | public function forgotPassword() |
| 91 | { |
| 92 | return view('forgot_password'); |
| 93 | } |
| 94 | |
| 95 | public function forgotPasswordProcess() |
| 96 | { |
| 97 | \$email = \$this->request->getPost('email'); |
| 98 | \$userModel = new UserModel(); |
| 99 | \$user = \$userModel->where('email', \$email)->first(); |
| 100 | |
| 101 | if (!\$user) { |
| 102 | session()->setFlashdata('pesan', 'Email tidak |
| 103 | ditemukan.'); |
| | return redirect()->to(base_url('/forgot-password')); |
| 104 | } |
| 105 | |
| 106 | session()->setFlashdata('pesan', 'Email untuk reset |
| 107 | password telah dikirim.'); |
| | return redirect()->to(base_url('/forgot-password')); |
| 108 | } |
| 109 | } |
| 110 | |

Tabel 3. Source Code UserController.php

AuthFilter.php

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

| | |
|----|---|
| 7 | use CodeIgniter\HTTP\ResponseInterface; |
| 8 | |
| 9 | class AuthFilter implements FilterInterface |
| 10 | { |
| 11 | public function before(RequestInterface \$request, \$arguments = null) |
| 12 | { |
| 13 | if(!session()->get('logged_in')){ |
| 14 | session()->setFlashdata('pesan', 'Login terlebih dahulu!'); |
| 15 | return redirect()->to(base_url('login')); |
| 16 | } |
| 17 | } |
| 18 | |
| 19 | public function after(RequestInterface \$request, ResponseInterface \$response, \$arguments = null) |
| 20 | { |
| 21 | |
| 22 | } |
| 23 | } |

Tabel 4. Source Code AuthFilter.php

Create.php

| | |
|----|--|
| 1 | <!DOCTYPE html> |
| 2 | <html lang="en"> |
| 3 | |
| 4 | <head> |
| 5 | <meta charset="UTF-8"> |
| 6 | <meta http-equiv="X-UA-Compatible" content="IE=edge"> |
| 7 | <meta name="viewport" content="width=device-width, initial-scale=1.0"> |
| 8 | <title>Tambah Data Buku</title> |
| 9 | <style> |
| 10 | body { |
| 11 | background-color: #393E46; |
| 12 | font-family: Arial, sans-serif; |
| 13 | margin: 0; |
| 14 | padding: 0; |
| 15 | } |
| 16 | |
| 17 | .container { |
| 18 | max-width: 600px; |
| 19 | margin: 100px auto; |
| 20 | padding: 20px; |
| 21 | background-color: #EEEEEE; |
| 22 | border-radius: 10px; |
| 23 | box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); |
| 24 | } |
| 25 | |
| 26 | .header { |

```
27         background-color: #222831;
28         color: #fff;
29         border-radius: 10px 10px 0 0;
30         padding: 10px 20px;
31         text-align: center;
32     }
33
34     .form-title {
35         margin: 0;
36         font-size: 24px;
37     }
38
39     .form-body {
40         padding: 20px;
41         background-color: #fff;
42         border-radius: 0 0 10px 10px;
43     }
44
45     .form-label {
46         font-weight: 600;
47         display: block;
48         margin-bottom: 5px;
49         color: #333;
50     }
51
52     .form-control {
53         width: 100%;
54         padding: 10px;
55         border-radius: 5px;
56         border: 1px solid #ddd;
57         box-sizing: border-box;
58         margin-bottom: 20px;
59     }
60
61     .btn {
62         display: inline-block;
63         padding: 10px 20px;
64         font-size: 16px;
65         border-radius: 5px;
66         text-align: center;
67         text-decoration: none;
68         color: #fff;
69         background-color: #00ADB5;
70         border: none;
71         cursor: pointer;
72         transition: background-color 0.3s ease;
73         margin: 0 auto;
74         display: block;
75     }
76
77     .btn:hover {
78         background-color: #138496;
```

```

79         }
80
81         .error-message {
82             color: red;
83             font-size: 14px;
84             margin-top: -15px;
85             margin-bottom: 15px;
86             display: none;
87         }
88
89         .button-container {
90             text-align: center;
91         }
92     </style>
93 </head>
94
95 <body>
96     <div class="container">
97         <div class="header">
98             <h2 class="form-title">Tambah Data Buku</h2>
99         </div>
100        <div class="form-body">
101            <form
102                id="bukuForm"
103                action="=
104                base_url('/buku/create')
105                ?"
106                method="post"
107                onsubmit="return
108                validateForm()"
109            >
110                <label
111                    for="judul"
112                    class="form-
113                    label">Judul</label>
114                <input
115                    type="text"
116                    class="form-control"
117                    id="judul"
118                    name="judul">
119                <div class="error-message" id="judulError">Judul
120                harus diisi dan berupa string.</div>
121
122                <label
123                    for="penulis"
124                    class="form-
125                    label">Penulis</label>
126                <input
127                    type="text"
128                    class="form-control"
129                    id="penulis"
130                    name="penulis">
131                <div
132                    class="error-message"
133                    id="penulisError">Penulis harus diisi dan berupa string.</div>
134
135                <label
136                    for="penerbit"
137                    class="form-
138                    label">Penerbit</label>
139                <input
140                    type="text"
141                    class="form-control"
142                    id="penerbit"
143                    name="penerbit">
144                <div
145                    class="error-message"
146                    id="penerbitError">Penerbit harus diisi dan berupa string.</div>
147
148                <label
149                    for="tahun_terbit"
150                    class="form-
151                    label">Tahun Terbit</label>
152                <input
153                    type="text"
154                    class="form-control"
155                    id="tahun_terbit"
156                    name="tahun_terbit">
157                <div class="error-message" id="tahunError">Tahun
158                Terbit harus diisi dan berupa angka antara 1800 dan 2024.</div>

```

```

117
118         <div class="button-container">
119             <button                                type="submit"
class="btn">Tambah</button>
120         </div>
121     </form>
122 </div>
123 </div>
124
125     <script>
126         function validateForm() {
127             document.getElementById('judulError').style.display =
128 'none';
129 document.getElementById('penulisError').style.display = 'none';
130 document.getElementById('penerbitError').style.display= 'none';
131 document.getElementById('tahunError').style.display = 'none';
132
133 let isValid = true;
134
135         const judul = document.getElementById('judul').value;
136         if (judul === '' || !isNaN(judul)) {
137             document.getElementById('judulError').style.display = 'block';
138             isValid = false;
139         }
140
141         const penulis = document.getElementById('penulis').value;
142         if (penulis === '' || !isNaN(penulis)) {
143             document.getElementById('penulisError').style.display='block';
144             isValid = false;
145         }
146
147         Const penerbit= document.getElementById('penerbit').value;
148         if (penerbit === '' || !isNaN(penerbit)) {
149             document.getElementById('penerbitError').style.display='block';
150             isValid = false;
151         }
152
153         ConsttahunTerbit=document.getElementById('tahun_terbit').value;
154         const tahunTerbitNumber = parseInt(tahunTerbit);
155         if (isNaN(tahunTerbitNumber) || tahunTerbitNumber <
1800 || tahunTerbitNumber > 2024) {
156             document.getElementById('tahunError').style.display = 'block';
157             isValid = false;
158         }
159
160         return isValid;
161     }
162 </script>
163 </body>
164
165 </html>

```

Tabel 5. Source Code Create.php

Edit.php

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-
scale=1.0">
7     <title>Edit Data Buku</title>
8     <style>
9         body {
10             background-color: #393E46;
11             font-family: Arial, sans-serif;
12             margin: 0;
13             padding: 0;
14         }
15         .container {
16             max-width: 600px;
17             margin: 100px auto;
18             padding: 20px;
19             background-color: #EEEEEE;
20             border-radius: 10px;
21             box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
22         }
23         .header {
24             background-color: #222831;
25             color: #ffffff;
26             padding: 10px 20px;
27             border-radius: 10px 10px 0 0;
28             text-align: center;
29         }
30         .form-body {
31             padding: 20px;
32             background-color: #fff;
33             border-radius: 0 0 10px 10px;
34         }
35         .form-label {
36             font-weight: 600;
37             margin-bottom: 8px;
38             display: block;
39             color: #333;
40         }
41         .form-control {
42             width: 100%;
43             padding: 10px;
44             border-radius: 5px;
45             border: 1px solid #ccc;
46             box-sizing: border-box;
47             margin-bottom: 20px;
48         }
49         .btn {
```



```

50         display: inline-block;
51         padding: 10px 20px;
52         font-size: 16px;
53         border-radius: 5px;
54         text-align: center;
55         text-decoration: none;
56         color: #fff;
57         background-color: #00ADB5;
58         border: none;
59         cursor: pointer;
60         transition: background-color 0.3s ease;
61         margin: 0 auto;
62         display: block;
63     }
64     .btn:hover {
65         background-color: #138496;
66     }
67     .error {
68         color: #ff0000;
69         font-size: 12px;
70         margin-bottom: 10px;
71     }
72     .button-container {
73         text-align: center;
74     }
75 </style>
76 </head>
77 <body>
78     <div class="container">
79         <div class="header">
80             <h2>Edit Data Buku</h2>
81         </div>
82         <div class="form-body">
83             <form action="<?= base_url('/buku/' . $buku['id'] .
'/edit') ?>" method="post" onsubmit="return validateForm()">
84                 <div class="mb-3">
85                     <label for="judul" class="form-label">Judul</label>
86                     <input type="text" id="judul" name="judul" class="form-
control" value="<?= $buku['judul'] ?>">
87                     <div id="judulError" class="error"></div>
88                 </div>
89                 <div class="mb-3">
90                     <label for="penulis" class="form-label">Penulis</label>
91                     <input type="text" id="penulis" name="penulis"
class="form-control" value="<?= $buku['penulis'] ?>">
92                     <div id="penulisError" class="error"></div>
93                 </div>
94                 <div class="mb-3">
95                     <label for="penerbit" class="form-label">Penerbit</label>
96                     <input type="text" id="penerbit" name="penerbit"
class="form-control" value="<?= $buku['penerbit'] ?>">
97                     <div id="penerbitError" class="error"></div>

```

```

98         </div>
99         <div class="mb-3">
100             <label for="tahun_terbit" class="form-label">Tahun
Terbit</label>
101             <input type="number" id="tahun_terbit" name="tahun_terbit"
class="form-control" value="<?= $buku['tahun_terbit'] ?>">
102                 <div id="tahunTerbitError" class="error"></div>
103             </div>
104             <div class="button-container">
105                 <button type="submit" class="btn">Edit</button>
106             </div>
107         </form>
108     </div>
109 </div>
110
111     <script>
112         function validateForm() {
113             let isValid = true;
114             const judul = document.getElementById('judul').value;
115             const penulis = document.getElementById('penulis').value;
116             const penerbit = document.getElementById('penerbit').value;
117             const tahunTerbit = document.getElementById('tahun_terbit').value;
118
119             if (judul === '') {
120                 document.getElementById('judulError').innerText =
'Judul harus diisi.';
121                 isValid = false;
122             } else {
123                 document.getElementById('judulError').innerText = '';
124             }
125
126             if (penulis === '') {
127                 document.getElementById('penulisError').innerText =
'Penulis harus diisi.';
128                 isValid = false;
129             } else {
130                 document.getElementById('penulisError').innerText = '';
131             }
132
133             if (penerbit === '') {
134                 document.getElementById('penerbitError').innerText =
'Penerbit harus diisi.';
135                 isValid = false;
136             } else {
137                 document.getElementById('penerbitError').innerText = '';
138             }
139
140             if (tahunTerbit === '' || isNaN(tahunTerbit) ||
tahunTerbit < 1800 || tahunTerbit > 2024) {
141                 document.getElementById('tahunTerbitError').innerText =
'Tahun
terbit harus diisi dan berada di antara 1800 dan 2024.';
142                 isValid = false;

```

| | |
|-----|---|
| 143 | }else{ |
| 144 | document.getElementById('tahunTerbitError').innerText = ''; |
| 145 | } |
| 146 | |
| 147 | return isValid; |
| 148 | } |
| 149 | </script> |
| 150 | </body> |
| 151 | </html> |

Tabel 6. Source Code Edit.php

Index.php

| | |
|----|--|
| 1 | <!DOCTYPE html> |
| 2 | <html lang="en"> |
| 3 | |
| 4 | <head> |
| 5 | <meta charset="UTF-8"> |
| 6 | <meta http-equiv="X-UA-Compatible" content="IE=edge"> |
| 7 | <meta name="viewport" content="width=device-width, initial-scale=1.0"> |
| 8 | <title>Home</title> |
| 9 | <style> |
| 10 | body { |
| 11 | background-color: #393E46; |
| 12 | font-family: Arial, sans-serif; |
| 13 | margin: 0; |
| 14 | padding: 0; |
| 15 | } |
| 16 | |
| 17 | .container { |
| 18 | max-width: 900px; |
| 19 | margin: 50px auto; |
| 20 | padding: 20px; |
| 21 | background-color: #EEEEEE; |
| 22 | border-radius: 10px; |
| 23 | box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1); |
| 24 | } |
| 25 | |
| 26 | .header { |
| 27 | text-align: center; |
| 28 | margin-bottom: 40px; |
| 29 | background-color: #222831; |
| 30 | color: #fff; |
| 31 | padding: 20px; |
| 32 | border-radius: 10px 10px 0 0; |
| 33 | } |
| 34 | |
| 35 | .content { |
| 36 | margin-bottom: 30px; |
| 37 | text-align: center; |

```
38     }
39
40     h1 {
41         font-size: 28px;
42         font-weight: bold;
43         color: #fff;
44         margin: 0;
45     }
46
47     h2 {
48         font-size: 24px;
49         font-weight: bold;
50         margin-bottom: 15px;
51         color: #333;
52     }
53
54     .btn {
55         display: inline-block;
56         padding: 10px 20px;
57         font-size: 14px;
58         border-radius: 5px;
59         text-decoration: none;
60         color: #fff;
61         text-align: center;
62         transition: background-color 0.3s ease;
63     }
64
65     .btn-info {
66         background-color: #00ADB5;
67         border: none;
68     }
69
70     .btn-info:hover {
71         background-color: #138496;
72     }
73
74     .btn-danger {
75         background-color: #dc3545;
76         border: none;
77     }
78
79     .btn-danger:hover {
80         background-color: #c82333;
81     }
82
83     .table {
84         width: 100%;
85         border-collapse: collapse;
86         margin-bottom: 30px;
87         background-color: #fff;
88     }
89
```

```
90         .table th,  
91         .table td {  
92             padding: 10px;  
93             border: 1px solid #ddd;  
94             vertical-align: middle;  
95         }  
96  
97         .table th {  
98             background-color: #f8f9fa;  
99             color: #333;  
100            font-weight: bold;  
101            text-align: left;  
102        }  
103  
104        .right-align {  
105            text-align: right;  
106        }  
107  
108        .table-actions {  
109            display: flex;  
110            gap: 5px;  
111            align-items: center;  
112        }  
113  
114        .table-actions a,  
115        .table-actions form {  
116            display: inline-block;  
117            margin: 0;  
118        }  
119  
120        .logout-container {  
121            text-align: center;  
122            margin-top: 20px;  
123        }  
124  
125        .header,  
126        .content,  
127        .table-container {  
128            margin-bottom: 20px;  
129        }  
130    </style>  
131 </head>  
132  
133 <body>  
134     <div class="container">  
135         <div class="header">  
136             <h1>Selamat Datang di Perpustakaan</h1>  
137         </div>  
138  
139         <div class="content">  
140             <h2>Daftar Buku</h2>
```

```

141     <a class="btn btn-info" href="<?= base_url('/buku/create')
    ?>">Tambah Data</a>
142         </div>
143
144         <div class="table-container">
145             <table class="table">
146                 <thead>
147                     <tr>
148                         <th>Judul</th>
149                         <th>Penulis</th>
150                         <th>Penerbit</th>
151                         <th>Tahun Terbit</th>
152                         <th>Aksi</th>
153                     </tr>
154                 </thead>
155                 <tbody>
156                     <?php foreach ($buku as $key => $value): ?>
157                         <tr>
158                             <td><?= $value['judul'] ?></td>
159                             <td><?= $value['penulis'] ?></td>
160                             <td><?= $value['penerbit'] ?></td>
161                             <td><?= $value['tahun_terbit'] ?></td>
162                             <td class="table-actions">
163                                 <a href="<?= base_url('/buku/' . $value['id'] . '/edit')
    ?>" class="btn btn-info btn-sm">Edit</a>
164                                 form action="<?= base_url('/buku/' . $value['id'] .
    '/delete') ?>" method="POST"
165                                 onsubmit="return confirm('Are you sure you want to delete
    this item?')" style="display:inline;">
166                                 <input type="hidden" name="_method" value="DELETE"
    class="form-hidden">
167                                 <button type="submit" class="btn btn-danger btn-
    sm">Hapus</button>
168
169                                 </form>
170                             </td>
171                         </tr>
172                     <?php endforeach; ?>
173                 </tbody>
174             </table>
175         </div>
176
177         <div class="logout-container">
178             <a href="<?= base_url('logout') ?>" class="btn btn-
    info">Logout</a>
179         </div>
180 </body>
181
182 </html>

```

Tabel 7. Source Code Index.php

Login.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-
scale=1.0">
8      <title>Login</title>
9      <style>
10         body {
11             background-color: #393E46;
12             font-family: Arial, sans-serif;
13             margin: 0;
14             padding: 0;
15         }
16
17         .container {
18             max-width: 400px;
19             margin: 100px auto;
20             padding: 20px;
21             background-color: #EEEEEE;
22             border-radius: 10px;
23             box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
24         }
25
26         .login-title {
27             font-size: 24px;
28             font-weight: bold;
29             margin-bottom: 30px;
30             color: #333;
31             text-align: center;
32         }
33
34         .form-outline {
35             margin-bottom: 20px;
36         }
37
38         .form-label {
39             font-weight: 600;
40             display: block;
41             margin-bottom: 5px;
42             color: #555;
43         }
44
45         .form-control {
46             width: 100%;
47             padding: 10px;
48             border-radius: 5px;
49             border: 1px solid #ddd;
```

```
50         box-sizing: border-box;
51     }
52
53     .btn {
54         display: inline-block;
55         width: 100%;
56         padding: 10px;
57         font-size: 16px;
58         border-radius: 5px;
59         text-align: center;
60         text-decoration: none;
61         color: #fff;
62         background-color: #00ADB5;
63         border: none;
64         cursor: pointer;
65         transition: background-color 0.3s ease;
66     }
67
68     .btn:hover {
69         background-color: #138496;
70     }
71
72     .text-center {
73         text-align: center;
74     }
75
76     .alert {
77         padding: 15px;
78         background-color: #f8d7da;
79         color: #721c24;
80         border: 1px solid #f5c6cb;
81         border-radius: 5px;
82         margin-bottom: 20px;
83         position: relative;
84     }
85
86     .alert-dismissible .btn-close {
87         position: absolute;
88         top: 10px;
89         right: 10px;
90         background: none;
91         border: none;
92         font-size: 20px;
93         cursor: pointer;
94         color: #721c24;
95     }
96
97     .mt-4 {
98         margin-top: 20px;
99     }
100
101     .alert .btn-close:hover {
```



```

102         color: #000;
103     }
104
105     a {
106         color: #00ADB5;
107         text-decoration: none;
108     }
109
110     a:hover {
111         text-decoration: underline;
112     }
113
114     .forgot-password {
115         display: block;
116         text-align: center;
117         margin-top: 10px;
118         color: #00ADB5;
119     }
120 </style>
121 </head>
122
123 <body>
124     <div class="container">
125         <?php if (!empty(session()->getFlashdata('pesan'))): ?>
126         <div class="alert alert-warning alert-dismissible fade
show" role="alert">
127             <?php echo session()->getFlashdata('pesan'); ?>
128             <button type="button" class="btn-close" aria-
label="Close">&times;</button>
129         </div>
130         <?php endif; ?>
131         <div class="login-form">
132             <h2 class="login-title">Login</h2>
133             <form action="<?= base_url('/login') ?>"
method="post">
134                 <div class="form-outline mb-4">
135                     <label class="form-label" for="email">Username/Email
address</label>
136                     <input id="email" class="form-control" name="username"
value="<?= old('username') ?>" />
137                 </div>
138
139                 <div class="form-outline mb-4">
140                     <label class="form-label" for="password">Password</label>
141                     <input type="password" id="password" name="password"
class="form-control" />
142                 </div>
143
144                 <button type="submit" class="btn btn-primary btn-
block">Sign In</button>
145             </form>

```

```

146     <a href="<?= base_url('/forgot-password') ?>" class="forgot-
password">Forgot Password?</a>
147         <div class="text-center mt-4">
148             <p>Don't have an account? <a href="<?= base_url('/signup')
?>">Sign Up</a></p>
149         </div>
150     </div>
151 </div>
152
153 <script>
154     document.querySelectorAll('.btn-close').forEach(button => {
155         button.addEventListener('click', () => {
156             button.parentElement.style.display = 'none';
157         });
158     });
159 </script>
160 </body>
161
162 </html>

```

Tabel 8. Source Code Login.php

BukuModel.php

```

1  <?php namespace App\Models;
2
3  use CodeIgniter\Model;
4
5  class BukuModel extends Model
6  {
7      protected $table      = 'buku';
8      protected $primaryKey = 'id';
9
10     protected $returnType     = 'array';
11     protected $useSoftDeletes = true;
12
13     protected $allowedFields = ['judul', 'penulis', 'penerbit',
'tahun_terbit'];
14
15     protected $useTimestamps = false;
16
17     protected $validationRules = [
18         'judul' => 'required|string',
19         'penulis' => 'required|string',
20         'penerbit' => 'required|string',
21         'tahun_terbit' => 'required|numeric|greater_than[1800]|less_than[
2024]';
22     ];
23     protected $validationMessages = [
24         'judul' => [
25             'required' => 'Judul harus diisi',
26             'string' => 'Judul harus berupa string'

```

```

27         ],
28         'penulis' => [
29             'required' => 'Penulis harus diisi',
30             'string' => 'Penulis harus berupa string'
31         ],
32         'penerbit' => [
33             'required' => 'Penerbit harus diisi',
34             'string' => 'Penerbit harus berupa string'
35         ],
36         'tahun_terbit' => [
37             'required' => 'Tahun Terbit harus diisi',
38             'numeric' => 'Tahun Terbit harus berupa angka',
39             'greater_than' => 'Tahun Terbit harus lebih besar dari 1800',
40             'less_than' => 'Tahun Terbit harus kurang dari 2024'
41         ]
42     ];
43 }

```

Tabel 9. Source Code BukuModel.php

UserModel.php

```

1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  class UserModel extends Model
8  {
9      protected $table          = 'user';
10     protected $primaryKey     = 'id';
11
12     protected $returnType      = 'array';
13     protected $useSoftDeletes  = false;
14
15     protected $allowedFields   = ['username', 'email', 'password'];
16
17     protected $useTimestamps   = true;
18
19     protected $validationRules = [
20         'username' => 'required|alpha_numeric_space|min_length[3]|is_unique[user.username]',
21         'email'    => 'required|valid_email|is_unique[user.email]',
22         'password' => 'required|min_length[8]',
23     ];
24     protected $validationMessages = [
25         'username' => [
26             'required' => 'Username is required',
27             'alpha_numeric_space' => 'Username can only contain alphanumeric characters and spaces',
28             'min_length' => 'Username must be at least 3 characters long',

```

| | |
|----|---|
| 29 | 'is_unique' => 'Username already exists' |
| 30 |], |
| 31 | 'email' => [|
| 32 | 'required' => 'Email is required', |
| 33 | 'valid_email' => 'Please enter a valid email address', |
| 34 | 'is_unique' => 'Email already exists' |
| 35 |], |
| 36 | 'password' => [|
| 37 | 'required' => 'Password is required', |
| 38 | 'min_length' => 'Password must be at least 8 characters long' |
| 39 |] |
| 40 |]; |
| 41 | protected \$skipValidation = false; |
| 42 | } |

Tabel 10. Source Code UserModel.php

UserSeeder.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('user')->insert([|
| 12 | 'username' => 'Hafiz', |
| 13 | 'email' => 'mhmmhdhafizansari12@gmail.com', |
| 14 | 'password' => password_hash('12345678', PASSWORD_DEFAULT), |
| 15 |]); |
| 16 | } |
| 17 | } |

Tabel 11. Source Code UserSeeder.php

Buku.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             'constraint' => 20,
15             'unsigned' => true,
16             'auto_increment' => true,
17         ],
18         'judul' => [
19             'type' => 'VARCHAR',
20             'constraint' => '255',
21         ],
22         'penulis' => [
23             'type' => 'VARCHAR',
24             'constraint' => '100',
25         ],
26         'penerbit' => [
27             'type' => 'VARCHAR',
28             'constraint' => '100',
29         ],
30         'tahun_terbit' => [
31             'type' => 'YEAR'
32         ],
33     ];
34     $this->forge->addKey('id', true);
35     $this->forge->createTable('buku');
36 }
37
38 public function down()
39 {
40     $this->forge->dropTable('buku', true);
41 }
42 }

```

Tabel 12. Source Code Buku.php

User.php

```

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

```

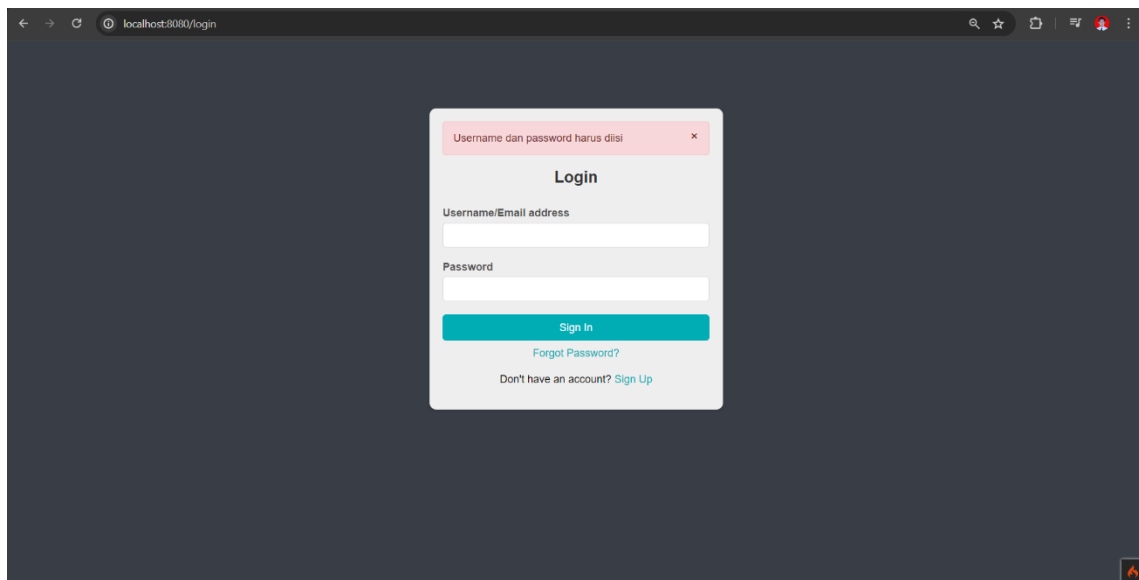
```

17         ],
18         'username' => [
19             'type' => 'VARCHAR',
20             'constraint' => '255',
21         ],
22         'email' => [
23             'type' => 'VARCHAR',
24             'constraint' => '255',
25         ],
26         'password' => [
27             'type' => 'TEXT',
28         ]
29     ]);
30     $this->forge->addKey('id', true);
31     $this->forge->createTable('user', true);
32 }
33
34 public function down()
35 {
36     $this->forge->dropTable('user', true);
37 }
38 }

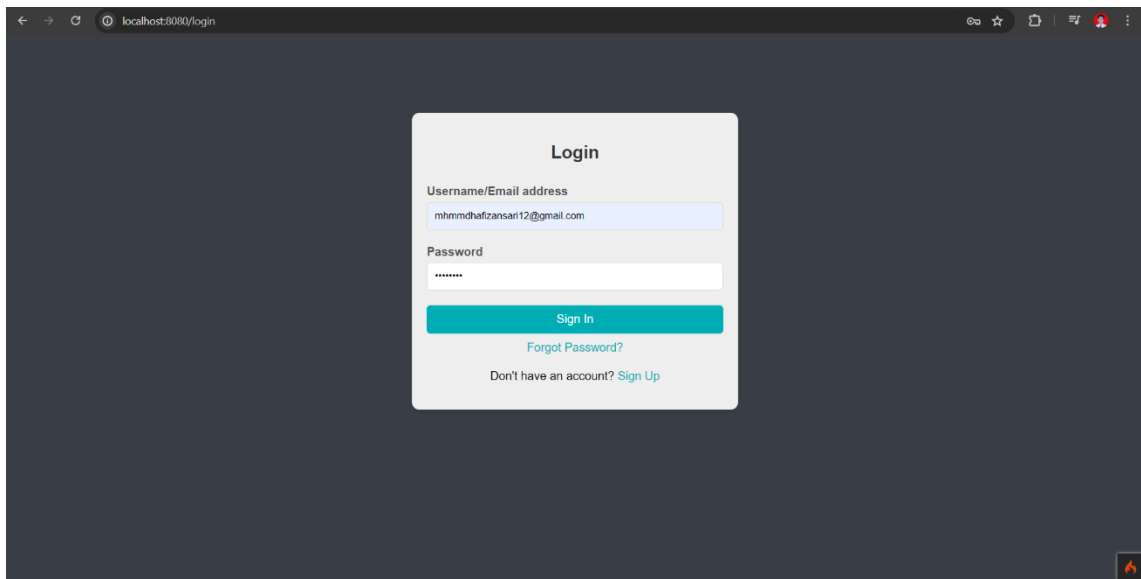
```

Tabel 13. Source Code User.php

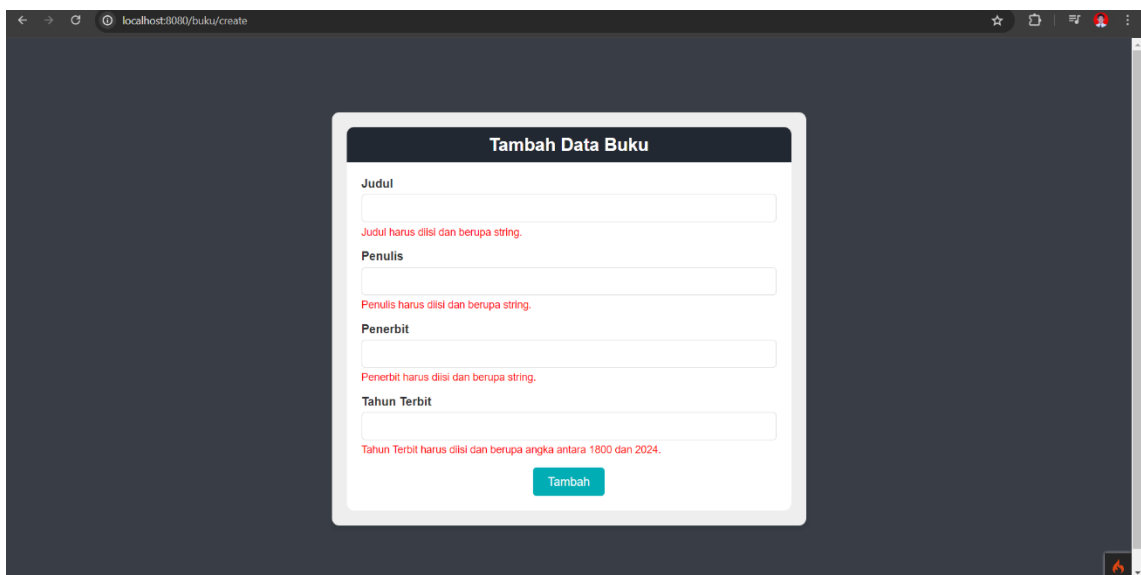
B. Output Program



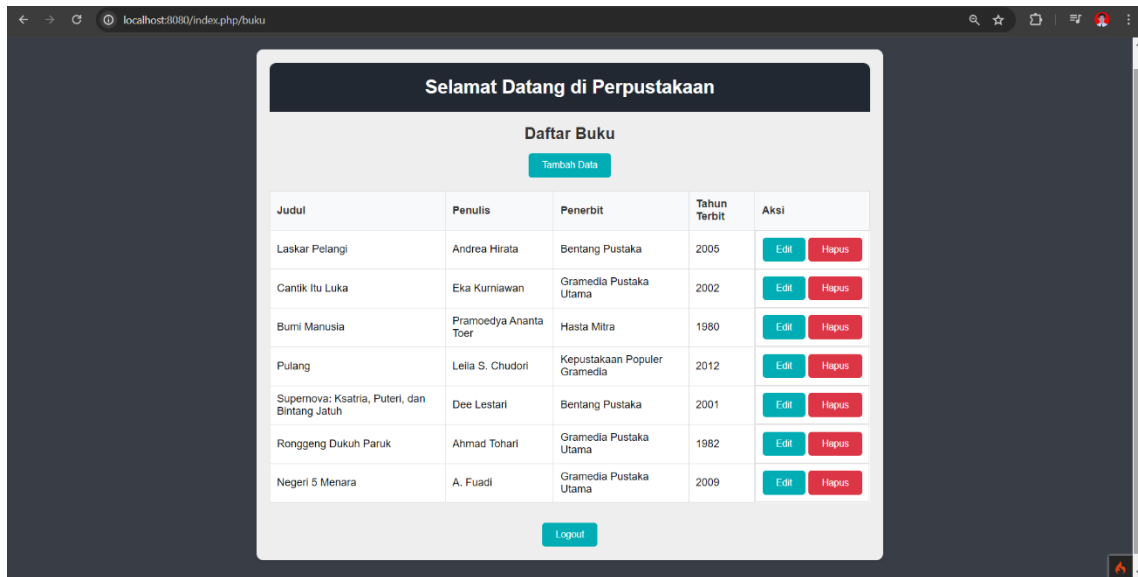
Gambar 2. Screenshot 1 Hasil Jawaban Praktikum 6



Gambar 3. Screenshot 2 Hasil Jawaban Praktikum 6



Gambar 4. Screenshot 3 Hasil Jawaban Praktikum 6



Gambar 5. Screenshot 4 Hasil Jawaban Praktikum 6

C. Pembahasan

Routes.php :

Pada baris [3] : Baris ini mendefinisikan namespace Config untuk file ini. Namespace membantu mengorganisir kode dan menghindari konflik nama.

Pada baris [5] : Baris ini mendapatkan instance dari layanan routing. Services::routes() memanggil layanan routing yang disediakan oleh framework.

Pada baris [7] : Baris ini mengatur namespace default untuk controller menjadi App\Controllers.

Pada baris [8] : Baris ini mengatur controller default menjadi Home.

Pada baris [9] : Baris ini mengatur metode default menjadi index.

Pada baris [10] : Baris ini mengatur agar tanda strip (-) dalam URI tidak diterjemahkan menjadi garis bawah (_) dalam nama metode controller.

Pada baris [11] : Baris ini mengatur penanganan default untuk halaman 404.

Pada baris [13] : Baris ini mendefinisikan grup rute dengan filter auth. Filter ini akan memastikan bahwa rute dalam grup ini hanya bisa diakses oleh pengguna yang sudah terautentikasi.

Pada baris [14] : Baris ini merupakan rute untuk akses ke root (/) yang akan memanggil metode index dari BukuController.

Pada baris [15] : Baris ini merupakan rute untuk akses ke /buku yang akan memanggil metode index dari BukuController..

Pada baris [16] : Baris ini merupakan rute untuk akses ke /buku/create yang akan memanggil metode create dari BukuController.

Pada baris [17] : Baris ini merupakan rute untuk mengirim data ke /buku/create yang akan memanggil metode store dari BukuController.

Pada baris [18] : Baris ini merupakan rute untuk akses ke /buku/{id}/edit (dengan id sebagai angka) yang akan memanggil metode edit dari BukuController dengan parameter id.

Pada baris [19] : Baris ini merupakan rute untuk mengirim data ke /buku/{id}/edit yang akan memanggil metode update dari BukuController dengan parameter id.

Pada baris [20] : Baris ini merupakan rute untuk menghapus data di /buku/{id}/delete yang akan memanggil metode delete dari BukuController dengan parameter id.

Pada baris [21] : Baris ini merupakan rute untuk akses ke /logout yang akan memanggil metode logout dari UserController.

Pada baris [24] : Baris ini merupakan rute untuk akses ke /login yang akan memanggil metode index dari UserController.

Pada baris [25] : Baris ini merupakan rute untuk mengirim data ke /login yang akan memanggil metode login dari UserController.

Pada baris [27] : Baris ini merupakan rute untuk akses ke /signup yang akan memanggil metode signUp dari UserController dan memberi nama rute ini sebagai signup.

Pada baris [28] : Baris ini merupakan rute untuk mengirim data ke /signup yang akan memanggil metode signUpProcess dari UserController.

Pada baris [30] : Baris ini merupakan rute untuk akses ke /forgot-password yang akan memanggil metode forgotPassword dari UserController.

Pada baris [31] : Baris ini merupakan rute untuk mengirim data ke /forgot-password yang akan memanggil metode forgotPasswordProcess dari UserController.

Pada baris [33] – [35] : Baris ini memeriksa apakah ada file konfigurasi rute khusus untuk lingkungan yang sedang berjalan. Jika ada, file tersebut akan di-include. APPPATH adalah konstanta yang menunjuk ke direktori aplikasi, dan ENVIRONMENT adalah konstanta yang mendefinisikan lingkungan (seperti development, production, dll).

BukuController.php :

Pada baris [3] : Baris ini mendefinisikan namespace App\Controllers untuk file ini. Namespace membantu mengorganisir kode dan menghindari konflik nama

Pada baris [5] : Mengimpor BaseController dari namespace App\Controllers.

Pada baris [6] : Mengimpor BukuModel dari namespace App\Models.

Pada baris [8] : Mendefinisikan kelas BukuController yang merupakan turunan dari BaseController.

Pada baris [10] : Metode ini digunakan untuk menampilkan daftar buku.

Pada baris [11] – [15] : Mengembalikan tampilan buku/index dengan data semua buku yang diambil menggunakan findAll() dari BukuModel.

Pada baris [17] : Metode ini digunakan untuk menampilkan formulir pembuatan buku.

Pada baris [18] – [20] : Mengembalikan tampilan buku/create.

Pada baris [22] : Metode ini digunakan untuk menyimpan data buku baru ke database.

Pada baris [24] : Mendapatkan instance layanan validasi.

Pada baris [25] – [30] : Mengatur aturan validasi untuk input.

Pada baris [32] – [34] : Memeriksa apakah input valid. Jika tidak, redirect kembali ke formulir dengan input dan pesan error.

Pada baris [36] : Membuat instance baru dari BukuModel.

Pada baris [37] – [42] : Menyimpan data buku baru ke database.

Pada baris [44] : Mengarahkan kembali ke halaman daftar buku.

Pada baris [47] : Metode ini digunakan untuk menampilkan formulir pengeditan buku berdasarkan ID.

Pada baris [48] – [52] : Mengembalikan tampilan buku/edit dengan data buku yang diambil menggunakan find(\$id) dari BukuModel.

Pada baris [54] : Metode ini digunakan untuk memperbarui data buku berdasarkan ID.

Pada baris [56] : Mendapatkan instance layanan validasi.

Pada baris [57] – [62] : Mengatur aturan validasi untuk input.

Pada baris [64] – [66] : Memeriksa apakah input valid. Jika tidak, redirect kembali ke formulir dengan input dan pesan error.

Pada baris [68] : Membuat instance baru dari BukuModel.

Pada baris [69] – [74] : Memperbarui data buku di database berdasarkan ID.

Pada baris [76] : Mengarahkan kembali ke halaman daftar buku.

Pada baris [79] : Metode ini digunakan untuk menghapus data buku berdasarkan ID.

Pada baris [81] : Membuat instance baru dari BukuModel.

Pada baris [82] : Menghapus data buku dari database berdasarkan ID.

Pada baris [83] : Mengarahkan kembali ke halaman daftar buku.

UserController.php :

Pada baris [3] – [6] : Mendefinisikan namespace App\Controllers dan mengimpor BaseController serta UserModel.

Pada baris [8] : Mendefinisikan kelas UserController yang mewarisi BaseController.

Pada baris [10] – [13] : Menampilkan halaman login dengan mengembalikan tampilan login.

Pada baris [15] – [43] : Menangani proses login pengguna. Mengecek input username dan password, memverifikasi keberadaan pengguna di database, memverifikasi kecocokan password, dan menetapkan sesi pengguna jika login berhasil. Jika ada kesalahan, mengarahkan kembali ke halaman login dengan pesan error.

Pada baris [45] – [49] : Menghancurkan sesi pengguna dan mengarahkan kembali ke halaman login.

Pada baris [51] – [57] : Menampilkan halaman pendaftaran. Jika pengguna sudah login, mereka akan diarahkan ke halaman daftar buku.

Pada baris [59] – [89] : Menangani proses pendaftaran pengguna baru. Melakukan validasi input, menyimpan data pengguna baru ke database dengan password yang telah dienkripsi, menetapkan sesi pengguna, dan mengarahkan pengguna ke halaman daftar buku setelah pendaftaran berhasil.

Pada baris [91] – [94] : Menampilkan halaman pemulihan kata sandi dengan mengembalikan tampilan forgot password.

Pada baris [96] – [110] : Menangani proses pemulihan kata sandi. Mengecek keberadaan email pengguna di database dan menetapkan pesan bahwa email untuk reset password telah dikirim. Jika email tidak ditemukan, menetapkan pesan error dan mengarahkan kembali ke halaman lupa password.

AuthFilter.php :

Pada baris [3] : Mendefinisikan namespace App\Filters untuk file ini.

Pada baris [5] – [7] : Mengimpor FilterInterface, RequestInterface, dan ResponseInterface dari framework CodeIgniter.

Pada baris [9] : Mendefinisikan kelas AuthFilter yang mengimplementasikan FilterInterface.

Pada baris [11] : Metode ini akan dijalankan sebelum request diproses.

Pada baris [13] : Mengecek apakah sesi logged_in ada dan bernilai true. Jika tidak, pengguna dianggap belum login.

Pada baris [14] : Menetapkan pesan flash “Login terlebih dahulu!” ke sesi.

Pada baris [15] : Mengarahkan pengguna ke halaman login.

Pada baris [19] : Metode ini akan dijalankan setelah request diproses. Saat ini, metode ini tidak melakukan apa-apa.

Create.php :

Pada baris [1] – [2] : Mendefinisikan tipe dokumen sebagai HTML5 dan menyetel bahasa dokumen ke Bahasa Inggris.

Pada baris [4] – [8] : Bagian <head> mengandung meta tag untuk set karakter dan viewport, serta tag <title> untuk judul halaman.

Pada baris [9] – [92] : Berisi CSS internal yang digunakan untuk mendefinisikan gaya tampilan halaman, termasuk warna latar, font, dan desain form.

Pada baris [96] – [123] : Sebuah container yang berisi form untuk menambahkan data buku. Form ini memiliki field untuk judul, penulis, penerbit, dan tahun terbit.

Pada baris [125] – [162] : Fungsi validateForm: Fungsi ini dijalankan ketika form disubmit. Fungsi ini memeriksa apakah input-input form sudah diisi dengan benar sesuai kriteria yang ditentukan:

- Judul: Harus diisi dan harus berupa string.

- Penulis: Sama seperti judul, harus diisi dan harus berupa string.
- Penerbit: Sama seperti judul, harus diisi dan harus berupa string.
- Tahun Terbit: Harus berupa angka di antara 1800 dan 2024.
- Jika ada kesalahan, pesan error akan ditampilkan dan form tidak akan disubmit.

Pada baris [163] – [165] : Menutup tag body dan HTML, menandai akhir dari dokumen HTML.

Edit.php :

HTML

- Head Section: Mengatur charset, compatibility, dan viewport. Tag <title> menetapkan judul halaman sebagai "Edit Data Buku".
- CSS Styles: Digunakan untuk mengatur tampilan halaman, seperti warna latar belakang, pengaturan font, margin, padding, dan desain elemen form seperti input fields dan tombol.
- Body Section:
 - Container: Kotak yang berisi keseluruhan form, yang diatur untuk tampil di tengah dan memiliki gaya visual yang menarik.
 - Header: Bagian kepala dengan judul "Edit Data Buku", yang menunjukkan fungsi halaman.
 - Form:
 - Terdiri dari beberapa div yang masing-masing mengandung label dan input untuk judul, penulis, penerbit, dan tahun terbit.
 - Nilai awal diatur dengan menggunakan data buku yang sudah ada (diambil dari variabel \$buku yang berisi data dari database).
 - Tombol submit untuk mengirimkan formulir setelah diedit.

JavaScript

- Fungsi validateForm(): Digunakan untuk validasi klien sebelum data dikirim ke server. Fungsi ini memeriksa:
 - Judul, Penulis, dan Penerbit: Harus diisi. Jika tidak, akan menampilkan pesan error.
 - Tahun Terbit: Harus diisi dengan angka valid antara 1800 dan 2024. Jika input tidak memenuhi kriteria ini, pesan error akan ditampilkan.
- JavaScript juga bertugas untuk mengosongkan pesan error jika masukan sudah diperbaiki dan valid.

Proses Pengeditan

- Form dikirim ke URL yang ditentukan dalam atribut action, yang menggabungkan base URL dengan ID buku dan endpoint untuk edit. Hal ini menunjukkan bahwa data form akan diproses oleh metode edit dari controller buku yang sesuai dengan ID buku.

Index.php :

Sebuah implementasi HTML, CSS, dan PHP yang digunakan untuk membuat halaman web yang mengelola daftar buku di perpustakaan. HTML mendefinisikan struktur halaman yang terdiri dari header, konten utama, dan sebuah tombol logout, yang semuanya dikelompokkan dalam sebuah div container untuk pemformatan. CSS digunakan untuk memberikan gaya pada elemen-elemen ini,

mengatur warna latar, font, dan efek interaksi seperti hover pada tombol, serta menentukan tampilan tabel yang memuat data buku. PHP diintegrasikan untuk mengendalikan dinamika data dengan mengulangi setiap buku dalam array \$buku, menampilkan atributnya, dan menyediakan fungsi untuk mengedit dan menghapus buku menggunakan form yang di-trigger dengan aksi POST. Selain itu, PHP juga digunakan untuk membangun URL yang dinamis yang diperlukan untuk operasi CRUD pada buku. Kode ini mencerminkan bagaimana HTML, CSS, dan PHP dapat bekerja bersama untuk membuat aplikasi web yang tidak hanya fungsional tapi juga visual menarik dan responsif.

Login.php :

Implementasi halaman login menggunakan HTML, CSS, dan PHP yang terintegrasi. Struktur HTML mendefinisikan sebuah form login di dalam sebuah container yang terpusat di tengah halaman. CSS digunakan untuk mengatur tampilan halaman, memberikan warna latar belakang yang gelap pada body dan warna latar yang terang pada container untuk kontras yang baik. Terdapat berbagai elemen form seperti input untuk username/email dan password yang diformat dengan gaya CSS agar tampil lebih menarik dan mudah digunakan. Tombol "Sign In" yang disediakan akan mengirimkan data form ke endpoint login menggunakan metode POST.

Jika terdapat pesan flash dari session, seperti pesan kesalahan, maka akan ditampilkan dalam sebuah elemen alert yang bisa ditutup oleh pengguna dengan mengklik tombol "close". Fungsi JavaScript kecil ditambahkan untuk menangani penutupan elemen alert ini secara dinamis. Di bagian bawah form, terdapat tautan "Forgot Password?" yang mengarahkan pengguna ke halaman pemulihan password dan sebuah pesan yang mengajak pengguna untuk mendaftar jika mereka belum memiliki akun, disertai dengan tautan ke halaman pendaftaran.

Dengan integrasi antara HTML untuk struktur, CSS untuk gaya, PHP untuk logika backend, dan JavaScript untuk interaksi dinamis, kode ini menyediakan solusi yang lengkap dan interaktif untuk halaman login di aplikasi web. Halaman ini dirancang untuk responsif dan user-friendly, memastikan pengguna memiliki pengalaman yang baik saat mencoba masuk ke dalam sistem.

BukuModel.php :

Pada baris [1] – [3] : Bagian ini mendeklarasikan namespace App\Models dan mengimpor kelas Model dari CodeIgniter yang akan dijadikan sebagai superclass.

Pada baris [5] : Deklarasi kelas BukuModel yang merupakan turunan dari kelas Model CodeIgniter.

Pada baris [7] – [15] :

- \$table: Nama tabel yang akan digunakan, yaitu buku.
- \$primaryKey: Kolom primary key dari tabel, yaitu id.
- \$returnType: Menentukan tipe data yang akan dikembalikan saat query dilakukan, yaitu array.
- \$useSoftDeletes: Mengaktifkan fitur soft delete, yang berarti data tidak benar-benar dihapus dari database melainkan hanya ditandai sebagai dihapus.
- \$allowedFields: Daftar kolom yang diizinkan untuk diisi atau diubah melalui model ini.
- \$useTimestamps: Menonaktifkan penggunaan otomatis kolom timestamp seperti created_at dan updated_at.

Pada baris [17] – [42] :

- `$validationRules`: Aturan validasi untuk kolom judul, penulis, penerbit, dan tahun_terbit. Setiap kolom harus diisi (required) dan memiliki tipe data tertentu (string untuk teks, numeric untuk angka). Selain itu, tahun_terbit harus berada dalam rentang tahun 1800 hingga 2023.
- `$validationMessages`: Pesan kesalahan yang akan ditampilkan jika validasi gagal. Pesan ini memberikan umpan balik spesifik tentang mengapa validasi gagal (misalnya, "Judul harus diisi" atau "Tahun Terbit harus lebih besar dari 1800").

UserModel.php :

Pada baris [1] – [3] : Bagian ini mendeklarasikan namespace `App\Models` dan mengimpor kelas `Model` dari `CodeIgniter` yang akan dijadikan superclass.

Pada baris [7] : Deklarasi kelas `UserModel` yang merupakan turunan dari kelas `Model` `CodeIgniter`.

Pada baris [9] – [17] :

- `$table`: Nama tabel yang akan digunakan, yaitu `user`.
- `$primaryKey`: Kolom primary key dari tabel, yaitu `id`.
- `$returnType`: Menentukan tipe data yang akan dikembalikan saat query dilakukan, yaitu `array`.
- `$useSoftDeletes`: Fitur soft delete dinonaktifkan, yang berarti data akan dihapus secara permanen dari database.
- `$allowedFields`: Daftar kolom yang diizinkan untuk diisi atau diubah melalui model ini. Dalam hal ini, kolom yang diizinkan adalah `username`, `email`, dan `password`.
- `$useTimestamps`: Mengaktifkan penggunaan otomatis kolom timestamp seperti `created_at` dan `updated_at`.

Pada baris [19] – [41] :

- `$validationRules`: Aturan validasi untuk kolom `username`, `email`, dan `password`. Setiap kolom harus memenuhi kriteria tertentu:
 - `username`: Harus diisi (required), hanya boleh mengandung karakter alfanumerik dan spasi (`alpha_numeric_space`), memiliki panjang minimal 3 karakter (`min_length[3]`), dan unik di tabel `user` (`is_unique[user.username]`).
 - `email`: Harus diisi (required), harus merupakan alamat email yang valid (`valid_email`), dan unik di tabel `user` (`is_unique[user.email]`).
 - `password`: Harus diisi (required) dan memiliki panjang minimal 8 karakter (`min_length[8]`).
- `$validationMessages`: Pesan kesalahan yang akan ditampilkan jika validasi gagal. Pesan ini memberikan umpan balik spesifik tentang mengapa validasi gagal.
- `$skipValidation`: Properti ini diatur ke `false`, yang berarti validasi akan selalu dilakukan sebelum data disimpan ke database.

UserSeeder.php :

Pada baris [3] : Baris ini menyatakan bahwa kelas `UserSeeder` berada di dalam namespace `App\Database\Seeds`. Namespace membantu dalam mengorganisir dan mengelola kelas di dalam aplikasi.

Pada baris [5] : Baris ini mengimpor kelas Seeder dari CodeIgniter yang akan dijadikan superclass untuk UserSeeder.

Pada baris [7] : Kelas UserSeeder dideklarasikan sebagai subclass dari Seeder. Ini berarti UserSeeder mewarisi semua properti dan metode dari kelas Seeder.

Pada baris [9] : Metode run adalah metode utama yang akan dipanggil saat seeder dijalankan. Metode ini bertanggung jawab untuk memasukkan data ke dalam database.

Pada baris [10] – [16] :

- `$this->db->table('user')`: Mengambil objek tabel user dari database yang terhubung.
- `->insert(...)`: Metode insert digunakan untuk memasukkan data baru ke dalam tabel. Data yang dimasukkan berupa array asosiatif di mana kunci array mewakili nama kolom dan nilai array adalah nilai yang akan dimasukkan ke kolom tersebut.
 - `'username' => 'Hafiz'`: Mengisi kolom username dengan nilai 'Hafiz'.
 - `'email' => 'mhmmmdhafizansari12@gmail.com'`: Mengisi kolom email dengan nilai 'mhmmmdhafizansari12@gmail.com'.
 - `'password' => password_hash('12345678', PASSWORD_DEFAULT)`: Mengisi kolom password dengan nilai hasil hash dari password '12345678' menggunakan algoritma hash default yang aman.

Buku.php :

Pada baris [3] : Baris ini menyatakan bahwa kelas Buku berada di dalam namespace App\Database\Migrations. Namespace membantu dalam mengorganisir dan mengelola kelas di dalam aplikasi.

Pada baris [5] : Baris ini mengimpor kelas Migration dari CodeIgniter yang akan dijadikan superclass untuk kelas Buku.

Pada baris [7] : Kelas Buku dideklarasikan sebagai subclass dari Migration. Ini berarti Buku mewarisi semua properti dan metode dari kelas Migration.

Pada baris [9] : Metode up adalah metode yang akan dipanggil ketika migrasi dijalankan. Metode ini bertanggung jawab untuk membuat tabel baru atau mengubah struktur tabel yang ada.

Pada baris [10] – [33] :

- `$this->forge->addField(...)`: Mendefinisikan kolom-kolom yang akan dibuat dalam tabel buku.
 - `'id'` adalah kolom primary key dengan tipe BIGINT, panjang maksimum 20, unsigned (hanya menerima nilai non-negatif), dan auto increment (nilai akan bertambah otomatis).
 - `'judul'` adalah kolom VARCHAR dengan panjang maksimum 255 karakter.
 - `'penulis'` adalah kolom VARCHAR dengan panjang maksimum 100 karakter.
 - `'penerbit'` adalah kolom VARCHAR dengan panjang maksimum 100 karakter.
 - `'tahun_terbit'` adalah kolom YEAR yang menyimpan tahun terbit buku.

Pada baris [34] : Baris ini mendefinisikan kolom id sebagai primary key dari tabel buku.

Pada baris [35] – [36] : Baris ini memerintahkan untuk membuat tabel buku dengan struktur yang telah didefinisikan.

Pada baris [38] : Metode down adalah metode yang akan dipanggil ketika migrasi dibatalkan atau di-rollback. Metode ini bertanggung jawab untuk menghapus tabel atau mengembalikan perubahan yang dibuat oleh metode up.

Pada baris [39] – [41] : Baris ini memerintahkan untuk menghapus tabel buku jika ada. Parameter kedua true memastikan bahwa operasi ini tidak menimbulkan kesalahan jika tabel tidak ada.

User.php :

Pada baris [3] : Baris ini menyatakan bahwa kelas User berada di dalam namespace App\Database\Migrations. Namespace membantu dalam mengorganisir dan mengelola kelas di dalam aplikasi.

Pada baris [5] : Baris ini mengimpor kelas Migration dari CodeIgniter yang akan dijadikan superclass untuk kelas User.

Pada baris [7] : Kelas User dideklarasikan sebagai subclass dari Migration. Ini berarti User mewarisi semua properti dan metode dari kelas Migration.

Pada baris [9] : Metode up adalah metode yang akan dipanggil ketika migrasi dijalankan. Metode ini bertanggung jawab untuk membuat tabel baru atau mengubah struktur tabel yang ada.

Pada baris [10] – [29] :

- `$this->forge->addField([...])`: Mendefinisikan kolom-kolom yang akan dibuat dalam tabel user.
- 'id' adalah kolom primary key dengan tipe INT, panjang maksimum 11, unsigned (hanya menerima nilai non-negatif), dan auto increment (nilai akan bertambah otomatis).
- 'username' adalah kolom VARCHAR dengan panjang maksimum 255 karakter.
- 'email' adalah kolom VARCHAR dengan panjang maksimum 255 karakter.
- 'password' adalah kolom TEXT yang menyimpan password dalam bentuk teks.

Pada baris [30] : Baris ini mendefinisikan kolom id sebagai primary key dari tabel user.

Pada baris [31] – [32] : Baris ini memerintahkan untuk membuat tabel user dengan struktur yang telah didefinisikan. Parameter kedua true memastikan bahwa tabel tidak akan dibuat ulang jika sudah ada.

Pada baris [34] : Metode down adalah metode yang akan dipanggil ketika migrasi dibatalkan atau di-rollback. Metode ini bertanggung jawab untuk menghapus tabel atau mengembalikan perubahan yang dibuat oleh metode up.

Pada baris [35] – [37] : Baris ini memerintahkan untuk menghapus tabel user jika ada. Parameter kedua true memastikan bahwa operasi ini tidak menimbulkan kesalahan jika tabel tidak ada.

D. Tautan Git

<https://github.com/MuhammadHafizAnsari/Praktikum-Pemrograman-Web-II/tree/ff37a8da771dac628f9a5e67571e5c9c92c4ea40/modul6/PRAK601>