

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Wanda Karliyanti NIM. 2210817220034

**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 : Wanda Karliyanti
NIM : 2210817220034

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	30
C. Pembahasan	31
LINK GITHUB	44

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	31
---	----

DAFTAR TABEL

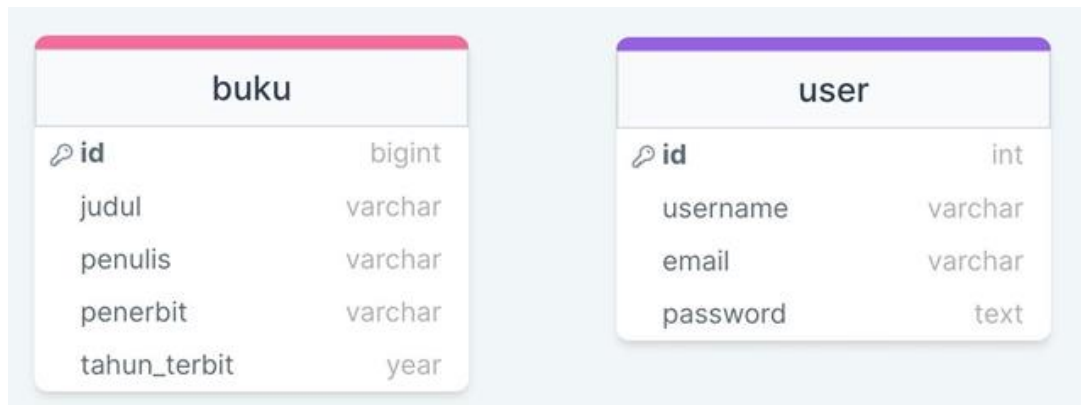
Tabel 1. Source Code Routes.php	7
Tabel 2. Source Code Auth.php	9
Tabel 3. Source Code Buku.php.....	13
Tabel 4. Source Code 2024-06-10-064318_CreateDbModul6.php	14
Tabel 5. Source Code UserSeeder.php	15
Tabel 6. Source Code AuthFilter.php	15
Tabel 7. Source Code BukuModel.php.....	16
Tabel 8. Source Code UserModel.php.....	17
Tabel 9. Source Code CreateView.php.....	21
Tabel 10. Source Code EditView.php.....	23
Tabel 11. Source Code IndexView.php	27
Tabel 12. Source Code LoginView.php.....	29

SOAL

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

Beberapa syarat yang harus ada pada website:

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



Simpan dalam folder github PRAK601

A. Source Code

```
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 $routes->setAutoRoute(false);
```

```

13
14 $routes->get('/', 'Home::index', ['filter' => 'auth']);
15 $routes->get('/buku', 'Buku::index', ['filter' =>
    'auth']);
16 $routes->get('/buku/create', 'Buku::create', ['filter'
    => 'auth']);
17 $routes->post('/buku/store', 'Buku::store', ['filter' =>
    'auth']);
18 $routes->get('/buku/(:num)/edit', 'Buku::edit/$1',
    ['filter' => 'auth']);
19 $routes->post('/buku/(:num)/update', 'Buku::update/$1',
    ['filter' => 'auth']);
20 $routes->post('/buku/(:num)/delete', 'Buku::delete/$1',
    ['filter' => 'auth']);
21 $routes->get('/dashboard', 'Home::dashboard', ['filter'
    => 'auth']);
22 $routes->get('/login', 'Auth::login');
23 $routes->post('/login', 'Auth::loginAuth');
24 $routes->get('/logout', 'Auth::logout');
25
26 // Include environment specific routes if available
27 if (is_file(APPPATH . 'Config/' . ENVIRONMENT .
    '/Routes.php')) {
28     require APPPATH . 'Config/' . ENVIRONMENT .
    '/Routes.php';
29 }

```

Tabel 1. Source Code Routes.php

```

1 <?php
2
3 namespace App\Controllers;
4
5 use App\Models\UserModel;
6 use CodeIgniter\Controller;
7
8 class Auth extends Controller{
9     public function store(){
10         $userModel = new UserModel();
11         $data = [
12             'username' => $this->request-
13 >getVar('username'),
14             'email' => $this->request-
15 >getVar('email'),
16             'password' => password_hash($this->request-
17 >getVar('password'), PASSWORD_DEFAULT),
18         ];
19         $userModel->save($data);
20         return redirect()->to('/login');
21     }
22 }

```

```

18     }
19
20     public function login(){
21         helper(['form']);
22         if ($this->request->getMethod() == 'post') {
23             return $this->loginAuth();
24         }
25         return view('LoginView');
26     }
27
28     public function loginAuth(){
29         $session = session();
30         $userModel = new UserModel();
31         $identifier = $this->request-
>getVar('username');
32         $password = $this->request->getVar('password');
33         $data = $userModel-
>getUserByEmailOrUsername($identifier);
34
35         if($data){
36             $pass = $data['password'];
37             $authenticatePassword =
password_verify($password, $pass);
38
39             if($authenticatePassword){
40                 $ses_data = [
41                     'id' => $data['id'],
42                     'username' => $data['username'],
43                     'logged_in' => TRUE
44                 ];
45                 $session->set($ses_data);
46                 return redirect()->to('/buku');
47             }else{
48                 $session->setFlashdata('msg', 'Password
salah');
49                 return redirect()->to('/login');
50             }
51         }else{
52             $session->setFlashdata('msg', 'Username atau
Email salah');
53             return redirect()->to('/login');
54         }
55     }
56
57     public function logout(){
58         $session = session();
59         $session->destroy();

```


60	return redirect()->to('/login');
61	}
62	}

Tabel 2. Source Code Auth.php

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Models\BukuModel;
6	use CodeIgniter\Controller;
7	
8	class Buku extends Controller{
9	public function __construct(){
10	helper(['form', 'url']);
11	}
12	
13	public function index(){
14	\$bukuModel = new BukuModel();
15	\$search = \$this->request->getGet('search');
16	\$sort = \$this->request->getGet('sort');
17	\$sort_order = \$this->request-
18	>getVar('sort_order');
19	\$query = \$bukuModel;
20	if (\$search) {
21	\$query = \$query->like('judul', \$search)
22	->orLike('penulis', \$search)
23	->orLike('penerbit', \$search)
24	->orLike('tahun_terbit', \$search);
25	}
26	
27	if (\$sort && in_array(\$sort, ['judul',
28	'penulis', 'penerbit', 'tahun_terbit'])) {
29	\$bukuModel->orderBy(\$sort, \$sort_order ?
30	\$sort_order : 'ASC');
31	}
32	\$data['bukus'] = \$query->findAll();
33	\$data['search'] = \$search;
34	\$data['sort'] = \$sort;
35	\$data['sort_order'] = \$sort_order;
36	return view('IndexView', \$data);
37	}
38	
39	public function create(){
40	return view('CreateView');

```

41     }
42
43     public function store(){
44         $validation = \Config\Services::validation();
45
46         $validation->setRules([
47             'judul' => [
48                 'label' => 'Judul',
49                 'rules' => 'required|string',
50                 'errors' => [
51                     'required' => 'Judul harus diisi.',
52                     'string' => 'Judul harus berupa
string.'
53                 ]
54             ],
55             'penulis' => [
56                 'label' => 'Penulis',
57                 'rules' => 'required|string',
58                 'errors' => [
59                     'required' => 'Penulis harus
diisi.',
60                     'string' => 'Penulis harus berupa
string.'
61                 ]
62             ],
63             'penerbit' => [
64                 'label' => 'Penerbit',
65                 'rules' => 'required|string',
66                 'errors' => [
67                     'required' => 'Penerbit harus
diisi.',
68                     'string' => 'Penerbit harus berupa
string.'
69                 ]
70             ],
71             'tahun_terbit' => [
72                 'label' => 'Tahun Terbit',
73                 'rules' => 'required|numeric|greater_than[1800]|less_than[2024]',
74                 'errors' => [
75                     'required' => 'Tahun Terbit harus
diisi.',
76                     'numeric' => 'Tahun Terbit harus
berupa angka.',
77                     'greater_than' => 'Tahun Terbit
harus lebih besar dari 1800.',

```

```

78         'less_than' => 'Tahun Terbit harus
kurang dari 2024.'
79     ]
80 ]
81 });
82
83     if($validation->withRequest($this->request)-
>run() == FALSE){
84         return view('CreateView', ['validation' =>
$validation]);
85     }else{
86         $bukuModel = new BukuModel();
87         $data = [
88             'judul' => $this->request-
>getPost('judul'),
89             'penulis' => $this->request-
>getPost('penulis'),
90             'penerbit' => $this->request-
>getPost('penerbit'),
91             'tahun_terbit' => $this->request-
>getPost('tahun_terbit'),
92         ];
93         $bukuModel->save($data);
94         return redirect()->to('/buku');
95     }
96 }
97
98 public function edit($id){
99     $bukuModel = new BukuModel();
100     $data['book'] = $bukuModel->find($id);
101     return view('EditView', $data);
102 }
103
104 public function update($id){
105     $validation = \Config\Services::validation();
106
107     $validation->setRules([
108         'judul' => [
109             'label' => 'Judul',
110             'rules' => 'required|string',
111             'errors' => [
112                 'required' => 'Judul harus diisi.',
113                 'string' => 'Judul harus berupa
string.'
114             ]
115         ],
116         'penulis' => [

```

```

117         'label' => 'Penulis',
118         'rules' => 'required|string',
119         'errors' => [
120             'required' => 'Penulis harus
diisi.',
121             'string' => 'Penulis harus berupa
string.'
122         ]
123     ],
124     'penerbit' => [
125         'label' => 'Penerbit',
126         'rules' => 'required|string',
127         'errors' => [
128             'required' => 'Penerbit harus
diisi.',
129             'string' => 'Penerbit harus berupa
string.'
130         ]
131     ],
132     'tahun_terbit' => [
133         'label' => 'Tahun Terbit',
134         'rules' =>
'required|numeric|greater_than[1800]|less_than[2024]',
135         'errors' => [
136             'required' => 'Tahun Terbit harus
diisi.',
137             'numeric' => 'Tahun Terbit harus
berupa angka.',
138             'greater_than' => 'Tahun Terbit
harus lebih besar dari 1800.',
139             'less_than' => 'Tahun Terbit harus
kurang dari 2024.'
140         ]
141     ]
142     ]);
143
144     if($validation->withRequest($this->request)-
>run() == FALSE){
145         $bukuModel = new BukuModel();
146         return view('EditView', [
147             'validation' => $validation,
148             'book' => $bukuModel->find($id)
149         ]);
150     }else{
151         $bukuModel = new BukuModel();
152         $data = [

```

153	<pre> 'judul' => \$this->request- >getPost('judul'), </pre>
154	<pre> 'penulis' => \$this->request- >getPost('penulis'), </pre>
155	<pre> 'penerbit' => \$this->request- >getPost('penerbit'), </pre>
156	<pre> 'tahun_terbit' => \$this->request- >getPost('tahun_terbit'), </pre>
157	<pre>]; </pre>
158	<pre> \$bukuModel->update(\$id, \$data); </pre>
159	<pre> return redirect()->to('/buku'); </pre>
160	<pre> } </pre>
161	<pre> } </pre>
162	
163	<pre> public function delete(\$id){ </pre>
164	<pre> \$bukuModel = new BukuModel(); </pre>
165	<pre> \$bukuModel->delete(\$id); </pre>
166	<pre> return redirect()->to('/buku'); </pre>
167	<pre> } </pre>
168	<pre> } </pre>

Tabel 3. Source Code Buku.php

1	<?php
2	
3	namespace App\Database\Migrations;
4	
5	use CodeIgniter\Database\Migration;
6	
7	class CreateDbModul6 extends Migration{
8	public function up(){
9	\$this->forge->addField([
10	'id' => [
11	'type' => 'INT',
12	'constraint' => 7,
13	'unsigned' => TRUE,
14	'auto_increment' => TRUE
15],
16	'username' => [
17	'type' => 'VARCHAR',
18	'constraint' => '200',
19],
20	'email' => [
21	'type' => 'VARCHAR',
22	'constraint' => '200',
23],
24	'password' => [
25	'type' => 'TEXT',
26],

27]);
28	\$this->forge->addKey('id', TRUE);
29	\$this->forge->createTable('user');
30	
31	\$this->forge->addField([
32	'id' => [
33	'type' => 'BIGINT',
34	'constraint' => 10,
35	'unsigned' => TRUE,
36	'auto_increment' => TRUE
37],
38	'judul' => [
39	'type' => 'VARCHAR',
40	'constraint' => '250',
41],
42	'penulis' => [
43	'type' => 'VARCHAR',
44	'constraint' => '250',
45],
46	'penerbit' => [
47	'type' => 'VARCHAR',
48	'constraint' => '250',
49],
50	'tahun_terbit' => [
51	'type' => 'YEAR',
52],
53]);
54	\$this->forge->addKey('id', TRUE);
55	\$this->forge->createTable('buku');
56	}
57	
58	public function down(){
59	\$this->forge->dropTable('user');
60	\$this->forge->dropTable('buku');
61	}
62	}

Tabel 4. Source Code 2024-06-10-064318_CreateDbModul6.php

1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;
6	
7	class UserSeeder extends Seeder{
8	public function run(){
9	\$data = [
10	'username' => 'wnda',

11	'email' => 'pengguna123@example.com',
12	'password' => password_hash('password',
13	PASSWORD_DEFAULT),
14];
15	\$this->db->table('user')->insert(\$data);
16	}
17	}

Tabel 5. Source Code UserSeeder.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	public function before(RequestInterface \$request,
11	\$args = null){
12	if (!session()->get('logged_in')) {
13	return redirect()->to('/login')->
14	with('msg', 'Login terlebih dahulu!');
15	}
16	public function after(RequestInterface \$request,
17	ResponseInterface \$response, \$args = null){
18	// Do nothing
19	}

Tabel 6. Source Code AuthFilter.php

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class BukuModel extends Model{
8	protected \$table = 'buku';
9	protected \$primaryKey = 'id';
10	protected \$useAutoIncrement = true;
11	protected \$returnType = 'array';
12	protected \$useSoftDeletes = false;
13	protected \$protectFields = true;

14	protected \$allowedFields = ['judul', 'penulis', 'penerbit', 'tahun_terbit'];
15	
16	protected bool \$allowEmptyInserts = false;
17	protected bool \$updateOnlyChanged = true;
18	
19	protected array \$casts = [];
20	protected array \$castHandlers = [];
21	
22	// Dates
23	protected \$useTimestamps = false;
24	protected \$dateFormat = 'datetime';
25	protected \$createdField = 'created_at';
26	protected \$updatedField = 'updated_at';
27	protected \$deletedField = 'deleted_at';
28	
29	// Validation
30	protected \$validationRules = [];
31	protected \$validationMessages = [];
32	protected \$skipValidation = false;
33	protected \$cleanValidationRules = true;
34	
35	// Callbacks
36	protected \$allowCallbacks = true;
37	protected \$beforeInsert = [];
38	protected \$afterInsert = [];
39	protected \$beforeUpdate = [];
40	protected \$afterUpdate = [];
41	protected \$beforeFind = [];
42	protected \$afterFind = [];
43	protected \$beforeDelete = [];
44	protected \$afterDelete = [];
45	}

Tabel 7. Source Code BukuModel.php

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class UserModel extends Model{
8	protected \$table = 'user';
9	protected \$primaryKey = 'id';
10	protected \$useAutoIncrement = true;
11	protected \$returnType = 'array';
12	protected \$useSoftDeletes = false;
13	protected \$protectFields = true;

14	protected \$allowedFields = ['username', 'email', 'password', 'created_at', 'updated_at'];
15	
16	public function getUserByEmailOrUsername(\$identifier)
17	{
18	return \$this->where('email', \$identifier)
19	->orWhere('username', \$identifier)
20	->first();
21	}
22	
23	protected bool \$allowEmptyInserts = false;
24	protected bool \$updateOnlyChanged = true;
25	
26	protected array \$casts = [];
27	protected array \$castHandlers = [];
28	
29	// Dates
30	protected \$useTimestamps = true;
31	protected \$dateFormat = 'datetime';
32	protected \$createdField = 'created_at';
33	protected \$updatedField = 'updated_at';
34	protected \$deletedField = 'deleted_at';
35	
36	// Validation
37	protected \$validationRules = [];
38	protected \$validationMessages = [];
39	protected \$skipValidation = false;
40	protected \$cleanValidationRules = true;
41	
42	// Callbacks
43	protected \$allowCallbacks = true;
44	protected \$beforeInsert = [];
45	protected \$afterInsert = [];
46	protected \$beforeUpdate = [];
47	protected \$afterUpdate = [];
48	protected \$beforeFind = [];
49	protected \$afterFind = [];
50	protected \$beforeDelete = [];
51	protected \$afterDelete = [];
52	}

Tabel 8. Source Code UserModel.php

1	<!DOCTYPE html>
2	<html>
3	<head>
4	<title>Form Buku</title>
5	<style>

6	body {	
7	margin: 0;	
8	padding: 0;	
9	display: flex;	
10	justify-content: center;	
11	align-items: center;	
12	height: 100vh;	
13	background: linear-gradient(to bottom	
	left, #DAF0F6, #DEF0FF, #C7DEF8, #85B7F1, #73AFF6);	
14	font-family: Arial, sans-serif;	
15	}	
16	.container {	
17	width: 40%;	
18	background-color: white;	
19	border-radius: 12px;	
20	box-shadow: 0 2px 4px rgba(0, 0, 0,	
	0.1);	
21	overflow: hidden;	
22	padding: 20px;	
23	}	
24	.form-header {	
25	padding: 20px;	
26	background-color: #F7F7F7;	
27	border-bottom: 1px solid #E0E0E0;	
28	text-align: center;	
29	}	
30	.form-header h3 {	
31	margin: 0;	
32	font-size: 25px;	
33	color: black;	
34	}	
35	.form-container {	
36	padding: 20px;	
37	}	
38	form label {	
39	margin-bottom: 10px;	
40	display: block;	
41	font-size: 14px;	
42	font-weight: bold;	
43	}	
44	form input[type="text"], form	
	input[type="number"] {	
45	width: 100%;	
46	margin-bottom: 20px;	
47	padding: 10px;	
48	border: 1px solid #CCCCCC;	
49	border-radius: 5px;	

```

50         box-sizing: border-box;
51     }
52     .button-container {
53         text-align: center;
54     }
55     .button-container input[type="submit"] {
56         width: 100%;
57         padding: 10px;
58         background-color: #73AFF6;
59         border: none;
60         border-radius: 5px;
61         color: white;
62         font-size: 16px;
63     }
64     .button-container
input[type="submit"]:hover {
65         background-color: #5D9BE1;
66     }
67     .error {
68         color: red;
69         font-size: 12px;
70         margin-top: -15px;
71         margin-bottom: 10px;
72     }
73     </style>
74     </head>
75     <body>
76         <div class="container">
77             <div class="form-header">
78                 <h3>Tambah Data Buku</h3>
79             </div>
80
81             <div class="form-container">
82                 <?php if (isset($validation)): ?>
83                     <div class="error">
84                         <?= $validation->listErrors()
85                         ?>
86                     </div>
87                 <?php endif; ?>
88
89                 <form      method="post"      action="<?=
base_url('buku/store') ?>" autocomplete="off">
90                     <?= csrf_field() ?>
91
92                     <label for="judul">Judul</label>
93                     <input  type="text"    name="judul"
id="judul" value="<?= set_value('judul') ?>" required>

```

```

93         <?php if (isset($validation) &&
$validation->hasError('judul')): ?>
94             <div class="error"><?=
$validation->getError('judul') ?></div>
95         <?php endif; ?>
96
97         <label
for="penulis">Penulis</label>
98         <input type="text" name="penulis"
id="penulis" value="<?= set_value('penulis') ?>"
required>
99         <?php if (isset($validation) &&
$validation->hasError('penulis')): ?>
100             <div class="error"><?=
$validation->getError('penulis') ?></div>
101         <?php endif; ?>
102
103         <label
for="penerbit">Penerbit</label>
104         <input type="text" name="penerbit"
id="penerbit" value="<?= set_value('penerbit') ?>"
required>
105         <?php if (isset($validation) &&
$validation->hasError('penerbit')): ?>
106             <div class="error"><?=
$validation->getError('penerbit') ?></div>
107         <?php endif; ?>
108
109         <label for="tahun_terbit">Tahun
Terbit</label>
110         <input type="number"
name="tahun_terbit" id="tahun_terbit" value="<?=
set_value('tahun_terbit') ?>" required>
111         <?php if (isset($validation) &&
$validation->hasError('tahun_terbit')): ?>
112             <div class="error"><?=
$validation->getError('tahun_terbit') ?></div>
113         <?php endif; ?>
114
115         <div class="button-container">
116             <input type="submit"
name="submit" value="Tambah">
117         </div>
118     </form>
119 </div>
120 </div>
121 </body>

```

122	</html>
-----	---------

Tabel 9. Source Code CreateView.php

1	<!DOCTYPE html>
2	<html>
3	<head>
4	<title>Edit Book</title>
5	<style>
6	body {
7	margin: 0;
8	padding: 0;
9	display: flex;
10	justify-content: center;
11	align-items: center;
12	height: 100vh;
13	background: linear-gradient(to bottom
14	left, #DAF0F6, #DEF0FF, #C7DEF8, #85B7F1, #73AFF6);
15	font-family: Arial, sans-serif;
16	}
17	.container {
18	width: 40%;
19	background-color: white;
20	border-radius: 12px;
21	box-shadow: 0 2px 4px rgba(0, 0, 0,
22	0.1);
23	overflow: hidden;
24	padding: 20px;
25	}
26	.form-header {
27	padding: 20px;
28	background-color: #F7F7F7;
29	border-bottom: 1px solid #E0E0E0;
30	text-align: center;
31	}
32	.form-header h3 {
33	margin: 0;
34	font-size: 25px;
35	color: black;
36	}
37	.form-container {
38	padding: 20px;
39	}
40	form label {
41	margin-bottom: 10px;
42	display: block;
43	font-size: 14px;
	font-weight: bold;
	}

44	<pre> form input[type="text"], form input[type="number"] { width: 100%; margin-bottom: 20px; padding: 10px; border: 1px solid #CCCCCC; border-radius: 5px; box-sizing: border-box; } .button-container { text-align: center; } .button-container input[type="submit"] { width: 100%; padding: 10px; background-color: #73AFF6; border: none; border-radius: 5px; color: white; font-size: 16px; } .button-container input[type="submit"]:hover { background-color: #5D9BE1; } .error { color: red; font-size: 12px; margin-top: -15px; margin-bottom: 10px; } </style> </head> <body> <div class="container"> <div class="form-header"> <h3>Edit Data Buku</h3> </div> <div class="form-container"> <form action="/buku/<?= \$book['id'] ?>/update" method="post"> <?= csrf_field() ?> <label>Judul</label> <input type="text" name="judul" value="<?= set_value('judul', \$book['judul']) ?>"> </pre>
----	--

87	<code><?php if (isset(\$validation) &&</code>
88	<code>\$validation->hasError('judul')): ?></code>
89	<code><div class="error"><?=</code>
90	<code>\$validation->getError('judul') ?></div></code>
91	<code><?php endif; ?></code>
92	<code><label>Penulis</label></code>
93	<code><input type="text" name="penulis"</code>
94	<code>value="<?= set_value('penulis', \$book['penulis']) ?>"></code>
95	<code><?php if (isset(\$validation) &&</code>
96	<code>\$validation->hasError('penulis')): ?></code>
97	<code><div class="error"><?=</code>
98	<code>\$validation->getError('penulis') ?></div></code>
99	<code><?php endif; ?></code>
100	<code><label>Penerbit</label></code>
101	<code><input type="text" name="penerbit"</code>
102	<code>value="<?= set_value('penerbit', \$book['penerbit'])</code>
103	<code>?>"></code>
104	<code><?php if (isset(\$validation) &&</code>
105	<code>\$validation->hasError('penerbit')): ?></code>
106	<code><div class="error"><?=</code>
107	<code>\$validation->getError('penerbit') ?></div></code>
108	<code><?php endif; ?></code>
109	<code><label>Tahun Terbit</label></code>
110	<code><input type="number"</code>
111	<code>name="tahun_terbit" value="<?=</code>
112	<code>set_value('tahun_terbit', \$book['tahun_terbit']) ?>"></code>
113	<code><?php if (isset(\$validation) &&</code>
114	<code>\$validation->hasError('tahun_terbit')): ?></code>
115	<code><div class="error"><?=</code>
116	<code>\$validation->getError('tahun_terbit') ?></div></code>
	<code><?php endif; ?></code>
	<code><div class="button-container"></code>
	<code><input type="submit"</code>
	<code>value="Update"></code>
	<code></div></code>
	<code></form></code>
	<code></div></code>
	<code></div></code>
	<code></body></code>
	<code></html></code>

Tabel 10. Source Code EditView.php

1	<code><!DOCTYPE html></code>
2	<code><html></code>

3	<head>
4	<title>Data Buku</title>
5	<style>
6	body {
7	margin: 0;
8	padding: 0;
9	display: flex;
10	flex-direction: column;
11	align-items: center;
12	background: linear-gradient(to bottom
13	left, #DAF0F6, #DEF0FF, #C7DEF8, #85B7F1, #73AFF6);
14	background-size: cover;
15	background-position: center;
16	font-family: Arial, sans-serif;
17	min-height: 100vh;
18	}
19	.navbar {
20	width: 100%;
21	display: flex;
22	justify-content: flex-end;
23	padding: 10px 20px 10px 0;
24	background-color: white;
25	border-radius: 5px;
26	box-shadow: 0 2px 4px rgba(0, 0, 0,
27	0.1);
28	}
29	.navbar a {
30	color: #73AFF6;
31	font-weight: bold;
32	text-decoration: underline;
33	}
34	.container {
35	width: 90%;
36	max-width: 1200px;
37	background-color: #FFFFFF;
38	border-radius: 10px;
39	box-shadow: 0 2px 10px rgba(0, 0, 0,
40	0.1);
41	margin: 20px 0;
42	padding: 20px;
43	}
44	.table-header {
45	padding: 10px 20px;
46	display: flex;
	justify-content: space-between;
	align-items: center;
	background-color: #F7F7F7;


```
47         border-bottom: 1px solid #E0E0E0;
48     }
49     .table-header h2 {
50         margin: 0;
51         font-size: 25px;
52         color: black;
53     }
54     .table-header .add-button {
55         padding: 5px 10px;
56         background-color: #73AFF6;
57         color: black;
58         border: none;
59         border-radius: 5px;
60         font-size: 14px;
61         text-decoration: none;
62     }
63     table {
64         width: 100%;
65         border-collapse: collapse;
66         margin-top: 20px;
67     }
68     table thead {
69         background-color: #F7F7F7;
70     }
71     table th, table td {
72         padding: 10px;
73         text-align: left;
74         border-right: 1px solid white;
75         border-left: 1px solid white;
76         border-bottom: 1px solid white;
77         font-size: 14px;
78     }
79     table th {
80         background-color: #E9ECEF;
81         font-weight: bold;
82     }
83     table tbody tr:nth-child(even) {
84         background-color: #F2F2F2;
85     }
86     .action-buttons {
87         display: flex;
88         gap: 10px;
89     }
90     .action-buttons a {
91         text-decoration: none;
92         color: #007BFF;
93         font-weight: bold;
```

```

94         }
95         .action-buttons form {
96             display: inline;
97         }
98         .action-buttons form button {
99             background: none;
100             border: none;
101             color: #007BFF;
102             font-weight: bold;
103             cursor: pointer;
104         }
105     </style>
106 </head>
107 <body>
108     <div class="navbar">
109         <a href="/logout">Logout</a>
110     </div>
111
112     <div class="container">
113         <div class="table-header">
114             <h2>Data Buku</h2>
115             <a href="{% base_url('buku/create') %}" class="add-button">Add New</a>
116         </div>
117         <table>
118             <thead>
119                 <tr>
120                     <th>ID</th>
121                     <th>Judul</th>
122                     <th>Penulis</th>
123                     <th>Penerbit</th>
124                     <th>Tahun Terbit</th>
125                     <th>Aksi</th>
126                 </tr>
127             </thead>
128
129             <tbody>
130                 <?php if (!empty($bukus)) : ?>
131                     <?php foreach ($bukus as $book)
132                         : ?>
133                         <tr>
134                             <td>{<%= $book['id'] %></td>
135                             <td>{<%= $book['judul'] %></td>
136                             <td>{<%= $book['penulis'] %></td>

```

```

136         <td><?=
    $book['penerbit'] ?></td>
137         <td><?=
    $book['tahun_terbit'] ?></td>
138         <td      class="action-
    buttons">
139             <a      href="<?=
    base_url("buku/{$book['id']}/edit") ?>">Edit</a>
140             <form method="post"
    action="<?= base_url("buku/{$book['id']}/delete") ?>">
141                 <?=
    csrf_field() ?>
142
143             <button
    type="submit" name="delete">Delete</button>
144             </form>
145         </td>
146     </tr>
147     <?php endforeach; ?>
148     <?php else : ?>
149     <tr>
150         <td colspan="6">No records
    found</td>
151     </tr>
152     <?php endif; ?>
153 </tbody>
154 </table>
155 </div>
156 </body>
157 </html>

```

Tabel 11. Source Code IndexView.php

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Login</title>
5          <style>
6              body {
7                  margin: 0;
8                  padding: 0;
9                  display: flex;
10                 justify-content: center;
11                 align-items: center;
12                 height: 100vh;
13                 background: linear-gradient(to bottom
    left, #DAF0F6, #DEF0FF, #C7DEF8, #85B7F1, #73AFF6);
14                 font-family: Arial, sans-serif;
15             }

```

16	.container {	
17	width: 30%;	
18	background-color: white;	
19	border-radius: 12px;	
20	box-shadow: 0 2px 4px rgba(0, 0, 0,	
	0.1);	
21	overflow: hidden;	
22	}	
23	.form-header {	
24	padding: 20px;	
25	background-color: #F7F7F7;	
26	border-bottom: 1px solid #E0E0E0;	
27	text-align: center;	
28	}	
29	.form-header h3 {	
30	margin: 0;	
31	font-size: 25px;	
32	color: black;	
33	}	
34	.form-container {	
35	padding: 20px;	
36	}	
37	form label {	
38	margin-bottom: 10px;	
39	display: block;	
40	font-size: 14px;	
41	font-weight: bold;	
42	}	
43	form input[type="text"],	form
	input[type="password"] {	
44	width: 100%;	
45	margin-bottom: 20px;	
46	padding: 10px;	
47	border: 1px solid #CCCCCC;	
48	border-radius: 5px;	
49	box-sizing: border-box;	
50	}	
51	.button-container {	
52	text-align: center;	
53	}	
54	.button-container input[type="submit"] {	
55	width: 100%;	
56	padding: 10px;	
57	background-color: #73AFF6;	
58	border: none;	
59	border-radius: 5px;	
60	color: white;	

61	font-size: 16px;
62	}
63	.button-container
	input[type="submit"]:hover {
64	background-color: #5D9BE1;
65	}
66	</style>
67	</head>
68	<body>
69	<div class="container">
70	<div class="form-header">
71	<h3>Login</h3>
72	</div>
73	
74	<div class="form-container">
75	<form method="post" action="/login"
	autocomplete="off">
76	<?= csrf_field() ?>
77	
78	<label
	for="username">Username</label>
79	<input type="text" name="username"
	id="username" required>
80	
81	<label
	for="password">Password</label>
82	<input type="password"
	name="password" id="password" required>
83	
84	<div class="button-container">
85	<input type="submit"
	name="login" value="Login">
86	</div>
87	</form>
88	</div>
89	</div>
90	</body>
91	</html>

Tabel 12. Source Code LoginView.php

B. Output Program

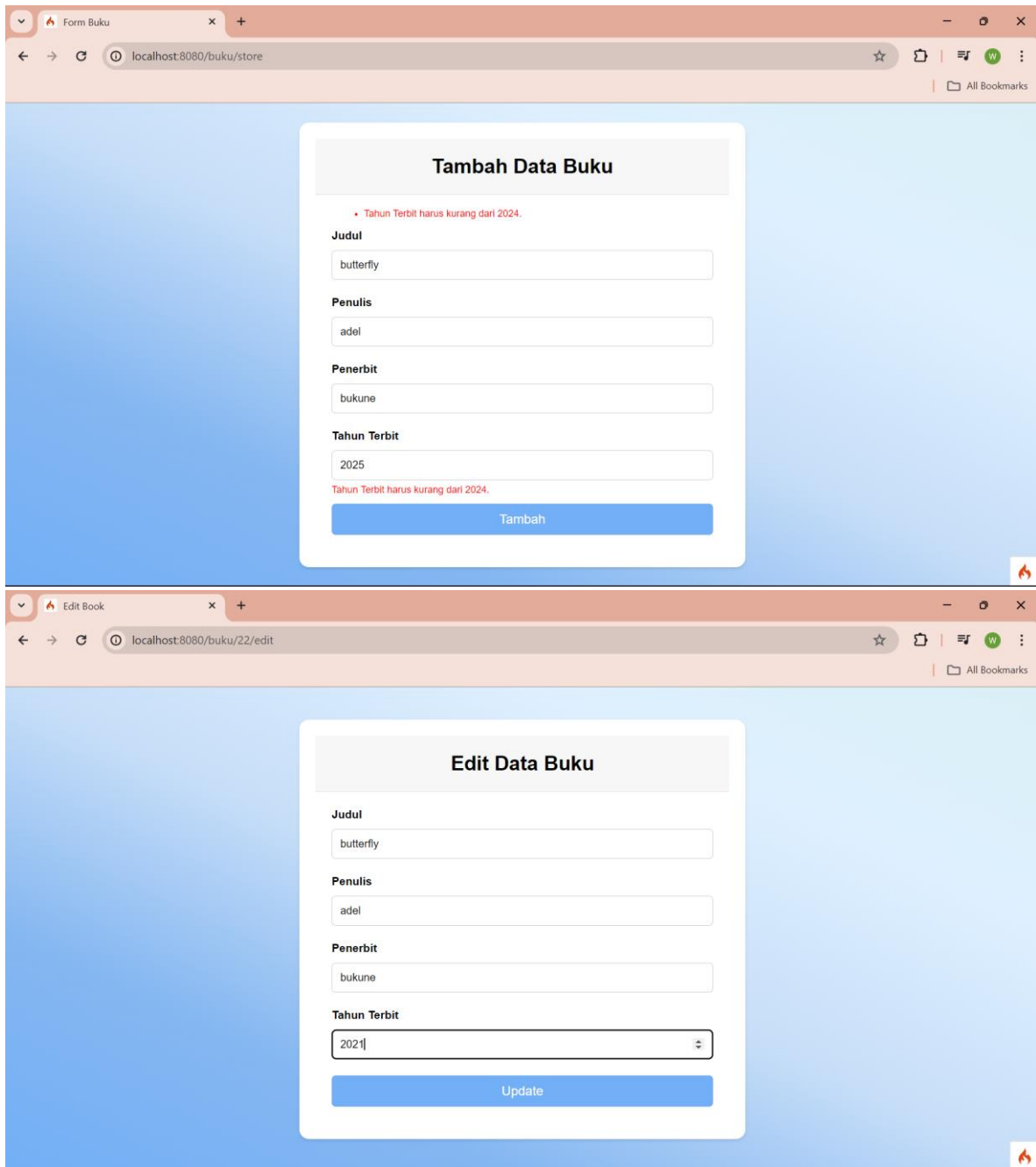
The image displays two screenshots of a web application running on a local host.

The top screenshot shows the **Login** page. The browser address bar indicates the URL is `localhost:8080/index.php/login`. The page features a central login form with the following fields and elements:

- Username:** A text input field containing the value `wnda`.
- Password:** A password input field with masked characters (dots).
- Login Button:** A blue button labeled `Login`.

The bottom screenshot shows the **Data Buku** page. The browser address bar indicates the URL is `localhost:8080/index.php/buku`. The page includes a **Logout** link in the top right corner and an **Add New** button. Below these is a table displaying book data:

ID	Judul	Penulis	Penerbit	Tahun Terbit	Aksi
1	terpikat	sasa	bukune	2022	Edit Delete



Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

- **Routes.php**

namespace Config digunakan untuk mendeklarasikan bahwa kode ini berada dalam namespace Config. `$routes = Services::routes()` ; adalah inisialisasi layanan rute dari CodeIgniter, yang memungkinkan kita untuk mendefinisikan berbagai rute.

- `setDefaultNamespace('App\Controllers')`: Menetapkan namespace default untuk controller, yaitu `App\Controllers`.
- `setDefaultController('Home')`: Menentukan controller default yang akan dipanggil jika tidak ada controller yang spesifik dalam URL, yaitu `Home`.
- `setDefaultMethod('index')`: Menentukan metode default yang akan dipanggil dalam controller, yaitu `index`.
- `setTranslateURIDashes(false)`: Menentukan apakah tanda hubung dalam URI harus diterjemahkan menjadi garis bawah dalam nama metode.
- `set404Override()`: Menentukan apakah akan ada penanganan khusus untuk halaman 404.
- `setAutoRoute(false)`: Menonaktifkan rute otomatis untuk alasan keamanan dan kontrol lebih ketat atas rute yang tersedia.

Berbagai rute didefinisikan menggunakan method `$routes->get()`, `$routes->post()`, dan lainnya. Setiap rute memiliki URI, target controller, dan metode, serta filter `auth` yang memastikan hanya pengguna yang terotentikasi yang bisa mengakses rute tersebut.

- `get('/', 'Home::index', ['filter' => 'auth'])`: Menetapkan rute untuk halaman utama yang akan memanggil metode `index` dari controller `Home` dengan filter `auth`.
- `get('/buku', 'Buku::index', ['filter' => 'auth'])`: Menetapkan rute untuk halaman daftar buku yang memanggil metode `index` dari controller `Buku` dengan filter `auth`.
- `get('/buku/create', 'Buku::create', ['filter' => 'auth'])`: Rute untuk halaman pembuatan buku baru.
- `post('/buku/store', 'Buku::store', ['filter' => 'auth'])`: Rute untuk menyimpan data buku baru.
- `get('/buku/(:num)/edit', 'Buku::edit/$1', ['filter' => 'auth'])`: Rute untuk mengedit buku dengan ID tertentu.
- `post('/buku/(:num)/update', 'Buku::update/$1', ['filter' => 'auth'])`: Rute untuk memperbarui data buku dengan ID tertentu.
- `post('/buku/(:num)/delete', 'Buku::delete/$1', ['filter' => 'auth'])`: Rute untuk menghapus buku dengan ID tertentu.
- `get('/dashboard', 'Home::dashboard', ['filter' => 'auth'])`: Rute untuk halaman dashboard.
- `get('/login', 'Auth::login')`: Rute untuk halaman login tanpa filter `auth` karena perlu diakses oleh pengguna yang belum login.
- `post('/login', 'Auth::loginAuth')`: Rute untuk proses autentikasi login.
- `get('/logout', 'Auth::logout')`: Rute untuk logout pengguna.

- **Auth.php**

`namespace App\Controllers` digunakan untuk mendeklarasikan bahwa kelas `Auth` berada dalam folder `Controllers` di aplikasi. `use App\Models\UserModel;` dan `use CodeIgniter\Controller;` digunakan untuk mengimpor model `UserModel` dan kelas `Controller` dari `CodeIgniter`.

Fungsi `store` bertanggung jawab untuk menyimpan data pengguna baru ke database.

- `UserModel` diinisialisasi untuk berinteraksi dengan tabel pengguna.
- Data pengguna diambil dari permintaan (`request`) menggunakan `$this->request->getVar('fieldname')`.
- Kata sandi di-hash menggunakan `password_hash` dengan algoritma default.
- Data disimpan ke database menggunakan metode `save` dari `UserModel`.
- Setelah penyimpanan berhasil, pengguna diarahkan ke halaman login dengan `redirect()->to('/login')`.

Fungsi `login` bertanggung jawab untuk menampilkan halaman login dan memproses data login.

- `helper(['form'])` memuat helper form untuk memudahkan pengelolaan form.
- Jika metode permintaan adalah `POST`, maka fungsi `loginAuth` dipanggil untuk memproses login.
- Jika tidak, halaman login (`LoginView`) ditampilkan.

Fungsi `loginAuth` bertanggung jawab untuk memproses autentikasi login pengguna.

- Sesi diinisialisasi dengan `session()`.
- `UserModel` diinisialisasi untuk berinteraksi dengan tabel pengguna.
- Identifikasi pengguna (bisa `username` atau `email`) dan kata sandi diambil dari permintaan.
- `getUserByEmailOrUsername` adalah metode model yang digunakan untuk mendapatkan data pengguna berdasarkan `username` atau `email`.
- Jika data pengguna ditemukan, kata sandi yang di-hash diverifikasi menggunakan `password_verify`.
- Jika verifikasi kata sandi berhasil, data sesi pengguna diatur dan pengguna diarahkan ke halaman buku (`/buku`).
- Jika kata sandi salah, pesan kesalahan disimpan dalam sesi (`setFlashdata`) dan pengguna diarahkan kembali ke halaman login.
- Jika `username` atau `email` tidak ditemukan, pesan kesalahan juga disimpan dalam sesi dan pengguna diarahkan kembali ke halaman login.

Fungsi `logout` bertanggung jawab untuk mengakhiri sesi pengguna.

- Sesi diinisialisasi dengan `session()`.
- Sesi dihancurkan menggunakan `$session->destroy()`.
- Pengguna diarahkan kembali ke halaman login dengan `redirect()->to('/login')`.

- **Buku.php**

`namespace App\Controllers` digunakan untuk mendeklarasikan bahwa kelas `Buku` berada dalam folder `Controllers` di aplikasi. `use App\Models\BukuModel;` dan `use CodeIgniter\Controller;` digunakan untuk mengimpor model `BukuModel` dan kelas `Controller` dari `CodeIgniter`.

Fungsi `__construct` memuat helper `form` dan `url` untuk mempermudah pengelolaan form dan URL dalam metode-metode lain di controller ini.

Fungsi `index` bertanggung jawab untuk menampilkan daftar buku dengan opsi pencarian dan pengurutan.

- `BukuModel` diinisialisasi untuk berinteraksi dengan tabel buku.
- `search`, `sort`, dan `sort_order` diambil dari parameter GET.
- Query awal disiapkan dari `BukuModel`.
- Jika ada parameter `search`, query diperbarui untuk mencakup kondisi `like` untuk kolom `judul`, `penulis`, `penerbit`, dan `tahun_terbit`.
- Jika ada parameter `sort` dan nilai `sort` valid, query diperbarui untuk mengurutkan berdasarkan kolom yang dipilih dengan urutan yang sesuai.
- Data buku diambil dengan `findAll()` dan disimpan dalam array `$data`.
- Halaman `IndexView` ditampilkan dengan data yang telah dikumpulkan.

Fungsi `create` bertanggung jawab untuk menampilkan form pembuatan buku baru.

- Menampilkan halaman `CreateView`.

Fungsi `store` bertanggung jawab untuk memvalidasi dan menyimpan data buku baru.

- Validasi diatur untuk memastikan semua kolom wajib (`judul`, `penulis`, `penerbit`, `tahun_terbit`) diisi dan sesuai dengan aturan.
- Jika validasi gagal, form pembuatan buku ditampilkan kembali dengan pesan kesalahan.
- Jika validasi berhasil, data buku disimpan menggunakan `save` dari `BukuModel`.
- Pengguna diarahkan ke halaman daftar buku dengan `redirect()->to('/buku')`.

Fungsi `edit` bertanggung jawab untuk menampilkan form pengeditan buku dengan data buku yang ada.

- Data buku diambil berdasarkan ID dengan `find`.
- Halaman `EditView` ditampilkan dengan data buku.

Fungsi `update` bertanggung jawab untuk memvalidasi dan memperbarui data buku yang ada.

- Validasi diatur sama seperti pada fungsi `store`.
- Jika validasi gagal, form pengeditan buku ditampilkan kembali dengan pesan kesalahan dan data buku yang ada.
- Jika validasi berhasil, data buku diperbarui menggunakan `update` dari `BukuModel`.

- Pengguna diarahkan ke halaman daftar buku dengan `redirect()->to('/buku')`.

Fungsi `delete` bertanggung jawab untuk menghapus data buku berdasarkan ID.

- Data buku dihapus menggunakan `delete` dari `BukuModel`.
- Pengguna diarahkan ke halaman daftar buku dengan `redirect()->to('/buku')`.

• **2024-06-10-064318_CreateDbModul6.php**

`namespace App\Database\Migrations` digunakan untuk mendeklarasikan bahwa kelas `CreateDbModul6` berada dalam folder `Migrations` di aplikasi. `use CodeIgniter\Database\Migration;` digunakan untuk mengimpor kelas `Migration` dari `CodeIgniter`.

Fungsi `up` bertanggung jawab untuk membuat tabel dalam database.

- `addField` digunakan untuk menambahkan kolom ke tabel.
- `addKey` digunakan untuk menambahkan kunci utama (primary key) ke tabel.
- `createTable` digunakan untuk membuat tabel dengan nama yang diberikan.

Fungsi `down` bertanggung jawab untuk menghapus tabel dalam database jika diperlukan.

- `dropTable('user')` menghapus tabel `user`.
- `dropTable('buku')` menghapus tabel `buku`.

• **UserSeeder.php**

`namespace App\Database\Seeds` digunakan untuk mendeklarasikan bahwa kelas `UserSeeder` berada dalam folder `Seeds` di aplikasi. `use CodeIgniter\Database\Seeder;` digunakan untuk mengimpor kelas `Seeder` dari `CodeIgniter`.

Fungsi `run` adalah fungsi utama yang akan dijalankan ketika seeder dieksekusi.

- Array `$data` digunakan untuk menyimpan data pengguna yang akan dimasukkan ke dalam tabel `user`.
 - `username` diatur dengan nilai `'wnda'`.
 - `email` diatur dengan nilai `'pengguna123@example.com'`.
 - `password` diatur dengan nilai hash dari string `'password'` menggunakan fungsi `password_hash` dengan algoritma default (`PASSWORD_DEFAULT`).
- `$this->db->table('user')->insert($data)` digunakan untuk memasukkan data ke dalam tabel `user`.
 - `$this->db->table('user')` memilih tabel `user`.
 - `insert($data)` memasukkan data yang ada di array `$data` ke dalam tabel yang dipilih.

- **AuthFilter.php**

`namespace App\Filters` digunakan untuk mendeklarasikan bahwa kelas `AuthFilter` berada dalam folder `Filters` di aplikasi. `use CodeIgniter\Filters\FilterInterface;` `use CodeIgniter\HTTP\RequestInterface;` dan `use CodeIgniter\HTTP\ResponseInterface;` digunakan untuk mengimpor antarmuka dan kelas yang diperlukan dari `CodeIgniter`.

Fungsi `before` dijalankan sebelum permintaan (request) diproses lebih lanjut.

- `if (!session()->get('logged_in')) { ... }` memeriksa apakah sesi `logged_in` tidak diatur atau bernilai `false`.
 - Jika pengguna belum login, maka pengguna akan diarahkan ke halaman login dengan `return redirect()->to('/login')->with('msg', 'Login terlebih dahulu!');`
 - `redirect()->to('/login')` mengarahkan pengguna ke halaman login.
 - `->with('msg', 'Login terlebih dahulu!')` menambahkan pesan flashdata yang akan ditampilkan di halaman login, memberi tahu pengguna untuk login terlebih dahulu.

Fungsi `after` dijalankan setelah permintaan diproses dan sebelum respons dikirim ke klien.

- Pada implementasi ini, fungsi `after` tidak melakukan apa-apa, sebagaimana dinyatakan dengan komentar `// Do nothing`.

- **BukuModel.php**

`namespace App\Models` digunakan untuk mendeklarasikan bahwa kelas `BukuModel` berada dalam folder `Models` di aplikasi. `use CodeIgniter\Model;` digunakan untuk mengimpor kelas `Model` dari `CodeIgniter`.

- `$table`: Menentukan nama tabel yang digunakan, yaitu `buku`.
- `$primaryKey`: Menentukan kolom kunci utama dari tabel, yaitu `id`.
- `$useAutoIncrement`: Menentukan bahwa kolom `id` akan menggunakan `auto increment`.
- `$returnType`: Menentukan tipe data yang dikembalikan sebagai `array`.
- `$useSoftDeletes`: Menentukan apakah `soft deletes` digunakan atau tidak. Pada kasus ini, nilainya `false` berarti tidak menggunakan `soft deletes`.
- `$protectFields`: Menentukan apakah hanya kolom yang diizinkan (`allowedFields`) yang dapat dimanipulasi. Pada kasus ini, nilainya `true`.
- `$allowedFields`: Menentukan kolom-kolom yang diizinkan untuk diisi, yaitu `judul`, `penulis`, `penerbit`, dan `tahun_terbit`.

- `$allowEmptyInserts`: Menentukan apakah insert tanpa data diperbolehkan. Nilainya `false` berarti tidak diperbolehkan.
- `$updateOnlyChanged`: Menentukan apakah hanya kolom yang berubah saja yang akan diperbarui. Nilainya `true`.
- `$casts` dan `$castHandlers`: Digunakan untuk casting data, namun pada kasus ini tidak ada casting yang didefinisikan.
- `$useTimestamps`: Menentukan apakah timestamps digunakan atau tidak. Pada kasus ini, nilainya `false`.
- `$dateFormat`: Menentukan format tanggal yang digunakan. Nilainya `datetime`.
- `$createdField`, `updatedField`, `deletedField`: Menentukan nama kolom untuk timestamps, namun tidak digunakan dalam kasus ini karena `$useTimestamps` bernilai `false`.
- `$validationRules`: Menyimpan aturan validasi, namun tidak ada aturan yang didefinisikan pada kasus ini.
- `$validationMessages`: Menyimpan pesan validasi khusus, namun tidak ada pesan yang didefinisikan.
- `$skipValidation`: Menentukan apakah validasi akan dilewati. Nilainya `false` berarti validasi tidak dilewati.
- `$cleanValidationRules`: Menentukan apakah aturan validasi akan dibersihkan. Nilainya `true`.
- `$allowCallbacks`: Menentukan apakah callback diizinkan. Nilainya `true`.
- `$beforeInsert`, `$afterInsert`, `$beforeUpdate`, `$afterUpdate`, `$beforeFind`, `$afterFind`, `$beforeDelete`, `$afterDelete`: Menyimpan nama-nama metode callback yang akan dipanggil pada waktu yang sesuai, namun pada kasus ini tidak ada callback yang didefinisikan.

- **UserModel.php**

Kelas `UserModel` berada dalam namespace `App\Models` dan meng-extend kelas `Model` dari `CodeIgniter`, yang menyediakan fitur-fitur untuk berinteraksi dengan database.

- `$table`: Menentukan nama tabel yang digunakan, yaitu `user`.
- `$primaryKey`: Menentukan kolom kunci utama dari tabel, yaitu `id`.
- `$useAutoIncrement`: Menentukan bahwa kolom `id` akan menggunakan `auto increment`.
- `$returnType`: Menentukan tipe data yang dikembalikan sebagai array.
- `$useSoftDeletes`: Menentukan apakah `soft deletes` digunakan atau tidak. Pada kasus ini, nilainya `false` berarti tidak menggunakan `soft deletes`.
- `$protectFields`: Menentukan apakah hanya kolom yang diizinkan (`allowedFields`) yang dapat dimanipulasi. Pada kasus ini, nilainya `true`.
- `$allowedFields`: Menentukan kolom-kolom yang diizinkan untuk diisi, yaitu `username`, `email`, `password`, `created_at`, dan `updated_at`.

Metode `getUserByEmailOrUsername` digunakan untuk mengambil data pengguna berdasarkan email atau username yang diberikan sebagai parameter `$identifier`. Metode ini menggunakan query `where` untuk mencocokkan entri di kolom email atau username, dan mengembalikan hasil pertama yang ditemukan.

- `$allowEmptyInserts`: Menentukan apakah insert tanpa data diperbolehkan. Nilainya `false` berarti tidak diperbolehkan.
- `$updateOnlyChanged`: Menentukan apakah hanya kolom yang berubah saja yang akan diperbarui. Nilainya `true`.
- `$casts` dan `$castHandlers`: Digunakan untuk casting data, namun pada kasus ini tidak ada casting yang didefinisikan.
- `$useTimestamps`: Menentukan apakah timestamps digunakan atau tidak. Pada kasus ini, nilainya `true`.
- `$dateFormat`: Menentukan format tanggal yang digunakan, yaitu `'datetime'`.
- `$createdField` dan `$updatedField`: Menentukan nama kolom untuk timestamps `created_at` dan `updated_at`.
- `$deletedField`: Dalam kasus ini, nama kolom untuk soft deletes tidak didefinisikan karena `useSoftDeletes` bernilai `false`.

Properti seperti `$validationRules`, `$validationMessages`, `$skipValidation`, dan `$cleanValidationRules` digunakan untuk aturan validasi, pesan validasi, serta pengaturan validasi lainnya, namun dalam implementasi ini tidak ada aturan validasi yang ditentukan.

Callbacks seperti `$beforeInsert`, `$afterInsert`, `$beforeUpdate`, `$afterUpdate`, `$beforeFind`, `$afterFind`, `$beforeDelete`, dan `$afterDelete` dapat digunakan untuk menambahkan logika sebelum atau setelah operasi tertentu dieksekusi di database.

• **CreateView.php**

Dokumen HTML dimulai dengan deklarasi `<!DOCTYPE html>` untuk mendefinisikan tipe dokumen dan versi HTML yang digunakan. Elemen `<html>` menjadi pembungkus seluruh konten dokumen.

Bagian `<head>` berisi metadata dokumen seperti judul dan gaya. Tag `<title>` memberikan judul "Form Buku" pada halaman. Blok `<style>` di dalam `<head>` mendefinisikan berbagai gaya CSS yang digunakan untuk memformat tampilan form.

Kode CSS internal digunakan untuk mengatur tata letak dan tampilan visual dari halaman. Ini mencakup pengaturan seperti background gradient, warna teks, ukuran font, dan tata letak elemen-elemen seperti form dan tombol.

- `body`: Mengatur margin, padding, tampilan fleksibel, pusatkan konten secara horizontal dan vertikal, tinggi 100vh, latar belakang dengan gradien warna, dan font default Arial.

- `.container`: Kontainer utama dengan lebar 40%, latar belakang putih, border-radius 12px, bayangan, dan padding 20px.
- `.form-header`: Bagian header form dengan padding, latar belakang abu-abu terang, border bawah, dan teks tengah.
- `.form-header h3`: Mengatur margin 0, ukuran font 25px, dan warna teks hitam.
- `.form-container`: Memberikan padding 20px pada kontainer form.
- `form label`: Mengatur margin bawah, tampilan blok, ukuran font, dan berat font.
- `form input`: Mengatur lebar penuh, margin bawah, padding, border, border-radius, dan box-sizing.
- `.button-container`: Mengatur teks tengah.
- `.button-container input[type="submit"]`: Mengatur lebar penuh, padding, latar belakang biru, border none, border-radius, warna putih, ukuran font, dan gaya hover.
- `.error`: Mengatur warna merah, ukuran font, dan margin untuk pesan error.

Bagian `<body>` berisi elemen `<div class="container">` yang menjadi kontainer utama form. Di dalamnya terdapat dua div: `.form-header` untuk header dan `.form-container` untuk kontainer form.

- `<?php if (isset($validation)): ?>` Mengecek apakah variabel `$validation` ada, yang menunjukkan adanya validasi.
- `<div class="error"> <?= $validation->listErrors() ?> </div>` Menampilkan pesan error jika ada kesalahan validasi.

Formulir menggunakan metode POST dan diarahkan ke `<?= base_url('buku/store') ?>`. Fungsi `csrf_field()` menambahkan token CSRF untuk keamanan.

- Label dan Input: Setiap label dan input teks diberi atribut `name`, `id`, dan nilai default menggunakan fungsi `set_value()`.
- Validasi Input: Jika ada kesalahan, pesan error spesifik akan ditampilkan di bawah input terkait.

Tombol submit ditempatkan dalam `.button-container` untuk mengirimkan form. Tombol ini memiliki gaya yang konsisten dengan tema form.

• EditView.php

Dokumen HTML dimulai dengan deklarasi `<!DOCTYPE html>` untuk mendefinisikan tipe dokumen dan versi HTML yang digunakan. Elemen `<html>` menjadi pembungkus seluruh konten dokumen.

Bagian `<head>` berisi metadata dokumen seperti judul dan gaya. Tag `<title>` memberikan judul "Edit Book" pada halaman. Blok `<style>` di dalam `<head>` mendefinisikan berbagai gaya CSS yang digunakan untuk memformat tampilan form.

Kode CSS internal digunakan untuk mengatur tata letak dan tampilan visual dari halaman. Ini mencakup pengaturan seperti background gradient, warna teks, ukuran font, dan tata letak elemen-elemen seperti form dan tombol.

- `body`: Mengatur margin, padding, tampilan fleksibel, pusatkan konten secara horizontal dan vertikal, tinggi 100vh, latar belakang dengan gradien warna, dan font default Arial.
- `.container`: Kontainer utama dengan lebar 40%, latar belakang putih, border-radius 12px, bayangan, dan padding 20px.
- `.form-header`: Bagian header form dengan padding, latar belakang abu-abu terang, border bawah, dan teks tengah.
- `.form-header h3`: Mengatur margin 0, ukuran font 25px, dan warna teks hitam.
- `.form-container`: Memberikan padding 20px pada kontainer form.
- `form label`: Mengatur margin bawah, tampilan blok, ukuran font, dan berat font.
- `form input`: Mengatur lebar penuh, margin bawah, padding, border, border-radius, dan box-sizing.
- `.button-container`: Mengatur teks tengah.
- `.button-container input[type="submit"]`: Mengatur lebar penuh, padding, latar belakang biru, border none, border-radius, warna putih, ukuran font, dan gaya hover.
- `.error`: Mengatur warna merah, ukuran font, dan margin untuk pesan error.

Bagian `<body>` berisi elemen `<div class="container">` yang menjadi kontainer utama form. Di dalamnya terdapat dua div: `.form-header` untuk header dan `.form-container` untuk kontainer form.

Formulir menggunakan metode POST dan diarahkan ke `/buku/<?= $book['id'] ?>/update`. Fungsi `csrf_field()` menambahkan token CSRF untuk keamanan.

- **Label dan Input**: Setiap label dan input teks diberi atribut `name` dan nilai default menggunakan fungsi `set_value()` untuk mengisi input dengan data yang ada dalam variabel `$book`.
- **Validasi Input**: Jika ada kesalahan, pesan error spesifik akan ditampilkan di bawah input terkait.
- `<?php if (isset($validation) && $validation->hasError('judul')): ?>` Mengecek apakah variabel `$validation` ada dan apakah terdapat kesalahan pada input 'judul'.
- `<div class="error"><?= $validation->getError('judul') ?></div>` Menampilkan pesan error jika ada kesalahan validasi.

Tombol submit ditempatkan dalam `.button-container` untuk mengirimkan form. Tombol ini memiliki gaya yang konsisten dengan tema form.

- **IndexView.php**

Dokumen HTML dimulai dengan deklarasi `<!DOCTYPE html>` untuk mendefinisikan tipe dokumen dan versi HTML yang digunakan. Elemen `<html>` menjadi pembungkus seluruh konten dokumen.

Bagian `<head>` berisi metadata dokumen seperti judul dan gaya. Tag `<title>` memberikan judul "Data Buku" pada halaman. Blok `<style>` di dalam `<head>` mendefinisikan berbagai gaya CSS yang digunakan untuk memformat tampilan halaman.

Kode CSS internal digunakan untuk mengatur tata letak dan tampilan visual dari halaman. Ini mencakup pengaturan seperti background gradient, warna teks, ukuran font, dan tata letak elemen-elemen seperti form dan tombol.

- `body`: Mengatur margin, padding, tampilan fleksibel, arah kolom, pusatkan konten secara horizontal dan vertikal, latar belakang dengan gradien warna, ukuran latar belakang, posisi latar belakang, font default Arial, dan tinggi minimum 100vh.
- `.navbar`: Mengatur lebar penuh, tampilan fleksibel, pusatkan konten di ujung kanan, padding, latar belakang putih, border-radius 5px, dan bayangan.
- `.navbar a`: Mengatur warna teks biru, font tebal, dan dekorasi teks garis bawah.
- `.container`: Kontainer utama dengan lebar 90%, maksimal lebar 1200px, latar belakang putih, border-radius 10px, bayangan, margin vertikal 20px, dan padding 20px.
- `.table-header`: Mengatur padding, tampilan fleksibel, pusatkan konten secara horizontal, latar belakang abu-abu terang, dan border bawah.
- `.table-header h2`: Mengatur margin 0, ukuran font 25px, dan warna teks hitam.
- `.table-header .add-button`: Mengatur padding, latar belakang biru, warna teks hitam, border none, border-radius 5px, ukuran font, dan dekorasi teks none.
- `table`: Mengatur lebar penuh, border-collapse, dan margin atas 20px.
- `table thead`: Mengatur latar belakang abu-abu terang.
- `table th, table td`: Mengatur padding, teks rata kiri, border samping dan bawah, dan ukuran font.
- `table th`: Mengatur latar belakang abu-abu lebih gelap dan font tebal.
- `table tbody tr:nth-child(even)`: Mengatur latar belakang abu-abu terang.
- `.action-buttons`: Mengatur tampilan fleksibel dan jarak antar elemen.
- `.action-buttons a`: Mengatur dekorasi teks none, warna teks biru, dan font tebal.
- `.action-buttons form`: Mengatur tampilan inline.
- `.action-buttons form button`: Mengatur latar belakang none, border none, warna teks biru, font tebal, dan cursor pointer.

Bagian `<body>` berisi dua elemen utama:

- `.navbar`: Elemen ini berisi tautan logout yang ditempatkan di ujung kanan.
- `.container`: Kontainer utama yang berisi tabel data buku.
- `.table-header`: Bagian header tabel dengan judul "Data Buku" dan tombol "Add New" untuk menambahkan data buku baru.
- `table`: Elemen tabel yang menampilkan data buku.
 - `thead`: Bagian header tabel yang berisi judul kolom: ID, Judul, Penulis, Penerbit, Tahun Terbit, dan Aksi.
 - `tbody`: Bagian isi tabel yang menampilkan data buku. Jika tidak ada data, baris "No records found" akan ditampilkan.
- `<?php if (!empty($bukus)) : ?>`: Mengecek apakah variabel `$bukus` tidak kosong.
- `<?php foreach ($bukus as $book) : ?>`: Melakukan iterasi untuk setiap buku dalam variabel `$bukus` dan menampilkan data buku dalam baris tabel.
 - `<?= $book['id'] ?>`: Menampilkan ID buku.
 - `<?= $book['judul'] ?>`: Menampilkan judul buku.
 - `<?= $book['penulis'] ?>`: Menampilkan penulis buku.
 - `<?= $book['penerbit'] ?>`: Menampilkan penerbit buku.
 - `<?= $book['tahun_terbit'] ?>`: Menampilkan tahun terbit buku.
 - `<?= base_url("buku/{$book['id']}/edit") ?>`: Tautan untuk mengedit data buku.
 - `<?= base_url("buku/{$book['id']}/delete") ?>`: Formulir untuk menghapus data buku dengan metode POST dan token CSRF untuk keamanan.

• LoginView.php

Dokumen HTML dimulai dengan deklarasi `<!DOCTYPE html>` untuk mendefinisikan tipe dokumen dan versi HTML yang digunakan. Elemen `<html>` membungkus seluruh konten dokumen.

Bagian `<head>` berisi metadata dokumen seperti judul dan gaya. Tag `<title>` memberikan judul "Login" pada halaman. Blok `<style>` di dalam `<head>` mendefinisikan berbagai gaya CSS yang digunakan untuk memformat tampilan halaman.

Kode CSS internal digunakan untuk mengatur tata letak dan tampilan visual dari halaman. Ini mencakup pengaturan seperti background gradient, warna teks, ukuran font, dan tata letak elemen-elemen seperti form dan tombol.

- `body`: Mengatur margin, padding, tampilan fleksibel, pusatkan konten secara horizontal dan vertikal, latar belakang dengan gradien warna, font default Arial, dan tinggi penuh layar (100vh).
- `.container`: Kontainer utama dengan lebar 30%, latar belakang putih, border-radius 12px, bayangan lembut, dan overflow tersembunyi.
- `.form-header`: Bagian header formulir dengan padding 20px, latar belakang abu-abu terang (#F7F7F7), border-bottom, dan teks tengah.

- `.form-header h3`: Mengatur margin 0, ukuran font 25px, dan warna teks hitam.
- `.form-container`: Kontainer formulir dengan padding 20px.
- `form label`: Mengatur margin bawah, tampilan blok, ukuran font 14px, dan font tebal.
- `form input[type="text"], form input[type="password"]`: Mengatur lebar penuh, margin bawah 20px, padding 10px, border 1px solid #CCCCCC, border-radius 5px, dan box-sizing border-box.
- `.button-container`: Kontainer untuk tombol dengan teks tengah.
- `.button-container input[type="submit"]`: Tombol submit dengan lebar penuh, padding 10px, latar belakang biru (#73AFF6), border none, border-radius 5px, warna teks putih, dan ukuran font 16px.
- `.button-container input[type="submit"]:hover`: Mengatur latar belakang biru lebih gelap (#5D9BE1) saat dihover.
- `.container`: Kontainer utama yang berisi formulir login.
- `.form-header`: Bagian header formulir dengan judul "Login".
- `.form-container`: Kontainer utama untuk formulir login.
- `form`: Elemen formulir dengan metode POST dan aksi menuju /login.
 - `<?= csrf_field() ?>`: PHP untuk menambahkan token CSRF untuk keamanan.
 - `label for="username"`: Label untuk input username.
 - `input type="text" name="username" id="username"`: Input teks untuk username dengan atribut `required` agar tidak boleh kosong.
 - `label for="password"`: Label untuk input password.
 - `input type="password" name="password" id="password"`: Input password dengan atribut `required` agar tidak boleh kosong.
 - `.button-container`: Kontainer untuk tombol submit.
 - `input type="submit" name="login" value="Login"`: Tombol submit dengan nilai "Login".

LINK GITHUB

<https://github.com/wandakarliyanti/Praktikum-Pemrograman-Web-II/tree/main/Modul%206>