

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Aeron Tjitradi

NIM. 2210817310010

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN WEB II
MODUL 5

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 : Aeron Tjitradi
NIM : 2210817310010

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

Eka Setya Wijaya, S.Kom., 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.....	21
C. Pembahasan	23
TAUTAN GIT	28

DAFTAR GAMBAR

Gambar 1. Konseptual Database	Error! Bookmark not defined.
Gambar 2. Tampilan Index.php	21
Gambar 3. Tampilan Buku.php	21
Gambar 4. Tampilan FormBuku.php	21
Gambar 5. Tampilan Member.php	22
Gambar 6. Tampilan FormMember.php	22
Gambar 7. Tampilan Peminjaman.php.....	Error! Bookmark not defined.
Gambar 8. Tampilan FormPeminjaman.php	Error! Bookmark not defined.

DAFTAR TABEL

Tabel 1. Source Code index.php	6
Tabel 2. Source Code koneksi.php.....	7
Tabel 3. Source Code model.php	8
Tabel 4. Source Code member.php	11
Tabel 5. Source Code formmember.php	12
Tabel 6. Source Code buku.php	13
Tabel 7. Source Code formbuku.php	14
Tabel 8. Source Code peminjaman.php	14
Tabel 9. Source Code formpeminjaman.php.....	15

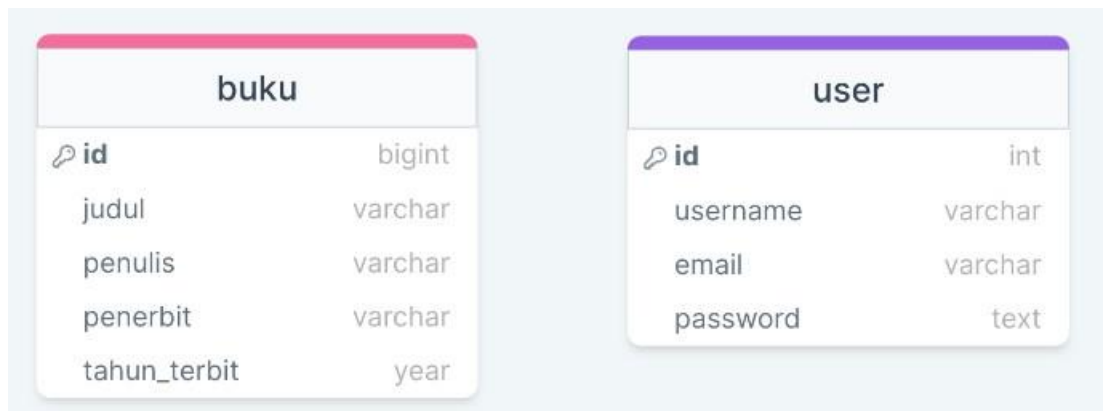
SOAL 1

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



Gambar 1. Soal Praktikum

A. Source Code

Tabel 1. Source Code Routes.php

Routes.php	
1	<?php
2	
3	use CodeIgniter\Router\RouteCollection;
4	
5	/**
6	* @var RouteCollection \$routes
7	*/
8	\$routes->get('/', 'Auth::login');
9	\$routes->get('/register', 'Auth::register');
10	\$routes->post('/auth/store', 'Auth::store');
11	\$routes->get('/login', 'Auth::login');
12	\$routes->post('/auth/loginAuth', 'Auth::loginAuth');
13	\$routes->get('/logout', 'Auth::logout');
14	
15	\$routes->get('/dashboard', 'Buku::index', ['filter' => 'auth']);
16	\$routes->get('/buku', 'Buku::index', ['filter' => 'auth']);

17	\$routes->get('/buku/create', 'Buku::create', ['filter' => 'auth']);
18	\$routes->post('/buku/store', 'Buku::store', ['filter' => 'auth']);
19	\$routes->get('/buku/edit/(:num)', 'Buku::edit/\$1', ['filter' => 'auth']);
20	\$routes->post('/buku/update/(:num)', 'Buku::update/\$1', ['filter' => 'auth']);
21	\$routes->get('/buku/delete/(:num)', 'Buku::delete/\$1', ['filter' => 'auth']);

Tabel 2. Source Code Auth.php

Auth.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	{
10	public function register()
11	{
12	return view('auth/register');
13	}
14	
15	public function store()
16	{
17	\$userModel = new UserModel();
18	\$data = [
19	'username' => \$this->request->getVar('username'),
20	'email' => \$this->request->getVar('email'),
21	'password' => password_hash(\$this->request->
22	getVar('password'), PASSWORD_DEFAULT),
23];
24	\$userModel->save(\$data);
25	return redirect()->to('/login');
26	}
27	public function login()
28	{
29	helper(['form']);
30	return view('auth/login');
31	}
32	
33	public function loginAuth()
34	{
35	\$session = session();
36	\$userModel = new UserModel();
37	\$identifier = \$this->request->getVar('identifier');
38	\$password = \$this->request->getVar('password');
39	\$data = \$userModel->getUserByEmailOrUsername(\$identifier);
40	
41	if (\$data) {
42	\$pass = \$data['password'];
43	\$authenticatePassword = password_verify(\$password, \$pass);
44	
45	if (\$authenticatePassword) {

46	\$ses_data = [
47	'id' => \$data['id'],
48	'username' => \$data['username'],
49	'logged_in' => TRUE
50];
51	\$session->set(\$ses_data);
52	return redirect()->to('/dashboard');
53	} else {
54	\$session->setFlashdata('msg', 'Password salah');
55	return redirect()->to('/login');
56	}
57	} else {
58	\$session->setFlashdata('msg', 'Username atau Email
59	salah');
59	return redirect()->to('/login');
60	}
61	}
62	
63	public function logout()
64	{
65	\$session = session();
66	\$session->destroy();
67	return redirect()->to('/login');
68	}
69	}

Tabel 3. Source Code Buku.php

Buku.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	{
10	public function __construct()
11	{
12	helper(['form', 'url']);
13	}
14	
15	public function index()
16	{
17	\$bukuModel = new BukuModel();
18	
19	// Mendapatkan keyword pencarian
20	\$search = \$this->request->getGet('search');
21	
22	// Mendapatkan parameter urut
23	\$sort = \$this->request->getGet('sort');
24	
25	//sort order
26	\$sort_order = \$this->request->getVar('sort_order');
27	
28	// Query dasar
29	\$query = \$bukuModel;


```

30
31     // Jika ada keyword pencarian
32     if ($search) {
33         $query = $query->like('judul', $search)
34                     ->orLike('penulis', $search)
35                     ->orLike('penerbit', $search)
36                     ->orLike('tahun_terbit', $search);
37     }
38
39     // Jika ada parameter urut
40     if ($sort && in_array($sort, ['judul', 'penulis', 'penerbit',
'tahun_terbit'])) {
41         $bukuModel->orderBy($sort, $sort_order ? $sort_order :
'ASC');
42     }
43
44     $data['buku'] = $query->findAll();
45     $data['search'] = $search;
46     $data['sort'] = $sort;
47     $data['sort_order'] = $sort_order;
48
49     return view('buku/index', $data);
50 }
51
52 public function create()
53 {
54     return view('buku/create');
55 }
56
57 public function store()
58 {
59     $validation = \Config\Services::validation();
60
61     $validation->setRules([
62         'judul' => [
63             'label' => 'Judul',
64             'rules' => 'required|string',
65             'errors' => [
66                 'required' => 'Judul harus diisi.',
67                 'string' => 'Judul harus berupa string.'
68             ]
69         ],
70         'penulis' => [
71             'label' => 'Penulis',
72             'rules' => 'required|string',
73             'errors' => [
74                 'required' => 'Penulis harus diisi.',
75                 'string' => 'Penulis harus berupa string.'
76             ]
77         ],
78         'penerbit' => [
79             'label' => 'Penerbit',
80             'rules' => 'required|string',
81             'errors' => [
82                 'required' => 'Penerbit harus diisi.',
83                 'string' => 'Penerbit harus berupa string.'
84             ]
85         ],

```

```

86         'tahun_terbit' => [
87             'label' => 'Tahun Terbit',
88             'rules' =>
            'required|numeric|greater_than[1800]|less_than[2024]',
89             'errors' => [
90                 'required' => 'Tahun Terbit harus diisi.',
91                 'numeric' => 'Tahun Terbit harus berupa angka.',
92                 'greater_than' => 'Tahun Terbit harus lebih besar
dari 1800.',
93                 'less_than' => 'Tahun Terbit harus kurang dari
2024.'
94             ]
95         ]
96     });
97
98     if ($validation->withRequest($this->request)->run() == FALSE)
99     {
100         return view('buku/create', ['validation' => $validation]);
101     } else {
102         $bukuModel = new BukuModel();
103         $data = [
104             'judul' => $this->request->getPost('judul'),
105             'penulis' => $this->request->getPost('penulis'),
106             'penerbit' => $this->request->getPost('penerbit'),
107             'tahun_terbit' => $this->request-
>getPost('tahun_terbit'),
108         ];
109         $bukuModel->save($data);
110         return redirect()->to('/buku');
111     }
112
113     public function edit($id)
114     {
115         $bukuModel = new BukuModel();
116         $data['buku'] = $bukuModel->find($id);
117         return view('buku/edit', $data);
118     }
119
120     public function update($id)
121     {
122         $validation = \Config\Services::validation();
123
124         $validation->setRules([
125             'judul' => [
126                 'label' => 'Judul',
127                 'rules' => 'required|string',
128                 'errors' => [
129                     'required' => 'Judul harus diisi.',
130                     'string' => 'Judul harus berupa string.'
131                 ]
132             ],
133             'penulis' => [
134                 'label' => 'Penulis',
135                 'rules' => 'required|string',
136                 'errors' => [
137                     'required' => 'Penulis harus diisi.',
138                     'string' => 'Penulis harus berupa string.'

```

```

139         ]
140     ],
141     'penerbit' => [
142         'label' => 'Penerbit',
143         'rules' => 'required|string',
144         'errors' => [
145             'required' => 'Penerbit harus diisi.',
146             'string' => 'Penerbit harus berupa string.'
147         ]
148     ],
149     'tahun_terbit' => [
150         'label' => 'Tahun Terbit',
151         'rules' =>
152         'required|numeric|greater_than[1800]|less_than[2024]',
153         'errors' => [
154             'required' => 'Tahun Terbit harus diisi.',
155             'numeric' => 'Tahun Terbit harus berupa angka.',
156             'greater_than' => 'Tahun Terbit harus lebih besar dari
1800.',
157             'less_than' => 'Tahun Terbit harus kurang dari 2024.'
158         ]
159     ]
160 ];
161 if ($validation->withRequest($this->request)->run() == FALSE) {
162     $bukuModel = new BukuModel();
163     return view('buku/edit', [
164         'validation' => $validation,
165         'buku' => $bukuModel->find($id)
166     ]);
167 } else {
168     $bukuModel = new BukuModel();
169     $data = [
170         'judul' => $this->request->getPost('judul'),
171         'penulis' => $this->request->getPost('penulis'),
172         'penerbit' => $this->request->getPost('penerbit'),
173         'tahun_terbit' => $this->request->getPost('tahun_terbit'),
174     ];
175     $bukuModel->update($id, $data);
176     return redirect()->to('/buku');
177 }
178 }
179
180 public function delete($id)
181 {
182     $bukuModel = new BukuModel();
183     $bukuModel->delete($id);
184     return redirect()->to('/buku');
185 }
186 }

```

Tabel 4. Source Code 2024-06-08-133024_CreateUsersTable.php

2024-06-08-162324 CreateUsersTable.php	
1	<?php
2	
3	namespace App\Database\Migrations;
4	

```

5 use CodeIgniter\Database\Migration;
6
7 class CreateUsersTable extends Migration
8 {
9     public function up()
10     {
11         $this->forge->addField([
12             'id' => ['type' => 'INT', 'auto_increment' =>
13 true],
14             'username' => ['type' => 'VARCHAR', 'constraint' =>
15 '100'],
16             'email' => ['type' => 'VARCHAR', 'constraint' =>
17 '100'],
18             'password' => ['type' => 'TEXT'],
19         ]);
20         $this->forge->addKey('id', true);
21         $this->forge->createTable('user');
22     }
23
24     public function down()
25     {
26         $this->forge->dropTable('user');
27     }
28 }

```

Tabel 5. Source Code 2024-06-08-134348_CreateBooksTable.php

2024-06-08-134348 CreateBooksTable.php	
1	<?php
2	
3	namespace App\Database\Migrations;
4	
5	use CodeIgniter\Database\Migration;
6	
7	class CreateBooksTable extends Migration
8	{
9	public function up()
10	{
11	\$this->forge->addField([
12	'id' => ['type' => 'BIGINT', 'auto_increment'
13	=> true],
14	'judul' => ['type' => 'VARCHAR', 'constraint' =>
15	'255'],
16	'penulis' => ['type' => 'VARCHAR', 'constraint' =>
17	'255'],
18	'penerbit' => ['type' => 'VARCHAR', 'constraint' =>
19	'255'],
20	'tahun_terbit' => ['type' => 'YEAR'],
21]);
22	\$this->forge->addKey('id', true);
23	\$this->forge->createTable('buku');
24	}
25	
26	public function down()
27	{
28	\$this->forge->dropTable('buku');
29	}
30	}

Tabel 6. Source Code AuthFilter.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	/**
12	* Do whatever processing this filter needs to do.
13	* By default it should not return anything during
14	* normal execution. However, when an abnormal state
15	* is found, it should return an instance of
16	* CodeIgniter\HTTP\Response. If it does, script
17	* execution will end and that Response will be
18	* sent back to the client, allowing for error pages,
19	* redirects, etc.
20	*
21	* @param RequestInterface \$request
22	* @param array null \$arguments
23	*
24	* @return RequestInterface ResponseInterface string void
25	*/
26	public function before(RequestInterface \$request, \$arguments =
27	null)
28	{
29	if (!session()->get('logged_in')) {
30	return redirect()->to('/login')->with('msg', 'Login
31	terlebih dahulu!');;
32	}
33	}
34	/**
35	* Allows After filters to inspect and modify the response
36	* object as needed. This method does not allow any way
37	* to stop execution of other after filters, short of
38	* throwing an Exception or Error.
39	*
40	* @param RequestInterface \$request
41	* @param ResponseInterface \$response
42	* @param array null \$arguments
43	*
44	* @return ResponseInterface void
45	*/
46	public function after(RequestInterface \$request, ResponseInterface
47	\$response, \$arguments = null)
48	{
49	//
50	}
51	}

Tabel 7. Source Code BukuModel.php

BukuModel.php	
1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class BukuModel extends Model
8	{
9	protected \$table = 'buku';
10	protected \$primaryKey = 'id';
11	protected \$useAutoIncrement = true;
12	protected \$returnType = 'array';
13	protected \$useSoftDeletes = false;
14	protected \$protectFields = true;
15	protected \$allowedFields = ['judul', 'penulis', 'penerbit', 'tahun_terbit'];
16	
17	protected bool \$allowEmptyInserts = false;
18	protected bool \$updateOnlyChanged = true;
19	
20	protected array \$casts = [];
21	protected array \$castHandlers = [];
22	
23	// Dates
24	protected \$useTimestamps = false;
25	protected \$dateFormat = 'datetime';
26	protected \$createdField = 'created_at';
27	protected \$updatedField = 'updated_at';
28	protected \$deletedField = 'deleted_at';
29	
30	// Validation
31	protected \$validationRules = [];
32	protected \$validationMessages = [];
33	protected \$skipValidation = false;
34	protected \$cleanValidationRules = true;
35	
36	// Callbacks
37	protected \$allowCallbacks = true;
38	protected \$beforeInsert = [];
39	protected \$afterInsert = [];
40	protected \$beforeUpdate = [];
41	protected \$afterUpdate = [];
42	protected \$beforeFind = [];
43	protected \$afterFind = [];
44	protected \$beforeDelete = [];
45	protected \$afterDelete = [];
46	}

Tabel 8. Source Code UserModel.php

peminjaman.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     protected $useAutoIncrement = true;
12     protected $returnType      = 'array';
13     protected $useSoftDeletes   = false;
14     protected $protectFields    = true;
15     protected $allowedFields    = ['username', 'email', 'password',
16     'created_at', 'updated_at'];
17
18     public function getUserByEmailOrUsername($identifier)
19     {
20         return $this->where('email', $identifier)
21             ->orWhere('username', $identifier)
22             ->first();
23     }
24
25     protected bool $allowEmptyInserts = false;
26     protected bool $updateOnlyChanged = true;
27
28     protected array $casts = [];
29     protected array $castHandlers = [];
30
31     // Dates
32     protected $useTimestamps = false;
33     protected $dateFormat    = 'datetime';
34     protected $createdField   = 'created_at';
35     protected $updatedField   = 'updated_at';
36     protected $deletedField   = 'deleted_at';
37
38     // Validation
39     protected $validationRules      = [];
40     protected $validationMessages   = [];
41     protected $skipValidation        = false;
42     protected $cleanValidationRules = true;
43
44     // Callbacks
45     protected $allowCallbacks = true;
46     protected $beforeInsert    = [];
47     protected $afterInsert     = [];
48     protected $beforeUpdate    = [];
49     protected $afterUpdate     = [];
50     protected $beforeFind      = [];
51     protected $afterFind       = [];
52     protected $beforeDelete    = [];
53     protected $afterDelete     = [];
54 }

```

Tabel 9. Source Code login.php

login.php

```

1 <?= $this->extend('layouts/main') ?>
2 <?= $this->section('content') ?>
3 <div class="mt-5">
4 <h2>Login</h2>
5 <?php if(session()->getFlashdata('msg')):?>
6     <div class="alert alert-danger"><?= session()->getFlashdata('msg')
7     ?></div>
8 <?php endif;?>
9 <form action="/auth/loginAuth" method="post">
10     <div class="form-group">
11         <label for="username">Username or Email:</label>
12         <input type="text" class="form-control" name="identifier"
13         id="identifier" required>
14     </div>
15     <div class="form-group">
16         <label for="password">Password:</label>
17         <input type="password" class="form-control" name="password"
18         id="password" required>
19     </div>
20     <button type="submit" class="btn btn-primary">Login</button>
21     <a href="/register" class="btn btn-link">Buat akun</a>
22 </form>
23 </div>
24 <?= $this->endSection() ?>

```

Tabel 10. Source Code register.php

register.php	
1	<?= \$this->extend('layouts/main') ?>
2	<?= \$this->section('content') ?>
3	<div class="mt-5">
4	<h2>Register</h2>
5	<form action="/auth/store" method="post">
6	<div class="form-group">
7	<label for="username">Username:</label>
8	<input type="text" class="form-control" name="username"
9	id="username" required>
10	</div>
11	<div class="form-group">
12	<label for="email">Email:</label>
13	<input type="email" class="form-control" name="email"
14	id="email" required>
15	</div>
16	<div class="form-group">
17	<label for="password">Password:</label>
18	<input type="password" class="form-control" name="password"
19	id="password" required>
20	</div>
21	<button type="submit" class="btn btn-primary">Register</button>
22	Kembali
23	</form>
24	</div>
25	<?= \$this->endSection() ?>

Tabel 11. Source Code create.php

create.php	
1	<?= \$this->extend('layouts/main') ?>
2	<?= \$this->section('content') ?>
3	
4	<div class="mt-5">
5	<h2>Tambah Buku</h2>
6	<?php if(isset(\$validation)):?>
7	<div class="alert alert-danger"><?= \$validation->listErrors()
	?></div>
8	<?php endif;?>
9	<form action="/buku/store" method="post">
10	<div class="form-group">
11	<label for="judul">Judul:</label>
12	<input type="text" class="form-control" name="judul"
	id="judul" required>
13	</div>
14	<div class="form-group">
15	<label for="penulis">Penulis:</label>
16	<input type="text" class="form-control" name="penulis"
	id="penulis" required>
17	</div>
18	<div class="form-group">
19	<label for="penerbit">Penerbit:</label>
20	<input type="text" class="form-control" name="penerbit"
	id="penerbit" required>
21	</div>
22	<div class="form-group">
23	<label for="tahun_terbit">Tahun Terbit:</label>
24	<input type="number" class="form-control" name="tahun_terbit"
	id="tahun_terbit" required>
25	</div>
26	<button type="submit" class="btn btn-primary">Tambahkan</button>
27	Kembali
28	</form>
29	</div>
30	<?= \$this->endSection() ?>

Tabel 12. Source Code edit.php

edit.php	
1	<?= \$this->extend('layouts/main') ?>
2	<?= \$this->section('content') ?>
3	
4	<div class="mt-5">
5	<h2>Edit Buku</h2>
6	<?php if(isset(\$validation)):?>
7	<div class="alert alert-danger"><?= \$validation->listErrors()
	?></div>
8	<?php endif;?>
9	<form action="/buku/update/<?= \$buku['id'] ?>" method="post">
10	<div class="form-group">
11	<label for="judul">Judul:</label>
12	<input type="text" class="form-control" name="judul"
	id="judul" value="<?= \$buku['judul'] ?>" required>
13	</div>
14	<div class="form-group">
15	<label for="penulis">Penulis:</label>

16	<code><input type="text" class="form-control" name="penulis" id="penulis" value="<?=\$buku['penulis'] ?>" required></code>
17	<code></div></code>
18	<code><div class="form-group"></code>
19	<code><label for="penerbit">Penerbit:</label></code>
20	<code><input type="text" class="form-control" name="penerbit" id="penerbit" value="<?=\$buku['penerbit'] ?>" required></code>
21	<code></div></code>
22	<code><div class="form-group"></code>
23	<code><label for="tahun_terbit">Tahun Terbit:</label></code>
24	<code><input type="number" class="form-control" name="tahun_terbit" id="tahun_terbit" value="<?=\$buku['tahun_terbit'] ?>" required></code>
25	<code></div></code>
26	<code><button type="submit" class="btn btn-primary">Ubah</button></code>
27	<code>Kembali</code>
28	<code></form></code>
29	<code></div></code>
30	<code><?=\$this->endSection() ?></code>

Tabel 13. Source Code index.php

index.php	
1	<code><?=\$this->extend('layouts/main') ?></code>
2	<code><?=\$this->section('content') ?></code>
3	
4	<code><div class="mt-5"></div></code>
5	
6	<code><h1 class="text-center mb-4">Data Buku</h1></code>
7	
8	<code><?php if (session()->get('logged_in')) : ?></code>
9	<code><div class="d-flex justify-content-between align-items-center mb-3"></code>
10	<code><div></code>
11	<code>Tambah Buku</code>
12	<code></div></code>
13	
14	<code><form method="get" action="/buku" class="form-inline"></code>
15	<code><div class="form-group"></code>
16	<code><input type="text" name="search" class="form-control" placeholder="Cari buku..." value="<?=\$search ?>"></code>
17	<code></div></code>
18	<code><div class="form-group mx-sm-3"></code>
19	<code><select name="sort" class="form-control"></code>
20	<code><option value="">Urutkan berdasarkan...</option></code>
21	<code><option value="judul" <?=\$sort == 'judul' ? 'selected' : '' ?>>Judul</option></code>
22	<code><option value="penulis" <?=\$sort == 'penulis' ? 'selected' : '' ?>>Penulis</option></code>
23	<code><option value="penerbit" <?=\$sort == 'penerbit' ? 'selected' : '' ?>>Penerbit</option></code>
24	<code><option value="tahun_terbit" <?=\$sort == 'tahun_terbit' ? 'selected' : '' ?>>Tahun Terbit</option></code>
25	<code></select></code>
26	<code></div></code>
27	<code><button type="submit" class="btn btn-outline-primary">Cari</button></code>

```

28         <a href="/buku" class="btn btn-outline-secondary ml-
29 2">Show all</a>
30     </form>
31     <div>
32         <a href="#" class="btn btn-success">Selamat Datang, <?=
33 session()->get('username') ?></a>
34         <a href="/logout" class="btn btn-danger">Logout</a>
35     </div>
36 <?php endif; ?>
37
38 <div class="table-responsive">
39     <table class="table table-bordered table-striped">
40         <thead class="thead-dark">
41             <tr>
42                 <th>
43                     <a href="/buku?sort=judul&sort_order=<?= $sort ==
44 'judul' && $sort_order == 'ASC' ? 'DESC' : 'ASC' ?>"
45 style="color:white">
46                         Judul <?= $sort == 'judul' ? ($sort_order ==
47 'ASC' ? '▲' : '▼') : '' ?>
48                     </a>
49                 </th>
50                 <th>
51                     <a href="/buku?sort=penulis&sort_order=<?= $sort
52 == 'penulis' && $sort_order == 'ASC' ? 'DESC' : 'ASC' ?>"
53 style="color:white">
54                         Penulis <?= $sort == 'penulis' ? ($sort_order
55 == 'ASC' ? '▲' : '▼') : '' ?>
56                     </a>
57                 </th>
58                 <th>
59                     <a href="/buku?sort=penerbit&sort_order=<?= $sort
60 == 'penerbit' && $sort_order == 'ASC' ? 'DESC' : 'ASC' ?>"
61 style="color:white">
62                         Penerbit <?= $sort == 'penerbit' ?
63 ($sort_order == 'ASC' ? '▲' : '▼') : '' ?>
64                     </a>
65                 </th>
66                 <th>
67                     <a href="/buku?sort=tahun_terbit&sort_order=<?=
68 $sort == 'tahun_terbit' && $sort_order == 'ASC' ? 'DESC' : 'ASC' ?>"
69 style="color:white">
70                         Tahun Terbit <?= $sort == 'tahun_terbit' ?
71 ($sort_order == 'ASC' ? '▲' : '▼') : '' ?>
72                     </a>
73                 </th>
74                 <th scope="col" class="text-center">Aksi</th>
75             </tr>
76         </thead>
77         <tbody>
78             <?php foreach ($bukus as $buku) : ?>
79                 <tr>
80                     <td><?= $buku['judul'] ?></td>
81                     <td><?= $buku['penulis'] ?></td>
82                     <td><?= $buku['penerbit'] ?></td>

```

```

72         <td><?= $buku['tahun_terbit'] ?></td>
73         <td class="text-center">
74             <a href="/buku/edit/<?= $buku['id'] ?>"
class="btn btn-outline-warning btn-sm">Edit</a>
75             <a href="/buku/delete/<?= $buku['id'] ?>"
class="btn btn-outline-danger btn-sm" onclick="return confirm('Are you
sure you want to delete this item?');">Delete</a>
76         </td>
77     </tr>
78     <?php endforeach; ?>
79 </tbody>
80 </table>
81 </div>
82 <div class="mb-5"></div>
83
84 <?= $this->endSection() ?>

```

Tabel 14. Source Code main.php

main.php	
1	<!DOCTYPE html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="UTF-8">
6	<meta name="viewport" content="width=device-width, initial-
	scale=1.0">
7	<title>PRAK601</title>
8	<link rel="stylesheet"
	href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
	.min.css">
9	</head>
10	
11	<body>
12	<div class="container">
13	<?= \$this->renderSection('content') ?>
14	</div>
15	<script src="https://code.jquery.com/jquery-
	3.5.1.slim.min.js"></script>
16	<script
	src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper
	.min.js"></script>
17	<script
	src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.m
	in.js"></script>
18	</body>
19	
20	</html>

B. Output Program



Login

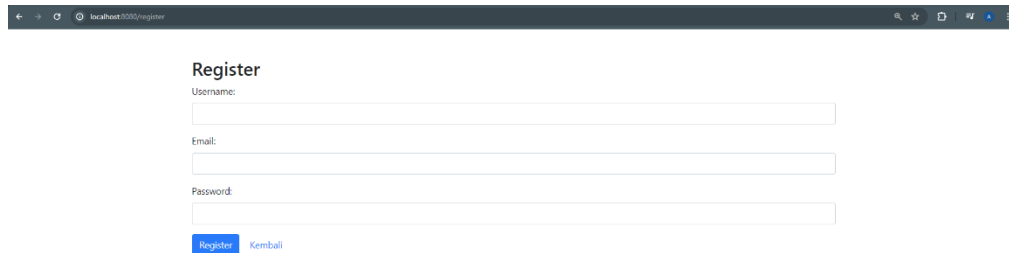
Login terlebih dahulu!

Username or Email:

Password:

Login Buat akun

Gambar 2. Tampilan .php



Register

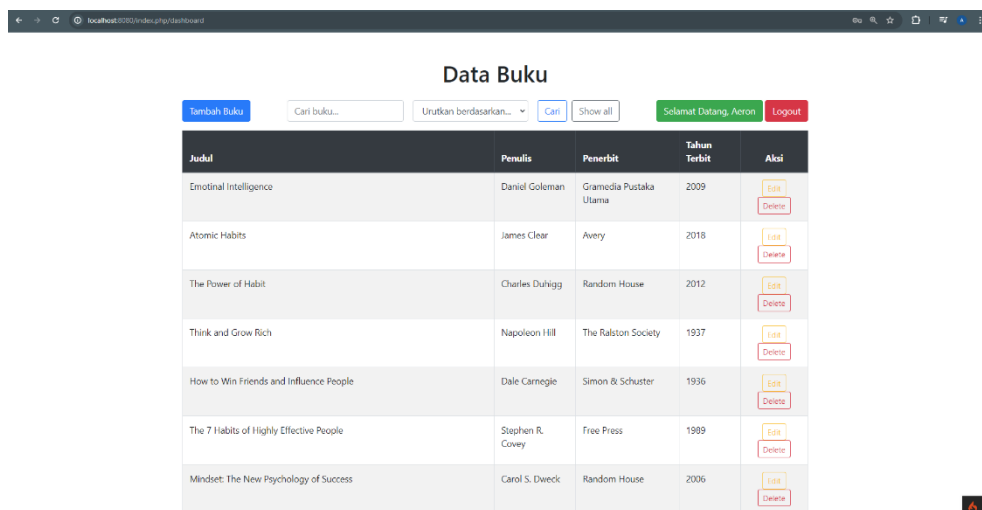
Username:

Email:

Password:

Register Kembali

Gambar 3. Tampilan register.php



Data Buku

Tambah Buku

Cari buku...

Urutkan berdasarkan...

Cari

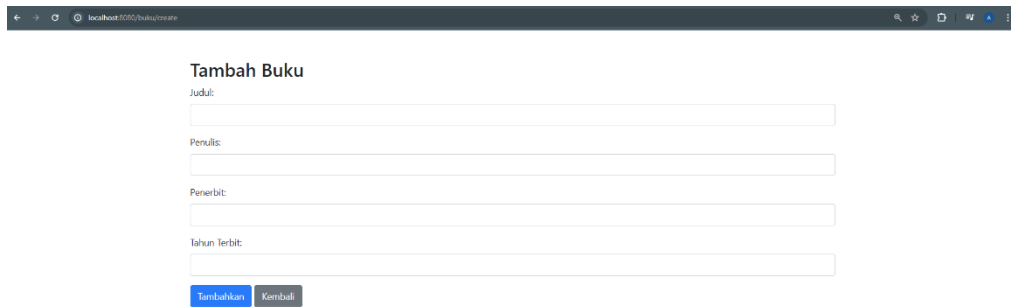
Show all

Selamat Datang, Aeron

Logout

Judul	Penulis	Penerbit	Tahun Terbit	Aksi
Emotional Intelligence	Daniel Goleman	Gramedia Pustaka Utama	2009	Edit Delete
Atomic Habits	James Clear	Avery	2018	Edit Delete
The Power of Habit	Charles Duhigg	Random House	2012	Edit Delete
Think and Grow Rich	Napoleon Hill	The Rakiton Society	1937	Edit Delete
How to Win Friends and Influence People	Dale Carnegie	Simon & Schuster	1936	Edit Delete
The 7 Habits of Highly Effective People	Stephen R. Covey	Free Press	1989	Edit Delete
Mindset: The New Psychology of Success	Carol S. Dweck	Random House	2006	Edit Delete

Gambar 4. Tampilan index.php



← → ↻ localhost:8080/buku/create

Tambah Buku

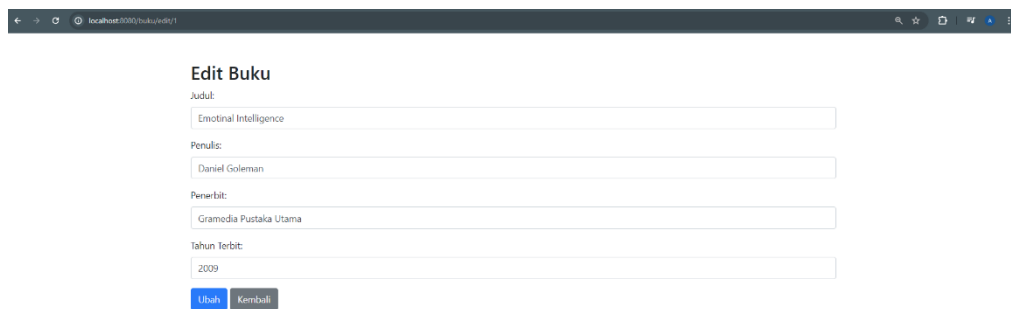
Judul:

Penulis:

Penerbit:

Tahun Terbit:

Gambar 5. Tampilan create.php



← → ↻ localhost:8080/buku/edit/1

Edit Buku

Judul:

Penulis:

Penerbit:

Tahun Terbit:

Gambar 6. Tampilan edit.php

C. Pembahasan

Routes.php

File `routes.php` digunakan untuk mendefinisikan rute aplikasi. Pada baris pertama, kita melihat rute untuk halaman utama yang diarahkan ke metode `index` dalam controller `Home`. Baris selanjutnya mendefinisikan rute untuk halaman login yang diarahkan ke metode `login` dalam controller `Auth`. Rute `/register` diarahkan ke metode `register` dalam controller `Auth`. Baris-baris berikutnya mengatur rute untuk berbagai operasi pada buku seperti `/buku` yang diarahkan ke metode `index` dalam controller `Buku`, `/buku/create` yang diarahkan ke metode `create`, dan `/buku/edit(/:num)` yang diarahkan ke metode `edit` dengan parameter ID buku. Terakhir, rute `/logout` diarahkan ke metode `logout` dalam controller `Auth`.

Auth.php

File `Auth.php` adalah controller untuk mengelola autentikasi pengguna. Pada metode `login`, sistem pertama kali menampilkan halaman login dengan `return view('auth/login');`. Jika ada data yang dikirim melalui metode POST, sistem memvalidasi input tersebut. Jika valid, metode `check_login` dipanggil untuk memverifikasi kredensial pengguna terhadap database. Jika kredensial benar, sesi pengguna dibuat menggunakan `session()->set` dan pengguna diarahkan ke halaman utama. Jika kredensial salah, pengguna diarahkan kembali ke halaman login dengan pesan kesalahan menggunakan `session()->setFlashdata`. Metode `register` berfungsi untuk menampilkan form pendaftaran dan memproses data pendaftaran yang dikirimkan. Setelah data valid, pengguna baru disimpan ke database dan diarahkan ke halaman login. Metode `logout` menghentikan sesi pengguna dengan `session()->destroy` dan mengarahkan mereka kembali ke halaman login.

Buku.php

File `Buku.php` adalah controller yang mengelola data buku. Metode `index` menampilkan daftar buku dengan memanggil model `BukuModel` dan mengakses metode `findAll` untuk mendapatkan semua data buku, kemudian menampilkannya dalam view `buku/index`. Metode `create` digunakan untuk menampilkan form penambahan buku

baru dengan ``return view('buku/create');``. Metode ``store`` memproses data dari form yang dikirim melalui POST, memvalidasi input, dan jika valid, menyimpan data buku baru ke database menggunakan metode ``save`` dari model ``BukuModel``. Setelah itu, pengguna diarahkan kembali ke daftar buku dengan pesan sukses. Metode ``edit`` menampilkan form pengeditan untuk buku yang dipilih berdasarkan ID buku, data buku diambil dari database menggunakan metode ``find`` dari model ``BukuModel``, dan ditampilkan dalam view ``buku/edit``. Metode ``update`` memproses data pengeditan yang dikirim melalui POST, memvalidasi input, dan jika valid, memperbarui data buku di database menggunakan metode ``update`` dari model ``BukuModel``. Terakhir, metode ``delete`` menghapus data buku dari database berdasarkan ID buku yang diberikan dan mengarahkan pengguna kembali ke daftar buku dengan pesan sukses.

2024-06-08-162324_CreateUsersTable.php

File ini adalah migrasi yang digunakan untuk membuat tabel ``users`` dalam database. Baris pertama mendefinisikan nama tabel sebagai ``users``. Selanjutnya, kolom-kolom dalam tabel didefinisikan, termasuk kolom ``id`` yang bertipe integer dan auto increment sebagai primary key, kolom ``username`` yang bertipe varchar dengan panjang maksimal 255 karakter, kolom ``email`` yang juga bertipe varchar, dan kolom ``password`` yang bertipe varchar untuk menyimpan hash kata sandi pengguna. Kolom ``created_at`` dan ``updated_at`` juga ditambahkan untuk mencatat waktu pembuatan dan pembaruan data. Pada akhirnya, tabel dibuat dengan metode ``createTable``.

2024-06-08-134348_CreateBooksTable.php

File migrasi ini digunakan untuk membuat tabel ``books`` dalam database. Pada baris pertama, nama tabel didefinisikan sebagai ``books``. Kolom ``id`` didefinisikan sebagai primary key dengan tipe integer dan auto increment. Kolom ``title`` bertipe varchar digunakan untuk menyimpan judul buku, dan kolom ``author`` juga bertipe varchar untuk menyimpan nama penulis. Kolom ``publisher`` dan ``year`` didefinisikan masing-masing untuk menyimpan nama penerbit dan tahun terbit buku. Terakhir, kolom ``created_at`` dan ``updated_at`` ditambahkan untuk mencatat waktu pembuatan dan pembaruan data. Metode ``createTable`` digunakan untuk membuat tabel ini dalam database.

AuthFilter.php

File `AuthFilter.php` adalah filter yang digunakan untuk mengamankan rute tertentu. Pada metode `before`, filter memeriksa apakah sesi pengguna aktif dengan memeriksa keberadaan sesi `isLoggedIn`. Jika tidak ada, pengguna akan diarahkan ke halaman login dengan `return redirect()->to('/login');`. Metode `after` tidak melakukan apa-apa dalam filter ini, tetapi disediakan untuk keperluan jika ada logika yang perlu dijalankan setelah request diproses. Filter ini memastikan bahwa hanya pengguna yang sudah login yang dapat mengakses rute yang dilindungi.

BukuModel.php

File `BukuModel.php` adalah model yang digunakan untuk mengelola data buku dalam database. Model ini mengextends `Model` dari framework dan mendefinisikan tabel yang digunakan sebagai `books`. Kolom-kolom yang dapat diisi didefinisikan dalam properti `allowedFields`, termasuk `title`, `author`, `publisher`, dan `year`. Properti ini memastikan bahwa hanya kolom-kolom tersebut yang dapat diisi secara massal untuk menghindari mass assignment vulnerability. Model ini juga menggunakan properti `useTimestamps` untuk mencatat waktu pembuatan dan pembaruan data secara otomatis.

peminjaman.php

File `peminjaman.php` adalah view yang digunakan untuk menampilkan data peminjaman buku. Dalam view ini, data peminjaman buku ditampilkan dalam bentuk tabel. Setiap baris tabel menampilkan informasi peminjaman seperti nama peminjam, judul buku, tanggal peminjaman, dan tanggal pengembalian. Ada juga tombol-tombol untuk mengedit dan menghapus data peminjaman. View ini memungkinkan pengguna untuk melihat, mengelola, dan memperbarui data peminjaman buku dengan mudah.

login.php

File `login.php` adalah view yang digunakan untuk menampilkan form login. Form ini meminta pengguna untuk memasukkan username dan password. Form ini juga memiliki tombol submit untuk mengirim data ke server. Jika ada pesan kesalahan, pesan tersebut ditampilkan di atas form. View ini menyediakan antarmuka bagi pengguna untuk masuk ke sistem.

register.php

File `register.php` adalah view yang digunakan untuk menampilkan form pendaftaran pengguna baru. Form ini meminta pengguna untuk memasukkan informasi seperti username, email, dan password. Setelah data dimasukkan, pengguna dapat mengirim form untuk mendaftar. Jika ada kesalahan dalam input, pesan kesalahan akan ditampilkan di atas form. View ini menyediakan antarmuka bagi pengguna untuk membuat akun baru.

create.php

File `create.php` adalah view yang digunakan untuk menampilkan form penambahan buku baru. Form ini meminta pengguna untuk memasukkan informasi buku seperti judul, penulis, penerbit, dan tahun terbit. Setelah data dimasukkan, pengguna dapat mengirim form untuk menyimpan buku baru ke database. Jika ada kesalahan dalam input, pesan kesalahan akan ditampilkan di atas form. View ini menyediakan antarmuka bagi pengguna untuk menambah buku baru ke sistem.

edit.php

File `edit.php` adalah view yang digunakan untuk menampilkan form pengeditan data buku. Form ini diisi dengan data buku yang ada dan memungkinkan pengguna untuk memperbarui informasi seperti judul, penulis, penerbit, dan tahun terbit. Setelah perubahan dibuat, pengguna dapat mengirim form untuk memperbarui data buku di database. Jika ada kesalahan dalam input, pesan kesalahan akan ditampilkan di atas form. View ini menyediakan antarmuka bagi pengguna untuk mengedit informasi buku yang sudah ada.

index.php

File `index.php` adalah view yang digunakan untuk menampilkan daftar buku. Dalam view ini, data buku ditampilkan dalam bentuk tabel. Setiap baris tabel menampilkan informasi buku seperti judul, penulis, penerbit, dan tahun terbit. Ada juga tombol-tombol untuk mengedit dan menghapus buku. View ini memungkinkan pengguna untuk melihat, mengelola, dan memperbarui data buku dengan mudah.

main.php

File `main.php` adalah template utama yang digunakan untuk menyusun tampilan antarmuka pengguna. Template ini mencakup struktur HTML dasar dan elemen-elemen seperti header, footer, dan menu navigasi. Bagian konten dinamis diisi dengan view-view spesifik seperti `index.php`, `create.php`, `edit.php`, dll., tergantung pada halaman yang sedang diakses. Template ini memastikan konsistensi tampilan dan pengalaman pengguna di seluruh aplikasi.formulir.

TAUTAN GIT

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/AeronTj/PemroWebII/tree/main/Modul6>