

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Muhammad Pasha Nabeel

NIM. 2210817110002

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

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN I
MODUL 6

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

Nama Praktikan : Muhammad Pasha Nabeel
NIM : 2210817110002

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

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

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code	7
B. Output Program.....	20
C. Pembahasan	22
D. Tautan Git	25

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	22
---	----

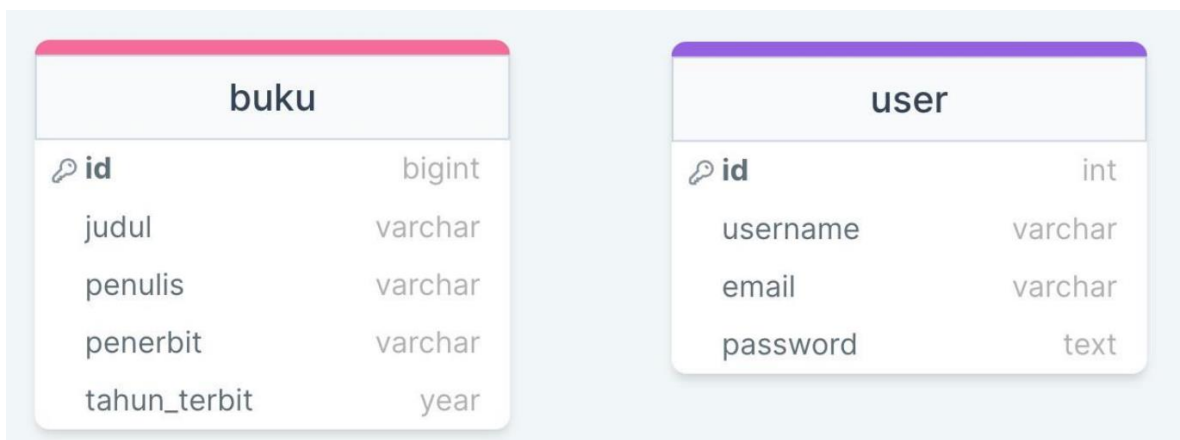
DAFTAR TABEL

Tabel 1 Source Code Soal 1	7
Tabel 2 Source Code Soal 1	8
Tabel 3 Source Code Soal 1	10
Tabel 4 Source Code Soal 1	11
Tabel 5 Source Code Soal 1	12
Tabel 6 Source Code Soal 1	13
Tabel 7 Source Code Soal 1	14
Tabel 8 Source Code Soal 1	15
Tabel 9 Source Code Soal 1	16
Tabel 10 Source Code Soal 1	17
Tabel 11 Source Code Soal 1	18

SOAL 1

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

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



Simpan dalam folder github PRAK 601

A. Source Code

File Routes.php

Tabel 1 Source Code Soal 1

1	<?php
2	
3	namespace Config;
4	
5	// Create a new instance of our RouteCollection class.
6	\$routes = Services::routes();
7	
8	/*
9	* -----
10	* Router Setup
11	* -----
12	*/
13	\$routes->setDefaultNamespace('App\Controllers');
14	\$routes->setDefaultController('Home');
15	\$routes->setDefaultMethod('index');
16	\$routes->setTranslateURIDashes(false);
17	\$routes->set404Override();
18	// The Auto Routing (Legacy) is very dangerous. It is easy to create vulnerable apps
19	// where controller filters or CSRF protection are bypassed.
20	// If you don't want to define all routes, please use the Auto Routing (Improved).
21	// Set `autoRoutesImproved` to true in `app/Config/Feature.php` and set the following to true.
22	// \$routes->setAutoRoute(false);
23	
24	/*
25	* -----
26	* Route Definitions
27	* -----
28	*/
29	
30	// We get a performance increase by specifying the default
31	// route since we don't have to scan directories.
32	\$routes->group('/', ["filter" => 'auth'], static function (\$routes) {
33	\$routes->get('/', 'BukuController::index');
34	\$routes->get('/buku', 'BukuController::index');
35	\$routes->get('/buku/create', 'BukuController::create');

36	<code>\$routes->post('/buku/create', 'BukuController::store');</code>
37	<code>\$routes->get('/buku/(:num)/edit', 'BukuController::edit/\$1');</code>
38	<code>\$routes->post('/buku/(:num)/edit', 'BukuController::update/\$1');</code>
39	<code>\$routes->delete('/buku/(:num)/delete', 'BukuController::delete/\$1');</code>
40	<code>\$routes->get('/logout', 'UserController::logout');</code>
41	<code>});</code>
42	
43	<code>\$routes->get('/login', 'UserController::index');</code>
44	<code>\$routes->post('/login', 'UserController::login');</code>
45	
46	<code>/*</code>
47	<code> * -----</code>
48	<code> * Additional Routing</code>
49	<code> * -----</code>
50	<code> *</code>
51	<code> * There will often be times that you need additional routing and you</code>
52	<code> * need it to be able to override any defaults in this file. Environment</code>
53	<code> * based routes is one such time. require() additional route files here</code>
54	<code> * to make that happen.</code>
55	<code> *</code>
56	<code> * You will have access to the \$routes object within that file without</code>
57	<code> * needing to reload it.</code>
58	<code> */</code>
59	<code>if (is_file(APPPATH . 'Config/' . ENVIRONMENT . '/Routes.php')) {</code>
60	<code> require APPPATH . 'Config/' . ENVIRONMENT . '/Routes.php';</code>
61	<code>}</code>

File BaseController.php

Tabel 2 Source Code Soal 1

1	<code><?php</code>
2	
3	<code>namespace App\Controllers;</code>
4	
5	<code>use CodeIgniter\Controller;</code>
6	<code>use CodeIgniter\HTTP\CLIRequest;</code>
7	<code>use CodeIgniter\HTTP\IncomingRequest;</code>
8	<code>use CodeIgniter\HTTP\RequestInterface;</code>
9	<code>use CodeIgniter\HTTP\ResponseInterface;</code>
10	<code>use Psr\Log\LoggerInterface;</code>
11	
12	<code>/**</code>


```

13  * Class BaseController
14  *
15  * BaseController provides a convenient place for loading components
16  * and performing functions that are needed by all your controllers.
17  * Extend this class in any new controllers:
18  *     class Home extends BaseController
19  *
20  * For security be sure to declare any new methods as protected or
21  private.
22  */
23  abstract class BaseController extends Controller
24  {
25      /**
26       * Instance of the main Request object.
27       *
28       * @var CLIRequest|IncomingRequest
29       */
30      protected $request;
31
32      /**
33       * An array of helpers to be loaded automatically upon
34       * class instantiation. These helpers will be available
35       * to all other controllers that extend BaseController.
36       *
37       * @var array
38       */
39      protected $helpers = [];
40
41      /**
42       * Be sure to declare properties for any property fetch you
43       initialized.
44       *
45       * The creation of dynamic property is deprecated in PHP 8.2.
46       */
47      // protected $session;
48
49      /**
50       * Constructor.
51       */
52      public function initController(RequestInterface $request,
53      ResponseInterface $response, LoggerInterface $logger)
54      {
55          // Do Not Edit This Line

```

52	parent::initController(\$request, \$response, \$logger);
53	
54	// Preload any models, libraries, etc, here.
55	
56	// E.g.: \$this->session = \Config\Services::session();
57	}
58	}

File BukuController.php

Tabel 3 Source Code Soal 1

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Controllers\BaseController;
6	use App\Models\BukuModel;
7	
8	class BukuController extends BaseController
9	{
10	public function index()
11	{
12	return view('buku/index', [
13	'buku' => (new BukuModel())->findAll()
14]);
15	}
16	
17	public function create()
18	{
19	return view('buku/create');
20	}
21	
22	public function store()
23	{
24	\$buku = new BukuModel();
25	
26	\$buku->insert([
27	'judul' => \$this->request->getPost('judul'),
28	'penulis' => \$this->request->getPost('penulis'),
29	'penerbit' => \$this->request->getPost('penerbit'),
30	'tahun_terbit' => \$this->request->getPost('tahun_terbit'),
31]);

32	
33	return redirect()->to('/buku');
34	}
35	
36	public function edit(\$id)
37	{
38	return view('buku/edit', [
39	'buku' => (new BukuModel())->find(\$id)
40]);
41	}
42	
43	public function update(\$id)
44	{
45	\$buku = new BukuModel();
46	
47	\$buku->update(\$id, [
48	'judul' => \$this->request->getPost('judul'),
49	'penulis' => \$this->request->getPost('penulis'),
50	'penerbit' => \$this->request->getPost('penerbit'),
51	'tahun_terbit' => \$this->request->getPost('tahun_terbit'),
52]);
53	
54	return redirect()->to('/buku');
55	}
56	
57	public function delete(\$id)
58	{
59	\$buku = new BukuModel();
60	
61	\$buku->delete(\$id);
62	
63	return redirect()->to('/buku');
64	}
65	}

File Home.php

Tabel 4 Source Code Soal 1

1	<?php
2	
3	namespace App\Controllers;
4	

5	class Home extends BaseController
6	{
7	public function index()
8	{
9	return view('login');
10	}
11	}

File UserController.php

Tabel 5 Source Code Soal 1

1	<?php
2	
3	namespace App\Controllers;
4	
5	use App\Controllers\BaseController;
6	use App\Models\UserModel;
7	
8	class UserController extends BaseController
9	{
	public function index()
10	{
11	return view('login');
12	}
13	
14	public function login()
15	{
16	\$user = new UserModel();
17	\$username = \$this->request->getPost('username');
18	\$password = \$this->request->getPost('password');
19	\$dataUser = \$user->where(['username' => \$username])->
	orWhere(['email' => \$username])->first();
20	if (\$dataUser) {
21	if (password_verify(\$password, \$dataUser['password'])) {
22	session()->set([
23	'username' => \$dataUser['username'],
24	'email' => \$dataUser['email'],
25	'logged_in' => true
26]);
27	
28	return redirect()->to(base_url('buku'));
29	} else {

30	session()->setFlashdata('pesan', 'Password salah');
31	return redirect()->to(base_url('login'));
32	}
33	} else {
34	session()->setFlashdata('pesan', 'Username atau email tidak ditemukan');
35	return redirect()->to(base_url('login'));
36	}
37	}
38	
39	public function logout()
40	{
41	session()->destroy();
42	return redirect()->to(base_url('login'));
43	}
44	
45	}

File 2024-06-14-172721_Buku.php

Tabel 6 Source Code Soal 1

1	<?php
2	
3	namespace App\Database\Migrations;
4	
5	use CodeIgniter\Database\Migration;
6	
7	class Buku extends Migration
8	{
9	public function up()
10	{
11	\$this->forge->addField([
12	'id' => [
13	'type' => 'INT',
14	'constraint' => 5,
15	'unsigned' => true,
16	'auto_increment' => true
17],
18	'judul' => [
19	'type' => 'VARCHAR',
20	'constraint' => 100,
21],

22	'penulis' => [
23	'type' => 'VARCHAR',
24	'constraint' => 100,
25],
26	'penerbit' => [
27	'type' => 'VARCHAR',
28	'constraint' => 100,
29],
30	'tahun_terbit' => [
31	'type' => 'YEAR',
32],
33]);
34	
35	\$this->forge->addKey('id', true);
36	\$this->forge->createTable('buku');
37	}
38	
39	public function down()
40	{
41	\$this->forge->dropTable('buku', true);
42	}
43	}

File 2024-06-14-172729_User.php

Tabel 7 Source Code Soal 1

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

17],
18	'username' => [
19	'type' => 'VARCHAR',
20	'constraint' => 100,
21],
22	'email' => [
23	'type' => 'VARCHAR',
24	'constraint' => 100,
25],
26	'password' => [
27	'type' => 'TEXT',
28	
29],
30	'created_at' => [
31	'type' => 'DATETIME',
32],
33	'updated_at' => [
34	'type' => 'DATETIME',
35],
36	'deleted_at' => [
37	'type' => 'DATETIME',
38],
39]);
40	
41	\$this->forge->addKey('id', true);
42	\$this->forge->createTable('user', true);
43	}
44	
45	public function down()
46	{
47	\$this->forge->dropTable('user', true);
48	}
49	}

File Buku.php

Tabel 8 Source Code Soal 1

1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;

6	use App\Models\BukuModel;
7	
8	class Buku extends Seeder
9	{
10	public function run()
11	{
12	\$this->db->table('buku')->insert([
13	'judul' => 'Belajar Pemrograman Java',
14	'penulis' => 'Petani Kode',
15	'penerbit' => 'PT Gramedia',
16	'tahun_terbit' => '2020'
17]);
18	
19	\$this->db->table('buku')->insert([
20	'judul' => 'Belajar Pemrograman PHP',
21	'penulis' => 'Petani Kode',
22	'penerbit' => 'PT Gramedia',
23	'tahun_terbit' => '2020'
24]);
25	
26	\$this->db->table('buku')->insert([
27	'judul' => 'Belajar Pemrograman Python',
28	'penulis' => 'Petani Kode',
29	'penerbit' => 'PT Gramedia',
30	'tahun_terbit' => '2020'
31]);
32	}
33	}

File UserSeeder.php

Tabel 9 Source Code Soal 1

1	<?php
2	
3	namespace App\Database\Seeds;
4	
5	use CodeIgniter\Database\Seeder;
6	
7	class UserSeeder extends Seeder
8	{
9	public function run()
10	{

11	\$this->db->table('user')->insert([
12	'username' => 'admin',
13	'email' => 'admin@mail.com',
14	'password' => password_hash('12345678', PASSWORD_DEFAULT),
15]);
16	}
17	}

File BukuModel.php

Tabel 10 Source Code Soal 1

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class BukuModel extends Model
8	{
9	protected \$DBGroup = 'default';
10	protected \$table = 'buku';
11	protected \$primaryKey = 'id';
12	protected \$useAutoIncrement = true;
13	protected \$returnType = 'array';
14	protected \$useSoftDeletes = false;
15	protected \$protectFields = true;
16	protected \$allowedFields = [
17	'judul',
18	'penulis',
19	'penerbit',
20	'tahun_terbit'
21];
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	}

File UserModel.php

Tabel 11 Source Code Soal 1

1	<?php
2	
3	namespace App\Models;
4	
5	use CodeIgniter\Model;
6	
7	class UserModel extends Model
8	{
9	protected \$DBGroup = 'default';
10	protected \$table = 'user';
11	protected \$primaryKey = 'id';
12	protected \$useAutoIncrement = true;
13	protected \$returnType = 'array';
14	protected \$useSoftDeletes = false;
15	protected \$protectFields = true;
16	protected \$allowedFields = [];
17	
18	// Dates
19	protected \$useTimestamps = false;
20	protected \$dateFormat = 'datetime';
21	protected \$createdField = 'created_at';
22	protected \$updatedField = 'updated_at';

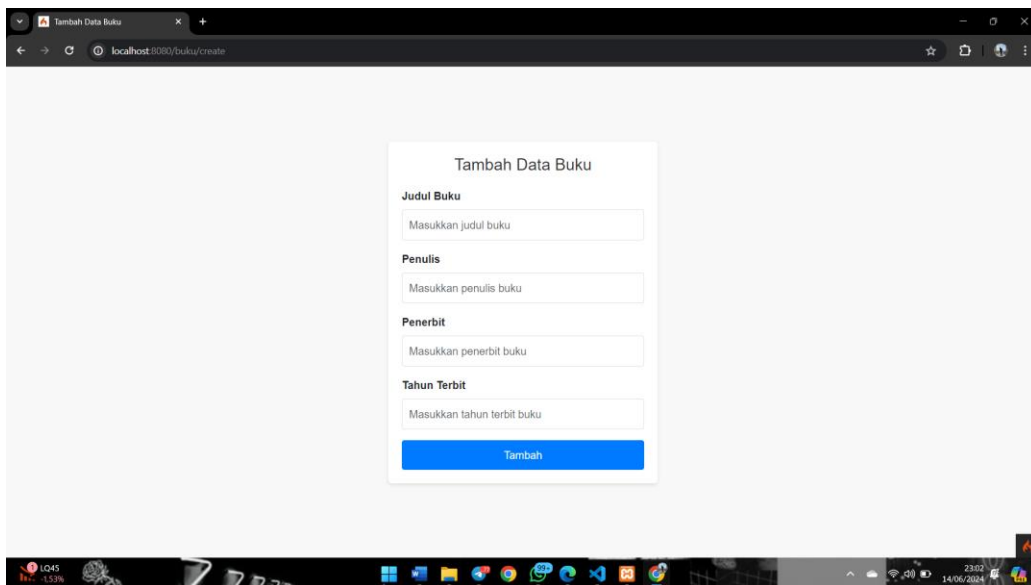
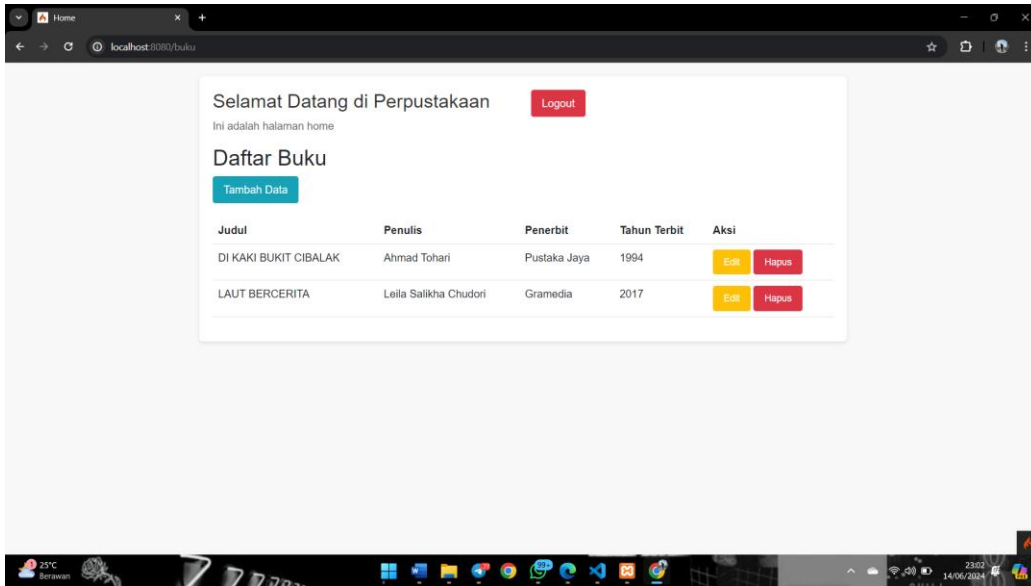
```
23     protected $deletedField = 'deleted_at';
24
25     // Validation
26     protected $validationRules      = [];
27     protected $validationMessages  = [];
28     protected $skipValidation       = false;
29     protected $cleanValidationRules = true;
30
31     // Callbacks
32     protected $allowCallbacks = true;
33     protected $beforeInsert    = [];
34     protected $afterInsert     = [];
35     protected $beforeUpdate    = [];
36     protected $afterUpdate     = [];
37     protected $beforeFind      = [];
38     protected $afterFind       = [];
39     protected $beforeDelete    = [];
40     protected $afterDelete     = [];
41 }
```

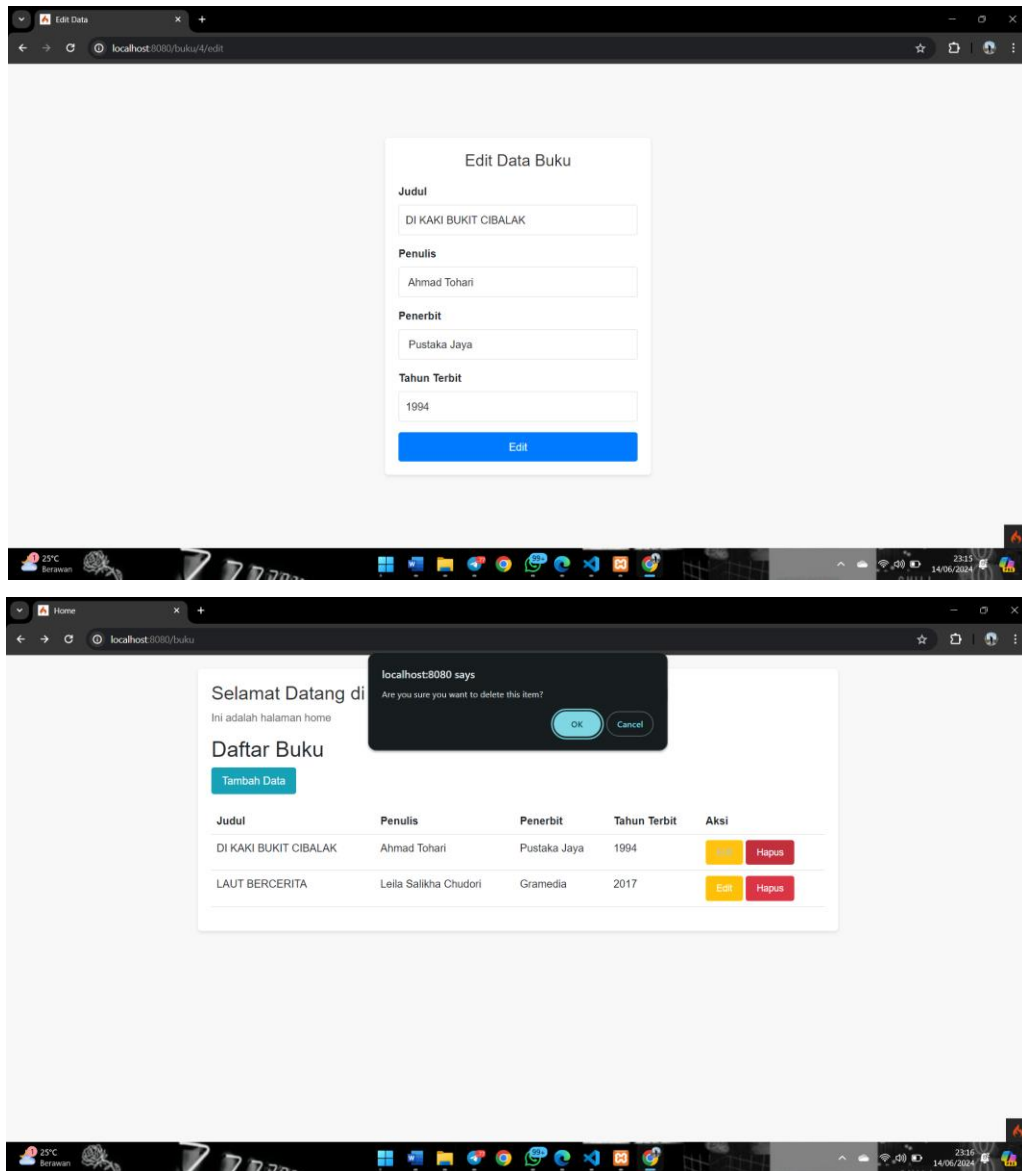
B. Output Program

The image displays two screenshots of a web browser window showing a login page. The browser's address bar indicates the URL is `localhost:8080/login`.

Top Screenshot: The login form is titled "Login". It contains a yellow message box that says "Login terlebih dahulu!". Below this, there are two input fields: "Username/Email address" and "Password". A blue "Sign in" button is located at the bottom of the form.

Bottom Screenshot: The login form is titled "Login". It contains a yellow message box that says "Username atau email tidak ditemukan". Below this, the "Username/Email address" field is filled with the text "pasha@09". The "Password" field is filled with a series of asterisks "*****". A blue "Sign in" button is located at the bottom of the form.





Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

1) Routes.php

Semua pengaturan ini digunakan untuk mengkonfigurasi rute CodeIgniter sehingga aplikasi dapat memproses URL dengan benar dan mengarahkannya ke kontrol dan metode yang sesuai. Dengan menggunakan konfigurasi rute ini, aplikasi memiliki rute yang sesuai untuk berbagai fungsi seperti menampilkan daftar buku, menambah buku baru, mengedit buku, menghapus buku, dan mengelola proses login dan logout pengguna.

2) BaseController.php

BaseController merupakan kelas yang diturunkan oleh semua kontroler dalam aplikasi. Ini memberikan tempat yang nyaman untuk memuat komponen dan melakukan fungsi yang dibutuhkan oleh semua kontroler, seperti inisialisasi objek, pengaturan konfigurasi umum, dan pemanggilan model atau library tertentu. Dapat mengubah dan menyesuaikan BaseController sesuai kebutuhan aplikasi.

3) BaseController.php

Kelas BukuController bertanggung jawab untuk menangani logika terkait buku dalam aplikasi. Melalui metode-metode ini, dapat menampilkan, membuat, mengedit, dan menghapus data buku. Model BukuModel digunakan untuk berinteraksi dengan data buku dalam database. View yang terkait dengan masing-masing metode digunakan untuk menampilkan tampilan yang sesuai kepada pengguna.

4) Home.php

Kelas Home bertanggung jawab untuk menangani logika terkait halaman beranda atau halaman utama aplikasi. Dalam contoh ini, hanya ada satu metode yang mengembalikan view halaman login kepada pengguna. Namun, dapat menambahkan metode lainnya sesuai kebutuhan, tergantung pada logika bisnis aplikasi.

5) UserController.php

Dalam keseluruhan, kelas UserController bertanggung jawab untuk mengelola tampilan login, proses autentikasi pengguna, dan proses keluar dari aplikasi. Dalam metode login() dan logout(), sesi pengguna diatur dan dihapus sesuai dengan tindakan yang dilakukan oleh pengguna.

6) 2024-06-14-172721_Buku.php

Kode yang Anda berikan adalah sebuah migrasi database dengan nama Buku. Migrasi database digunakan untuk membuat dan mengatur struktur database

dalam aplikasi. Pada codingan tersebut, terdapat definisi struktur kolom-kolom yang akan ada di tabel buku dalam database.

7) 2024-06-14-172729_User.php

Pada codingan tersebut, terdapat migrasi untuk membuat tabel user dalam database. [9- 39] Codingan tersebut adalah bagian dari metode `up()` dalam migrasi untuk membuat tabel user dalam database. Pada bagian ini, dilakukan pengaturan struktur kolom menggunakan `$this->forge->addField()`.

8) Buku.php

seeder (pembenam) yang digunakan untuk mengisi data awal pada tabel buku dalam database. Seeder ini adalah bagian dari fitur database seeding pada CodeIgniter, yang memungkinkan kita untuk dengan mudah menambahkan data awal ke dalam tabel. Dalam metode `run()`, terdapat tiga pemanggilan `$this->db->table('buku')->insert()` yang digunakan untuk menyisipkan data buku ke dalam tabel buku. Setiap pemanggilan `insert()` menerima sebuah array yang berisi data untuk setiap kolom dalam tabel. Dengan menjalankan seeder ini, data buku-buku tersebut akan ditambahkan ke dalam tabel buku dalam database saat migrasi dijalankan atau saat seeder secara eksplisit dijalankan melalui command line interface (CLI) CodeIgniter. Hal ini memungkinkan kita untuk memiliki data awal yang konsisten dan dapat digunakan dalam pengembangan atau pengujian aplikasi.

9) UserSeeder.php

Codingan tersebut adalah sebuah seeder (pembenam) yang digunakan untuk mengisi data awal pada tabel user dalam database. Seeder ini adalah bagian dari fitur database seeding pada CodeIgniter, yang memungkinkan kita untuk dengan mudah menambahkan data awal ke dalam tabel. Dalam metode `run()`, terdapat satu pemanggilan `$this->db->table('user')->insert()` yang digunakan untuk menyisipkan data pengguna ke dalam tabel user. Pemanggilan `insert()` ini menerima sebuah array yang berisi data untuk setiap kolom dalam tabel. Dengan menjalankan seeder ini, data pengguna tersebut akan ditambahkan ke dalam tabel user dalam database saat migrasi dijalankan atau saat seeder secara eksplisit dijalankan melalui command line interface (CLI) CodeIgniter. Seeder ini dapat

membantu dalam membuat pengguna awal dengan cepat untuk keperluan pengembangan atau pengujian aplikasi.

10) BukuModel.php

Codingan tersebut adalah sebuah model dalam CodeIgniter yang merupakan representasi dari tabel "buku" dalam database. Model ini mengextends class CodeIgniter\Model yang menyediakan fitur-fitur dasar untuk melakukan operasi database seperti insert, update, delete, dan select. Dengan menggunakan model ini, dapat melakukan operasi database seperti insert, update, delete, dan select terhadap tabel "buku" dengan mudah. Model ini juga menyediakan fitur validasi data dan 43 callbacks yang dapat digunakan untuk memanipulasi data sebelum dan setelah operasi dilakukan.

11) UserModel.php

Codingan tersebut adalah sebuah model dalam CodeIgniter yang merepresentasikan tabel "user" dalam database. Model ini mengextends class CodeIgniter\Model yang menyediakan fitur-fitur dasar untuk melakukan operasi database seperti insert, update, delete, dan select. Model ini juga mendefinisikan properti-properti yang terkait dengan timestamps, validasi data, dan callbacks. Namun, dalam contoh codingan ini, properti properti tersebut tidak diatur dengan nilai tertentu. Ini berarti timestamps, validasi, dan callbacks tidak secara eksplisit didefinisikan atau digunakan dalam model ini.

D. Tautan Git

<https://github.com/MPashaNabeel/PRAK6-WEB2>