

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB II
MODUL 6**



CRUD DAN LOGIN PADA CODEIGNITER

Oleh:

Muhammad Aulia Rasyid NIM. 2210817210028

**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 : Muhammad Aulia Rasyid
NIM : 2210817210028

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ammarin Ihsan
NIM. 2010817210002

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

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL 1	6
A. Source Code	6
B. Output Program.....	22
C. Pembahasan	23
D. Tautan Git	26

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1	23
---	----

DAFTAR TABEL

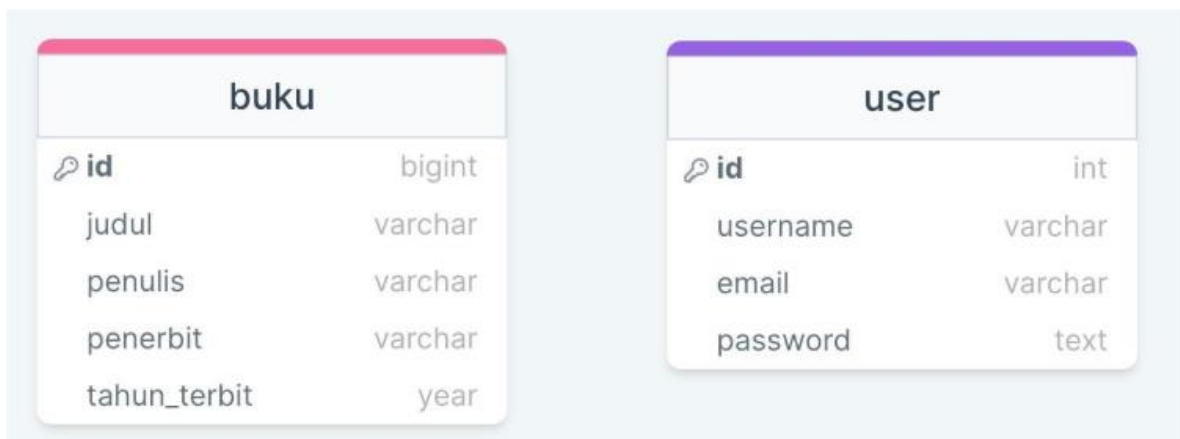
Tabel 1. Source Code Soal 1 Routes.php	6
Tabel 2. Source Code Soal 1 UserController.php	8
Tabel 3. Source Code Soal 1 BaseController.php	10
Tabel 4. Source Code Soal 1 BukuController.php	11
Tabel 5. Source Code Soal 1 AuthFilter.php.....	13
Tabel 6. Source Code Soal 1 View.php.....	13
Tabel 7. Source Code Soal 1 BukuModel.php	15
Tabel 8. Source Code Soal 1 UserModel.php	16
Tabel 9. Source Code Soal 1 prak6.sql.....	17

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

Routes.php

Tabel 1. Source Code Soal 1 Routes.php

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
19 // easy to create vulnerable apps
20 // where controller filters or CSRF protection are
21 // bypassed.
22 // If you don't want to define all routes, please use the
23 // Auto Routing (Improved).
24 // Set ` $autoRoutesImproved` to true in
25 // `app/Config/Feature.php` and set the following to true.
26 // $routes->setAutoRoute(false);
27
28 /*
29  * -----
30  * Route Definitions
31  * -----
32  */
33 // We get a performance increase by specifying the default
34 // route since we don't have to scan directories.
35 $routes->group('/', ["filter" => 'auth'], static function
36 ($routes) {
37     $routes->get('/', 'BukuController::index');
38     $routes->get('/buku', 'BukuController::index');
39     $routes->get('/buku/create',
40 'BukuController::create');
41     $routes->post('/buku/create',
42 'BukuController::store');

```

37	\$routes->get('/buku/(:num)/edit',
	'BukuController::edit/\$1');
38	\$routes->post('/buku/(:num)/edit',
	'BukuController::update/\$1');
39	\$routes->delete('/buku/(:num)/delete',
	'BukuController::delete/\$1');
40	\$routes->get('/logout', 'UserController::logout');
41	});
42	
43	
44	\$routes->get('/login', 'UserController::index');
45	\$routes->post('/login', 'UserController::login');
46	
47	/*
48	* -----

49	* Additional Routing
50	* -----

51	*
52	* There will often be times that you need additional
	routing and you
53	* need it to be able to override any defaults in this
	file. Environment
54	* based routes is one such time. require() additional
	route files here
55	* to make that happen.
56	*
57	* You will have access to the \$routes object within that
	file without
58	* needing to reload it.
59	*/
60	if (is_file(APPPATH . 'Config/' . ENVIRONMENT .
	'/Routes.php')) {
61	require APPPATH . 'Config/' . ENVIRONMENT .
	'/Routes.php';
62	}

UserController.php

Tabel 2. Source Code Soal 1 UserController.php

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


```

6 use App\Models\UserModel;
7
8 class UserController extends BaseController
9 {
10     public function index()
11     {
12         return view('login');
13     }
14
15     public function login()
16     {
17         $user = new UserModel();
18         $username = $this->request->getPost('username');
19         $pw = $this->request->getPost('password');
20         $dataUser = $user->where(['username' =>
21 $username])->orWhere(['email' => $username])->first();
22         if ($dataUser) {
23             if (password_verify($pw,
24 $dataUser['password'])) {
25                 session()->set([
26                     'username' => $dataUser['username'],
27                     'email' => $dataUser['email'],
28                     'logged_in' => true
29                 ]);
30                 return redirect()->to(base_url('buku'));
31             } else {
32                 session()->setFlashdata('pesan',
33 'Password salah');
34                 return redirect()-
35 >to(base_url('login'));
36             }
37         } else {
38             session()->setFlashdata('pesan', 'Username
39 atau email tidak ditemukan');
40             return redirect()->to(base_url('login'));
41         }
42     }
43
44     public function logout()
45     {
46         session()->destroy();
47         return redirect()->to(base_url('login'));
48     }
49 }

```

BaseController.php

Tabel 3. Source Code Soal 1 BaseController.php

```
1 <?php
2
3 namespace App\Controllers;
4
5 use CodeIgniter\Controller;
6 use CodeIgniter\HTTP\CLIRequest;
7 use CodeIgniter\HTTP\IncomingRequest;
8 use CodeIgniter\HTTP\RequestInterface;
9 use CodeIgniter\HTTP\ResponseInterface;
10 use Psr\Log\LoggerInterface;
11
12 /**
13  * Class BaseController
14  *
15  * BaseController provides a convenient place for loading
16  * components
17  * and performing functions that are needed by all your
18  * controllers.
19  * Extend this class in any new controllers:
20  *     class Home extends BaseController
21  *
22  * For security be sure to declare any new methods as
23  * protected or private.
24  */
25 abstract class BaseController extends Controller
26 {
27     /**
28      * Instance of the main Request object.
29      *
30      * @var CLIRequest|IncomingRequest
31      */
32     protected $request;
33
34     /**
35      * An array of helpers to be loaded automatically
36      * upon
37      * class instantiation. These helpers will be
38      * available
39      * to all other controllers that extend
40      * BaseController.
41      *
42      * @var array
43      */
44     protected $helpers = [];
```

39	
40	/**
41	* Be sure to declare properties for any property
42	fetch you initialized.
43	* The creation of dynamic property is deprecated in
44	PHP 8.2.
45	*/
46	/**
47	* Constructor.
48	*/
49	public function initController(RequestInterface
50	\$request, ResponseInterface \$response, LoggerInterface
51	\$logger)
52	{
53	// Do Not Edit This Line
54	parent::initController(\$request, \$response,
55	\$logger);
56	// Preload any models, libraries, etc, here.
57	// E.g.: \$this->session =
58	\Config\Services::session();
	}
	}

BukuController.php

Tabel 4. Source Code Soal 1 BukuController.php

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

```

17     public function create()
18     {
19         return view('buku/create');
20     }
21
22     public function store()
23     {
24         $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	}

AuthFilter.php

Tabel 5. Source Code Soal 1 AuthFilter.php

1	<?php
2	
3	namespace App\Filters;
4	
5	use CodeIgniter\Filters\FilterInterface;
6	use CodeIgniter\HTTP\RequestInterface;
7	use CodeIgniter\HTTP\ResponseInterface;
8	
9	class AuthFilter implements FilterInterface
10	{
11	public function before(RequestInterface \$request,
	\$args = null)
12	{
13	if(!session()->get('logged_in')){
14	session()->setFlashdata('pesan', 'Login
	terlebih dahulu!');
15	return redirect()->to(base_url('login'));
16	}
17	}
18	
19	public function after(RequestInterface \$request,
	\ResponseInterface \$response, \$args = null)
20	{
21	// Do something here
22	}
23	}

View.php

Tabel 6. Source Code Soal 1 View.php

1	<?php
2	
3	namespace Config;

```

4
5 use CodeIgniter\Config\View as BaseView;
6 use CodeIgniter\View\ViewDecoratorInterface;
7
8 class View extends BaseView
9 {
10     /**
11      * When false, the view method will clear the data
12      between each
13      * call. This keeps your data safe and ensures there
14      is no accidental
15      * leaking between calls, so you would need to
16      explicitly pass the data
17      * to each view. You might prefer to have the data
18      stick around between
19      * calls so that it is available to all views. If
20      that is the case,
21      * set $saveData to true.
22      *
23      * @var bool
24      */
25     public $saveData = true;
26
27     /**
28      * Parser Filters map a filter name with any PHP
29      callable. When the
30      * Parser prepares a variable for display, it will
31      chain it
32      * through the filters in the order defined,
33      inserting any parameters.
34      * To prevent potential abuse, all filters MUST be
35      defined here
36      * in order for them to be available for use within
37      the Parser.
38      *
39      * Examples:
40      * { title|esc(js) }
41      * { created_on|date(Y-m-d)|esc(attr) }
42      *
43      * @var array
44      */
45     public $filters = [];
46
47     /**
48      * Parser Plugins provide a way to extend the
49      functionality provided

```

```

39      * by the core Parser by creating aliases that will
be replaced with
40      * any callable. Can be single or tag pair.
41      *
42      * @var array
43      */
44      public $plugins = [];
45
46      /**
47      * View Decorators are class methods that will be run
in sequence to
48      * have a chance to alter the generated output just
prior to caching
49      * the results.
50      *
51      * All classes must implement
CodeIgniter\View\ViewDecoratorInterface
52      *
53      * @var class-string<ViewDecoratorInterface>[]
54      */
55      public array $decorators = [];
56  }

```

BukuModel.php

Tabel 7. Source Code Soal 1 BukuModel.php

```

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	}

UserModel.php

Tabel 8. Source Code Soal 1 UserModel.php

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	}

prak6.sql

Tabel 9. Source Code Soal 1 prak6.sql

1	-- phpMyAdmin SQL Dump
2	-- version 5.2.1
3	-- https://www.phpmyadmin.net/
4	--
5	-- Host: 127.0.0.1
6	-- Waktu pembuatan: 12 Jun 2024 pada 12.28
7	-- Versi server: 10.4.32-MariaDB
8	-- Versi PHP: 8.2.12
9	
10	SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11	START TRANSACTION;
12	SET time_zone = "+00:00";
13	
14	
15	/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

```

16  /*!40101 SET
    @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17  /*!40101 SET
    @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18  /*!40101 SET NAMES utf8mb4 */;
19
20  --
21  -- Database: `prak6`
22  --
23
24  -- -----
    ----
25
26  --
27  -- Struktur dari tabel `buku`
28  --
29
30  CREATE TABLE `buku` (
31    `id` int(5) UNSIGNED NOT NULL,
32    `judul` varchar(100) NOT NULL,
33    `penulis` varchar(100) NOT NULL,
34    `penerbit` varchar(100) NOT NULL,
35    `tahun_terbit` year(4) NOT NULL
36  ) ENGINE=InnoDB DEFAULT CHARSET=utf8
    COLLATE=utf8_general_ci;
37
38  --
39  -- Dumping data untuk tabel `buku`
40  --
41
42  INSERT INTO `buku` (`id`, `judul`, `penulis`,
    `penerbit`, `tahun_terbit`) VALUES
43  (1, 'Belajar Pemrograman Java', 'Petani Kode', 'PT
    Gramedia', '2020'),
44  (2, 'Belajar Pemrograman PHP', 'Petani Kode', 'PT
    Gramedia', '2020'),
45  (3, 'Belajar Pemrograman Python', 'Petani Kode', 'PT
    Gramedia', '2020');
46
47  -- -----
    ----
48
49  --
50  -- Struktur dari tabel `migrations`
51  --
52
53  CREATE TABLE `migrations` (

```

```

54 `id` bigint(20) UNSIGNED NOT NULL,
55 `version` varchar(255) NOT NULL,
56 `class` varchar(255) NOT NULL,
57 `group` varchar(255) NOT NULL,
58 `namespace` varchar(255) NOT NULL,
59 `time` int(11) NOT NULL,
60 `batch` int(11) UNSIGNED NOT NULL
61 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_general_ci;
62
63 --
64 -- Dumping data untuk tabel `migrations`
65 --
66
67 INSERT INTO `migrations` (`id`, `version`, `class`,
68 `group`, `namespace`, `time`, `batch`) VALUES
69 (1, '2024-06-10-145837',
  'App\\Database\\Migrations\\User', 'default', 'App',
  1718031677, 1),
70 (2, '2024-06-10-145842',
  'App\\Database\\Migrations\\Buku', 'default', 'App',
  1718031677, 1);
71
72 -----
73 ----
74
75 --
76 -- Struktur dari tabel `user`
77 --
78
79 CREATE TABLE `user` (
80 `id` int(5) UNSIGNED NOT NULL,
81 `username` varchar(100) NOT NULL,
82 `email` varchar(100) NOT NULL,
83 `password` text NOT NULL,
84 `created_at` datetime NOT NULL,
85 `updated_at` datetime NOT NULL,
86 `deleted_at` datetime NOT NULL
87 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_general_ci;
88
89 --
90 -- Dumping data untuk tabel `user`
91 --

```

```

91  INSERT INTO `user` (`id`, `username`, `email`,
    `password`, `created_at`, `updated_at`, `deleted_at`)
92  VALUES
    (1, 'admin', 'admin@mail.com',
    '$2y$10$2hDetJuThR9ZWahcot2O9O3OOrwpjvJMdx/VqqmjMrX5KTc
    RJRTi.', '0000-00-00 00:00:00', '0000-00-00 00:00:00',
    '0000-00-00 00:00:00');
93
94  --
95  -- Indexes for dumped tables
96  --
97
98  --
99  -- Indeks untuk tabel `buku`
100 --
101 ALTER TABLE `buku`
102     ADD PRIMARY KEY (`id`);
103
104 --
105 -- Indeks untuk tabel `migrations`
106 --
107 ALTER TABLE `migrations`
108     ADD PRIMARY KEY (`id`);
109
110 --
111 -- Indeks untuk tabel `user`
112 --
113 ALTER TABLE `user`
114     ADD PRIMARY KEY (`id`);
115
116 --
117 -- AUTO_INCREMENT untuk tabel yang dibuang
118 --
119
120 --
121 -- AUTO_INCREMENT untuk tabel `buku`
122 --
123 ALTER TABLE `buku`
124     MODIFY `id` int(5) UNSIGNED NOT NULL AUTO_INCREMENT,
    AUTO_INCREMENT=4;
125
126 --
127 -- AUTO_INCREMENT untuk tabel `migrations`
128 --
129 ALTER TABLE `migrations`
130     MODIFY `id` bigint(20) UNSIGNED NOT NULL
    AUTO_INCREMENT, AUTO_INCREMENT=3;

```

```

131
132 --
133 -- AUTO_INCREMENT untuk tabel `user`
134 --
135 ALTER TABLE `user`
136     MODIFY `id` int(5) UNSIGNED NOT NULL AUTO_INCREMENT,
137     AUTO_INCREMENT=2;
138
139 COMMIT;
140
141 /*!40101 SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
142 /*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
143 /*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

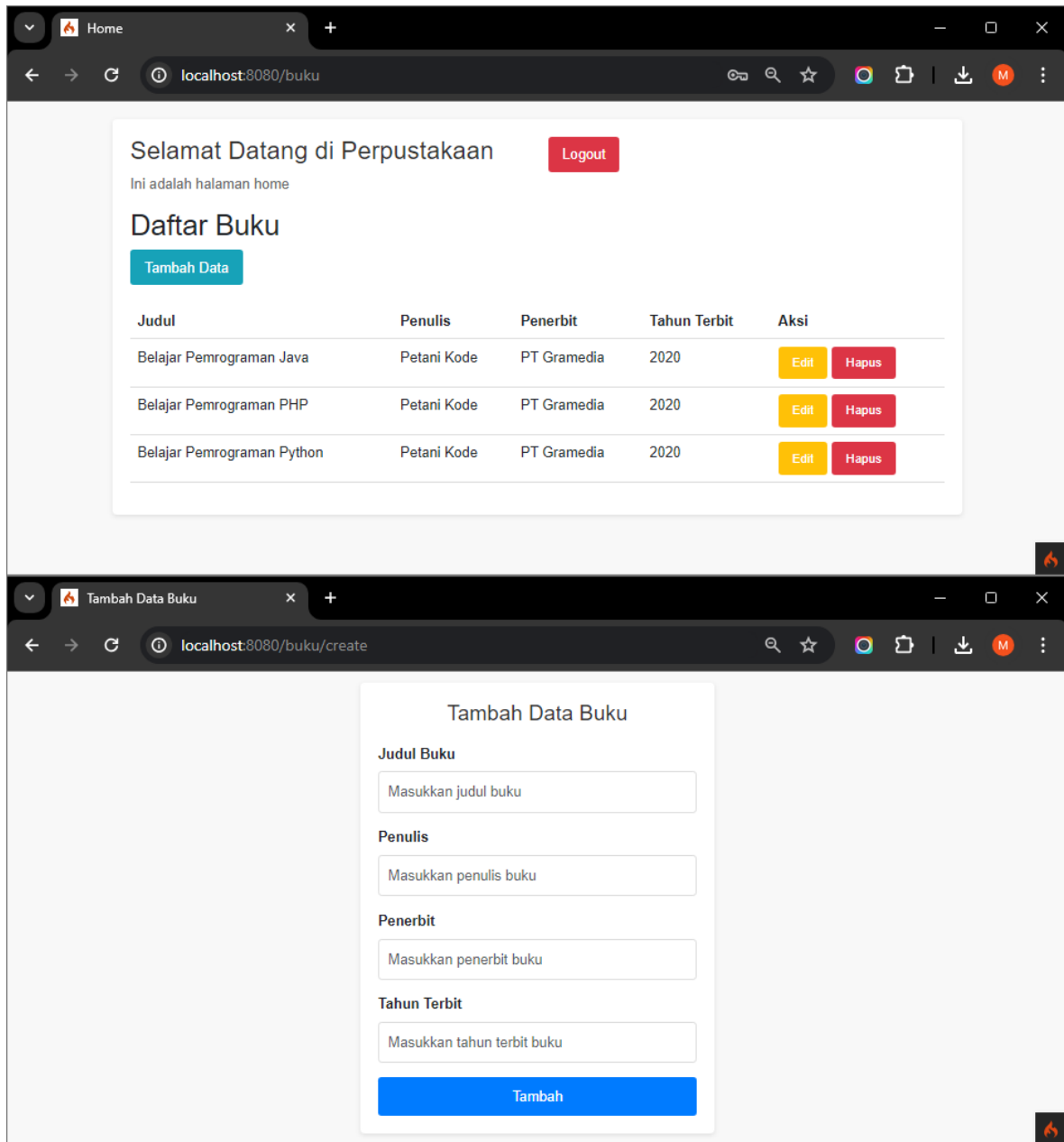
```

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 two input fields: "Username/Email address" and "Password". Below these fields is a blue button labeled "Sign in".

Bottom Screenshot: The login form is the same as the top one, but it now includes a yellow error message box above the input fields that reads "Login terlebih dahulu!". The "Sign in" button remains visible below the fields.



Gambar 1. Screenshot Hasil Jawaban Soal 1

C. Pembahasan

- Routes.php
 - Mengonfigurasi routing untuk aplikasi PHP dengan CodeIgniter 4. Pertama, namespace Config dan instance RouteCollection dibuat untuk mendefinisikan rute. Pengaturan router meliputi namespace default untuk controller (App\Controllers), controller default (Home), metode default (index), dan pengaturan agar garis miring di URL tidak diterjemahkan menjadi garis bawah di nama metode controller. Juga diatur tindakan untuk halaman tidak ditemukan (404).

- Peringatan tentang bahaya Auto Routing (Legacy) disertakan, dengan saran menggunakan Auto Routing (Improved) dan menonaktifkan auto routing.
- Rute-rute dengan filter auth hanya dapat diakses oleh pengguna terautentikasi. Rute ini mencakup halaman utama (/), daftar buku (/buku), pembuatan (/buku/create), penyimpanan (/buku/create), pengeditan (/buku/(:num)/edit), pembaruan (/buku/(:num)/edit), penghapusan (/buku/(:num)/delete), dan logout (/logout).
- Rute login didefinisikan secara terpisah untuk menampilkan halaman login (/login) dan memproses login (/login). Terakhir, file rute tambahan spesifik untuk environment dimuat jika ada, untuk menambahkan atau mengoverride rute yang ada.
- UserController.php
 - Bagian dari controller UserController dalam aplikasi CodeIgniter 4 yang menangani proses login dan logout pengguna. Pada metode index, tampilan login dikembalikan. Pada metode login, data pengguna diambil dari model UserModel berdasarkan username atau email yang dimasukkan. Jika pengguna ditemukan dan password yang dimasukkan cocok dengan yang ada di database, sesi pengguna diatur dengan data username, email, dan status login, kemudian diarahkan ke halaman buku. Jika password salah, pesan kesalahan disimpan dalam sesi dan pengguna diarahkan kembali ke halaman login. Jika username atau email tidak ditemukan, pesan kesalahan serupa juga disimpan dalam sesi dan pengguna diarahkan kembali ke halaman login. Metode logout menghancurkan sesi dan mengarahkan pengguna kembali ke halaman login.
- BaseController.php
 - Mendefinisikan BaseController dalam aplikasi CodeIgniter 4, yang merupakan kelas abstrak yang mewarisi dari Controller. Kelas ini menyediakan tempat yang nyaman untuk memuat komponen dan melakukan fungsi yang dibutuhkan oleh semua controller lain. Variabel request dideklarasikan sebagai instance dari objek utama Request, baik itu CLIRequest atau IncomingRequest. Properti helpers adalah array dari helper yang akan dimuat secara otomatis saat kelas diinstansiasi, dan tersedia untuk semua controller yang memperluas BaseController. Kelas ini juga mendeklarasikan properti yang diinisialisasi untuk menghindari penggunaan properti dinamis yang tidak dianjurkan di PHP 8.2. Metode initController adalah konstruktor yang memanggil konstruktor induk dan dapat digunakan untuk memuat model, library, atau komponen lain yang diperlukan. Misalnya, sesi dapat dimuat dengan `$this->session = \Config\Services::session();`.
- BukuController.php
 - Mendefinisikan BukuController dalam aplikasi CodeIgniter 4, yang bertanggung jawab untuk mengelola data buku. Metode index menampilkan semua buku dengan mengambil data dari BukuModel dan mengirimkannya ke tampilan buku/index. Metode create mengarahkan ke tampilan formulir pembuatan buku baru. Metode store menyimpan data buku baru yang diterima

dari formulir ke dalam database melalui BukuModel dan kemudian mengarahkan kembali ke daftar buku. Metode edit menampilkan formulir pengeditan untuk buku berdasarkan ID yang diterima, dan data buku diambil dari BukuModel lalu dikirim ke tampilan buku/edit. Metode update memperbarui data buku berdasarkan ID dengan data yang diterima dari formulir dan kemudian mengarahkan kembali ke daftar buku. Metode delete menghapus buku berdasarkan ID dari database melalui BukuModel dan mengarahkan kembali ke daftar buku.

- AuthFilter.php
 - Mendefinisikan AuthFilter, sebuah filter dalam aplikasi CodeIgniter 4 yang mengimplementasikan FilterInterface. Filter ini digunakan untuk memeriksa apakah pengguna sudah login sebelum mengakses halaman tertentu. Metode before dijalankan sebelum request diproses. Jika pengguna belum login (diperiksa dengan session()->get('logged_in')), filter akan menyetel pesan flash 'Login terlebih dahulu!' dan mengarahkan pengguna ke halaman login. Metode after dijalankan setelah request diproses, tetapi dalam contoh ini tidak melakukan apa-apa. Filter ini memastikan bahwa hanya pengguna yang terautentikasi yang dapat mengakses halaman-halaman tertentu dalam aplikasi.
- View.php
 - Mendefinisikan kelas View yang memperluas BaseView dalam namespace Config pada aplikasi CodeIgniter 4. Kelas ini mengonfigurasi beberapa aspek tampilan aplikasi. Properti saveData diset ke true, yang berarti data akan disimpan antara pemanggilan metode view sehingga tersedia untuk semua tampilan. Properti filters adalah array yang memetakan nama filter dengan callable PHP, digunakan oleh Parser untuk memproses variabel sebelum ditampilkan, untuk mencegah penyalahgunaan, semua filter harus didefinisikan di sini. Properti plugins adalah array yang memungkinkan ekstensi fungsionalitas Parser dengan membuat alias yang digantikan oleh callable, baik dalam bentuk tunggal atau tag pair. Terakhir, properti decorators adalah array dari kelas yang mengimplementasikan ViewDecoratorInterface, yang akan dijalankan secara berurutan untuk mengubah output yang dihasilkan sebelum hasilnya di-cache.
- BukuModel.php
 - Definisi dari model BukuModel dalam aplikasi CodeIgniter 4, yang mengatur interaksi dengan tabel buku dalam database. Properti-properti yang dikonfigurasi termasuk pengaturan koneksi database (\$DBGroup), nama tabel (\$table), kunci primer (\$primaryKey), penggunaan auto increment untuk kunci primer (\$useAutoIncrement), tipe data yang dihasilkan (\$returnType), serta daftar field yang diizinkan untuk dimasukkan atau diubah (\$allowedFields).
 - Selain itu, terdapat properti untuk mengatur penggunaan soft deletes (\$useSoftDeletes), proteksi field (\$protectFields), dan pengaturan tanggal

(\$useTimestamps, \$dateFormat, \$createdField, \$updatedField, \$deletedField). Properti untuk validasi data (\$validationRules, \$validationMessages, \$skipValidation, \$cleanValidationRules) dan callback-method (\$allowCallbacks, \$beforeInsert, \$afterInsert, \$beforeUpdate, \$afterUpdate, \$beforeFind, \$afterFind, \$beforeDelete, \$afterDelete) juga ditentukan. Model ini bertindak sebagai perantara antara controller dan database, memungkinkan operasi CRUD pada tabel buku.

- UserModel.php
 - Definisi dari model UserModel dalam aplikasi CodeIgniter 4. Model ini bertanggung jawab untuk berinteraksi dengan tabel user dalam database. Properti-properti yang dikonfigurasi meliputi pengaturan koneksi database (\$DBGroup), nama tabel (\$table), kunci primer (\$primaryKey), penggunaan auto increment untuk kunci primer (\$useAutoIncrement), tipe data yang dihasilkan (\$returnType), serta pengaturan penggunaan soft deletes (\$useSoftDeletes), proteksi field (\$protectFields), dan tanggal (\$useTimestamps, \$dateFormat, \$createdField, \$updatedField, \$deletedField).
 - Selain itu, terdapat properti untuk validasi data (\$validationRules, \$validationMessages, \$skipValidation, \$cleanValidationRules) dan callback-method (\$allowCallbacks, \$beforeInsert, \$afterInsert, \$beforeUpdate, \$afterUpdate, \$beforeFind, \$afterFind, \$beforeDelete, \$afterDelete). Meskipun dalam contoh ini, properti \$allowedFields tidak diisi, model ini masih dapat digunakan untuk operasi CRUD dasar pada tabel user, seperti pengambilan data, penyimpanan, dan penghapusan.
- Prak6.sql
 - Merupakan dump SQL yang berisi definisi struktur tabel dan data untuk database prak6 yang digunakan dalam aplikasi. Tabel-tabel yang dijelaskan meliputi buku, migrations, dan user. Setiap tabel memiliki struktur yang didefinisikan, termasuk nama kolom, tipe data, atribut, serta indeks yang diterapkan.
 - Selain itu, data juga dimasukkan ke dalam tabel-tabel tersebut menggunakan pernyataan INSERT INTO, memberikan contoh data yang disimpan dalam database. Pada tabel user, terdapat contoh pengguna dengan username 'admin', alamat email 'admin@mail.com', dan password yang telah di-hash. Setiap tabel juga memiliki primary key yang sesuai, dan beberapa di antaranya memiliki atribut AUTO_INCREMENT untuk memastikan nilai yang unik secara otomatis.

D. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/rasszmhmmd/Web-2-Prak-601.git>