

**LAPORAN PRAKTIKUM  
PEMROGRAMAN WEB II  
MODUL 6**



**CRUD DAN LOGIN PADA CODEIGNITER**

**Oleh:**

**Nursyahna Puteri**

**NIM. 2210817120007**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS LAMBUNG MANGKURAT  
JUNI 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 : Nursyahna Puteri  
NIM : 2210817120007

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

|  |          |
|--|----------|
| <b>LEMBAR PENGESAHAN.....</b>                      | <b>2</b> |
| <b>DAFTAR ISI .....</b>                            | <b>3</b> |
| <b>DAFTAR GAMBAR .....</b>                         | <b>6</b> |
| <b>DAFTAR TABEL .....</b>                          | <b>7</b> |
| <b>SOAL PRAKTIKUM.....</b>                         | <b>8</b> |
| A. Source Code <i>Routes.php</i> .....             | 9        |
| B. Pembahasan <i>Routes.php</i> .....              | 10       |
| C. Source Code <i>BaseController.php</i> .....     | 12       |
| D. Pembahasan <i>BaseController.php</i> .....      | 14       |
| E. Source Code <i>BukuController.php</i> .....     | 16       |
| F. Pembahasan <i>BukuController.php</i> .....      | 18       |
| G. Source Code <i>Home.php</i> .....               | 22       |
| H. Pembahasan <i>Home.php</i> .....                | 23       |
| I. Source Code <i>RegisterController.php</i> ..... | 24       |
| J. Pembahasan <i>RegsiterController.php</i> .....  | 26       |
| K. Source <i>UserController.php</i> .....          | 29       |
| L. Pembahasan <i>UserController.php</i> .....      | 31       |
| M. Source Code <i>AuthFilter.php</i> .....         | 34       |
| N. Pembahasan <i>AuthFilter.php</i> .....          | 35       |
| O. Source Code <i>Create.php</i> .....             | 36       |
| P. Ouput Program <i>create.php</i> .....           | 41       |
| Q. Pembahasan <i>Create.php</i> .....              | 42       |
| R. Source Code <i>edit.php</i> .....               | 50       |

|     |   |     |
|-----|---|-----|
| S.  | Ouput Program <i>edit.php</i> .....             | 56  |
| T.  | Pembahasan <i>edit.php</i> .....                | 56  |
| U.  | Source Code <i>index.php</i> .....              | 63  |
| V.  | Output Program <i>index.php</i> .....           | 68  |
| W.  | Pembahasan <i>index.php</i> .....               | 68  |
| X.  | Source Code <i>login.php</i> .....              | 75  |
| Y.  | Output Program <i>login.php</i> .....           | 78  |
| Z.  | Pembahasan <i>login.php</i> .....               | 78  |
| AA. | Source Code <i>register.php</i> .....           | 84  |
| BB. | Ouput Program <i>register.php</i> .....         | 89  |
| CC. | Pembahasan <i>register.php</i> .....            | 90  |
| DD. | Source Code <i>welcome_mesaage.php</i> .....    | 96  |
| EE. | Output Program <i>welcome_mesaage.php</i> ..... | 98  |
| FF. | Pembahasan <i>welcome_mesaage.php</i> .....     | 99  |
| GG. | Source Code <i>BukuModel.php</i> .....          | 102 |
| HH. | Pembahasan <i>BukuModel.php</i> .....           | 104 |
| II. | Source Code <i>UserModel.php</i> .....          | 106 |
| JJ. | Pembahasan <i>UserModel.php</i> .....           | 109 |
| KK. | Source Code <i>UserSeeder.php</i> .....         | 113 |
| LL. | Pembahasan <i>UserSeeder.php</i> .....          | 114 |
| MM. | Source Code <i>2024-06-12-202936_Buku</i> ..... | 115 |
| NN. | Pembahasan <i>2024-06-12-202936_Buku</i> .....  | 116 |
| OO. | Source Code <i>2024-06-12-202940_User</i> ..... | 119 |
| PP. | Pembahasan <i>2024-06-12-202940_User</i> .....  | 121 |
| QQ. | Source Code <i>buku.sql</i> .....               | 123 |

|     |                          |     |
|-----|--------------------------|-----|
| RR. | Pembahasan buku.sql..... | 127 |
| SS. | Tautan Git .....         | 127 |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 1. Screenshot Hasil Jawaban Modul 6 create.php .....          | 42 |
| Gambar 2. Screenshot Hasil Jawaban Modul 6 edit.php.....             | 56 |
| Gambar 3. Screenshot Hasil Jawaban Modul 6 index.php .....           | 68 |
| Gambar 4. Screenshot Hasil Jawaban Modul 6 login.php.....            | 78 |
| Gambar 5. Screenshot Hasil Jawaban Modul 6 register.php .....        | 89 |
| Gambar 6. Screenshot Hasil Jawaban Modul 6 welcome_message.php ..... | 99 |

## DAFTAR TABEL

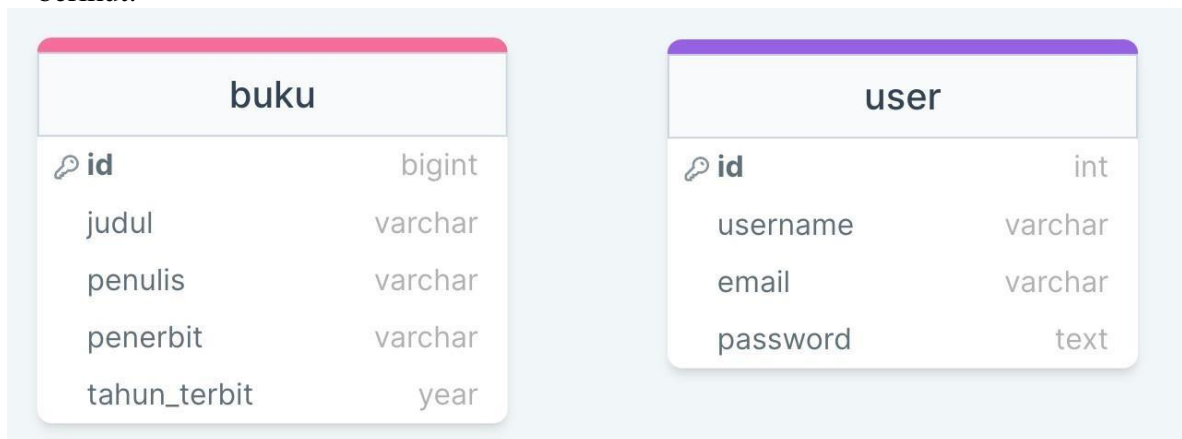
|  |     |
|--|-----|
| Table 1. Source Code Modul 6 Routes.php.....               | 9   |
| Table 2. Source Code Modul 6 BaseController.php.....       | 12  |
| Table 3. Source Code Modul 6 BukuController.php.....       | 16  |
| Table 4. Source Code Modul 6 Home.php .....                | 22  |
| Table 5. Source Code Modul 6 RegisterController.php .....  | 24  |
| Table 6. Source Code Modul 6 UserController.php .....      | 29  |
| Table 7. Source Code Modul 6 AuthFilter.php .....          | 34  |
| Table 8. Source Code Modul 6 Create.php .....              | 36  |
| Table 9. Source Code Modul 6 edit.php.....                 | 50  |
| Table 10. Source Code Modul 6 index.php .....              | 63  |
| Table 11. Source Code Modul 6 login.php.....               | 75  |
| Table 12. Source Code Modul 6 register.php .....           | 84  |
| Table 13. Source Code Modul 6 welcome_message.php.....     | 96  |
| Table 14. Source Code Modul 6 BukuModel.php.....           | 102 |
| Table 15. Source Code Modul 6 UserModel.php .....          | 106 |
| Table 16. Source Code Modul 6 UserSeeder.php .....         | 113 |
| Table 17. Source Code Modul 6 2024-06-12-202936_Buku ..... | 115 |
| Table 18. Source Code Modul 6 2024-06-12-202940_User ..... | 119 |
| Table 19. Source Code Modul 6 buku.sql .....               | 123 |

## SOAL PRAKTIKUM

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 dengan nama folder github Github: PRAK601



## A. Source Code *Routes.php*

Table 1. Source Code Modul 6 *Routes.php*

|    |   |
|----|---|
| 1  | <?php   |
| 2  |   |
| 3  | namespace Config;   |
| 4  |   |
| 5  | \$routes = Services::routes();  |
| 6  |   |
| 7  | \$routes->setDefaultNamespace('App\Controllers');                         |
| 8  | \$routes->setDefaultController('Home');                                   |
| 9  | \$routes->setDefaultMethod('index');                                      |
| 10 | \$routes->setTranslateURIDashes(false);                                   |
| 11 | \$routes->set404Override();   |
| 12 |   |
| 13 | \$routes->get('/', 'Home::index', ['filter' => 'auth']);                  |
| 14 | \$routes->get('/buku', 'BukuController::index');                          |
| 15 | \$routes->get('/buku/create', 'BukuController::create');                  |
| 16 | \$routes->post('/buku/create', 'BukuController::store');                  |
| 17 | \$routes->get('/buku/(:num)/edit',<br>'BukuController::edit/\$1');        |
| 18 | \$routes->post('/buku/(:num)/edit',<br>'BukuController::update/\$1');     |
| 19 | \$routes->delete('/buku/(:num)/delete',<br>'BukuController::delete/\$1'); |
| 20 |   |
| 21 | \$routes->get('/login', 'UserController::index');                         |
| 22 |   |
| 23 | \$routes->get('/register', 'RegisterController::index');                  |
| 24 | \$routes->post('/register', 'RegisterController::store');                 |
| 25 |   |
| 26 | \$routes->post('/login', 'UserController::login');                        |

|    |   |
|----|---|
| 27 | <code>\$routes-&gt;get('/logout', 'UserController::logout');</code> |
| 28 |   |
| 29 | <code>if (is_file(APPPATH . 'Config/' . ENVIRONMENT .</code>        |
|    | <code>'/Routes.php')) {</code>                                      |
| 30 | <code>    require APPPATH . 'Config/' . ENVIRONMENT .</code>        |
|    | <code>    '/Routes.php';</code>                                     |
| 31 | <code>}</code>  |

## B. Pembahasan *Routes.php*

Pada baris [1] [3] [5], syntax

```
<?php

namespace Config;

$routes = Services::routes();
```

pada baris-baris tersebut digunakan untuk mendefinisikan namespace Config dan menginisialisasi variabel \$routes dengan layanan routes() dari CodeIgniter. Tujuan dari penggunaan objek '\$routes' tersebut adalah untuk mengatur atau routing permintaan. Pada penggunaan kode ini merupakan awal dalam memberikan pengaturan rute yang terorganisir dengan baik untuk aplikasi web berbasis CodeIgniter.

Pada baris [7] [8] [9], syntax

```
$routes->setDefaultNamespace('App\Controllers');

$routes->setDefaultController('Home');

$routes->setDefaultMethod('index');
```

Pada baris-baris tersebut digunakan untuk mengatur nilai default untuk namespace kontroler (App\Controllers), kontroler default (Home), dan metode default (index). Disaat tidak ada rute yang cocok, aplikasi web akan mengarahkan ke Home controller dan metode index.

Pada baris [10] [11], syntax

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

```
$routes->set404Override();
```

pada baris-baris tersebut terdapat beberapa fungsi dari penggunaan kode-kode yang ada, penggunaan `$routes->setTranslateURIDashes(false);` digunakan untuk mengatur agar karakter dash (-) dalam URI tidak diartikan sebagai pemisah segmen. Lalu penggunaan `$routes->set404Override();` untuk mengaktifkan penggantian halaman 404 default CodeIgniter dengan penggunaan kustom.

Pada baris [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27], syntax

```
$routes->get('/', 'Home::index', ['filter' => 'auth']);

$routes->get('/buku', 'BukuController::index');

$routes->get('/buku/create', 'BukuController::create');

$routes->post('/buku/create', 'BukuController::store');

$routes->get('/buku/(:num)/edit', 'BukuController::edit/$1');

$routes->post('/buku/(:num)/edit',
'BukuController::update/$1');

$routes->delete('/buku/(:num)/delete',
'BukuController::delete/$1');

$routes->get('/login', 'UserController::index');

$routes->get('/register', 'RegisterController::index');

$routes->post('/register', 'RegisterController::store');

$routes->post('/login', 'UserController::login');

$routes->get('/logout', 'UserController::logout');
```

pada baris-baris tersebut mendefinisikan beberapa rute yang tersedia dalam aplikasi web yang dibuat. Pada aplikais tersbeut terdapat rute untuk halaman Home yang diberlengkapi dengan filter autentikasi, memastikan hanya pengguna yang terautentikasi yang dapat mengaksesnya. Selanjutnya, rute-rute untuk BukuController disediakan, termasuk rute index untuk menampilkan daftar buku, create untuk menampilkan formulir pembuatan buku, store

untuk menyimpan data buku baru, edit untuk menampilkan formulir edit buku, update untuk memperbarui informasi buku yang ada, dan delete untuk menghapus buku menggunakan metode HTTP yang sesuai seperti GET, POST, dan DELETE. Selanjutnya terdapat juga rute untuk UserController, yang memiliki rute index untuk menampilkan profil pengguna atau daftar pengguna, login untuk proses autentikasi pengguna, dan logout untuk proses keluar dari sesi pengguna. Memiliki juga rute-rute untuk RegisterController, termasuk rute index untuk menampilkan formulir pendaftaran pengguna baru untuk menyimpan data pendaftaran tersebut.

Pada baris [29] [30] [31], syntax

```
if (is_file(APPPATH . 'Config/' . ENVIRONMENT . '/Routes.php'))
{
    require APPPATH . 'Config/' . ENVIRONMENT . '/Routes.php';
}
```

Penggunaan baris-baris tersebut untuk memeriksa apakah ada file khusus rute untuk lingkungan tertentu. Apabila ada, file tersebut dimuat untuk menggantikan atau menambahkan rute yang ada.

### C. Source Code *BaseController.php*

Table 2. Source Code Modul 6 *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 abstract class BaseController extends Controller
13 {
14     /**
15      * Instance of the main Request object.
16      *
17      * @var CLIRequest|IncomingRequest
18      */
19     protected $request;
20
21     /**
22      * An array of helpers to be loaded automatically
23      upon
24      * class instantiation. These helpers will be
25      available
26      * to all other controllers that extend
27      BaseController.
28      *
29      * @var array
30      */
31     protected $helpers = [];
32
33     public function initController(RequestInterface
34     $request, ResponseInterface $response,
35     LoggerInterface $logger)
36     {
37         parent::initController($request, $response,
38         $logger);
39     }

```

|    |   |
|----|---|
| 36 | } |
|----|---|

#### D. Pembahasan *BaseController.php*

Pada baris [1] [2] [3] [4] [5] [6] [7] [8] [9] [10], syntax

```
<?php

namespace App\Controllers;

use CodeIgniter\Controller;

use CodeIgniter\HTTP\CLIRequest;

use CodeIgniter\HTTP\IncomingRequest;

use CodeIgniter\HTTP\RequestInterface;

use CodeIgniter\HTTP\ResponseInterface;

use Psr\Log\LoggerInterface;
```

penggunaan baris-baris tersebut terdapat untuk mendefinisikan namespace `App\Controllers` yang menunjukkan bahwa kelas `BaseController` berada dalam direktori `App\Controllers`. Pada baris tersebut juga terdapat use statements yang digunakan untuk mengimpor kelas-kelas yang diperlukan dari framework CodeIgniter dan PSR-3 Logger untuk digunakan dalam kelas `BaseController`.

Pada baris [12], syntax `abstract class BaseController extends Controller` digunakan untuk menyatakan bahwa `BaseController` adalah kelas abstrak yang mengimplementasikan `Controller` dari CodeIgniter. Hal ini berarti semua kontroler dalam aplikasi yang meng-extend `BaseController` akan mewarisi fungsionalitas dasar dari `Controller`.

Pada baris [13] [14] [15] [16] [17] [18] [19], syntax

```
{

    /**
```

```

    * Instance of the main Request object.
    *
    * @var CLIRequest|IncomingRequest
    */
    protected $request;

```

penggunaan properti tersebut untuk menyimpan instance dari objek utama Request. Hal ini dapat berupa CLIRequest yang digunakan untuk permintaan CLI atau IncomingRequest yang digunakan untuk permintaan HTTP.

Pada baris [21] [22] [23] [24] [25] [26] [27] [28], syntax

```

/**
    * An array of helpers to be loaded automatically upon
    * class instantiation. These helpers will be available
    * to all other controllers that extend BaseController.
    *
    * @var array
    */
    protected $helpers = [];

```

penggunaan properti tersebut yang berisi daftar helpers atau bantuan yang akan dimuat secara otomatis saat kelas BaseController di-instanstiasi. Helpers ini akan tersedia untuk semua kontroler lain yang meng-extend BaseController.

Pada baris [31] [32] [33] [34] [35] [36], syntax

```

public function initController(RequestInterface $request,
    ResponseInterface $response, LoggerInterface $logger)
    {

```

```

        parent::initController($request, $response, $logger);

    }

}

```

pada baris-baris tersebut merupakan metode inisialisasi yang akan dipanggil secara otomatis oleh framework saat sebuah kontroler di-instanstiasi. Metode tersebut menerima tiga parameter, yaitu `$request` atau objek `Request`, `$response` atau objek `Response`, dan `$logger` atau objek `Logger`. Lalu pada baris tersebut juga memanggil metode `initController()` dari kelas induk (`Controller`) menggunakan `parent::initController()` untuk menginisialisasi kelas `Controller`.

#### E. Source Code *BukuController.php*

Table 3. Source Code Modul 6 *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-
29 >getPost('penulis'),
29             'penerbit' => $this->request-
30 >getPost('penerbit'),
30             'tahun_terbit' => $this->request-
31 >getPost('tahun_terbit'),
31         ]);
32
33         return redirect()->to('/buku');
34     }
35
36     public function edit($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 }

```

## F. Pembahasan BukuController.php

Pada baris [1] [2] [3] [4] [5] [6], syntax

```
<?php
```

```
namespace App\Controllers;
```

```
use App\Controllers\BaseController;  
  
use App\Models\BukuModel;
```

pada baris-baris terdapat beberap apenggunaan, terdapat penggunaan untuk mendefinisikan namespace App\Controllers, yang menunjukkan bahwa kelas BukuController berada dalam direktori App\Controllers. Selanjutnya, terdapat penggunaan kode mengimpor BaseController untuk meng-extend dari kelas dasar yang telah dijelaskan sebelumnya. Lalu terdapat juga kode untuk mengimpor BukuModel, yang merupakan model untuk entitas buku. Model ini bertanggung jawab untuk mengakses dan memanipulasi data dari tabel buku dalam database.

Pada baris [8], syntax class BukuController extends BaseController memberikan keterangan bahwa BukuController meng-extend atau mewarisi fungsionalitas dari BaseController.

Pada abris [10] [11] [12] [13] [14] [15], syntax public function index()

```
{  
  
    return view('buku/index', [  
  
        'buku' => (new BukuModel())->findAll()  
  
    ]);  
  
}
```

penggunaan baris-baris tersebut untuk mengatasi permintaan untuk halaman indeks buku dengan menggunakan BukuModel untuk mengambil semua data buku menggunakan metode findAll(). Data yang diambil kemudian dikirimkan ke view buku/index, di mana view akan menerima data tersebut dalam variabel \$buku untuk ditampilkan kepada pengguna.

Pada baris [17] [18] [19 ] [20], syntax

```
public function create()  
  
    {  
  
        return view('buku/create');  
  
    }
```

Penggunaan baris-baris tersebut memiliki tanggung jawab atas permintaan untuk menampilkan formulir pembuatan buku dengan mengembalikan view buku/create. View ini berisi formulir kosong yang memungkinkan pengguna untuk menambahkan informasi buku baru sesuai dengan kebutuhan.

Pada baris [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34], syntax

```
public function store()

{

    $buku = new BukuModel();

    $buku->insert([

        'judul' => $this->request->getPost('judul'),

        'penulis' => $this->request->getPost('penulis'),

        'penerbit' => $this->request->getPost('penerbit'),

        'tahun_terbit' => $this->request->getPost('tahun_terbit'),

    ]);

    return redirect()->to('/buku');

}
```

Penggunaan baris-baris tersebut digunakan untuk mengatasi permintaan untuk menyimpan data buku baru yang dikirimkan melalui formulir. Dengan menggunakan kode-kode tersebut akan dilakukan pembuatan instance BukuModel untuk memanipulasi data buku. Kemudian, menggunakan metode insert() dari BukuModel untuk menambahkan data buku baru ke dalam database. Setelah data berhasil disimpan, pengguna langsung diarahkan kembali ke halaman indeks buku, memastikan bahwa proses penambahan buku berjalan lancar.

Pada baris [37] [38] [39] [40] [41], syntax

```
public function edit($id)
```

```

{
    return view('buku/edit', [
        'buku' => (new BukuModel())->find($id)
    ]);
}

```

penggunaan baris-baris tersebut digunakan untuk menangani permintaan untuk menampilkan formulir edit buku berdasarkan ID tertentu dengan menggunakan BukuModel. Dengan menggunakan kode-kode tersebut akan dilakukan pencarian buku menggunakan metode find(\$id) dari BukuModel untuk mengambil data buku yang sesuai dengan ID yang diminta. Setelah data buku ditemukan, informasi buku tersebut kemudian dikirimkan ke view buku/edit.

Pada baris [43] [44] [45] [46] [47] [48] [49] [50] [1] [52] [53] [54], syntax

```

public function update($id)
{
    $buku = new BukuModel();

    $buku->update($id, [
        'judul' => $this->request->getPost('judul'),
        'penulis' => $this->request->getPost('penulis'),
        'penerbit' => $this->request->getPost('penerbit'),
        'tahun_terbit' => $this->request->getPost('tahun_terbit'),
    ]);
}

```

```

        return redirect()->to('/buku');
    }

```

penggunaan baris-baris tersebut digunakan untuk mengatasi permintaan untuk menyimpan perubahan pada buku dengan ID tertentu yang dilakukan melalui formulir edit. Proses dimulai dengan menggunakan BukuModel untuk memanggil metode update() dengan parameter ID buku yang ingin diperbarui dan data yang telah diubah. Setelah perubahan berhasil diterapkan pada database, pengguna akan diarahkan kembali ke halaman indeks buku.

Pada baris [57] [58] [59] [60] [61] [62] [63] [64] [65], syntax

```

public function delete($id)
{
    $buku = new BukuModel();

    $buku->delete($id);

    return redirect()->to('/buku');
}
}

```

penggunaan baris-baris tersebut digunakan untuk mengatasi permintaan untuk menghapus buku berdasarkan ID tertentu dengan menggunakan BukuModel. Proses yang akan dilakukan dengan kode-kode tersebut adalah akan memanggil metode delete() dari BukuModel dengan parameter ID buku yang akan dihapus. Setelah buku berhasil dihapus dari database, pengguna akan diarahkan kembali ke halaman indeks buku.

## G. Source Code *Home.php*

Table 4. Source Code Modul 6 *Home.php*

|    |                                   |
|----|-----------------------------------|
| 1  | <?php                             |
| 2  |                                   |
| 3  | namespace App\Controllers;        |
| 4  |                                   |
| 5  | class Home extends BaseController |
| 6  | {                                 |
| 7  | public function index()           |
| 8  | {                                 |
| 9  | return view('welcome_message');   |
| 10 | }                                 |
| 11 | }                                 |

## H. Pembahasan *Home.php*

Pada baris [1] [2] [3], syntax

```
<?php
```

```
namespace App\Controllers;
```

pada baris tersebut memiliki fungsi untuk mendefinisikan namespace App\Controllers. Sehingga menunjukkan bahwa kelas Home berada di dalam direktori App\Controllers.

Pada baris [5], syntax `class Home extends BaseController` memiliki fungsi untuk mendeklarasikan kelas Home yang meng-extend atau mewarisi fungsionalitas dari BaseController. BaseController itu sendiri merupakan kelas dasar yang memuat fungsionalitas dasar yang dibutuhkan oleh semua kontroler dalam aplikasi.

Pada baris [6] [7] [8] [9] [10] [11], syntax

```
{
    public function index()
    {
        return view('welcome_message');
```

```

    }

}

```

penggunaan baris-baris tersebut berisikan metode `index()`, metode `index` adalah metode publik yang menangani permintaan untuk halaman utama atau halaman indeks dari kontroler `Home`. Metode ini menggunakan fungsi `view()` untuk memuat view yang disebut `'welcome_message'`. View ini dapat berisi markup HTML dan elemen-elemen tampilan halaman depan aplikasi, berfungsi sebagai halaman selamat datang yang diperlihatkan kepada pengguna saat mereka mengunjungi aplikasi atau situs web tersebut.

## I. Source Code *RegisterController.php*

Table 5. Source Code Modul 6 *RegisterController.php*

|    |  |
|----|--|
| 1  | <code>&lt;?php</code>  |
| 2  |  |
| 3  | <code>namespace App\Controllers;</code>                      |
| 4  |  |
| 5  | <code>use App\Models\UserModel;</code>                       |
| 6  |  |
| 7  | <code>class RegisterController extends BaseController</code> |
| 8  | <code>{</code>   |
| 9  | <code>    protected \$model;</code>                          |
| 10 |  |
| 11 | <code>    public function __construct()</code>               |
| 12 | <code>    {</code>   |
| 13 | <code>        \$this-&gt;model = new UserModel();</code>     |
| 14 | <code>        \$this-&gt;helpers = ['form', 'url'];</code>   |
| 15 | <code>    }</code>   |
| 16 |  |
| 17 | <code>    public function index()</code>                     |
| 18 | <code>    {</code>   |
| 19 | <code>        \$data = [</code>                              |
| 20 | <code>            'title' =&gt; 'Register'</code>            |



```

21         ];
22
23         return view('register', $data);
24     }
25
26     public function store()
27     {
28         $data = $this->request->getPost(['username',
29 'email', 'password']);
30
31         if (! $this->validate([
32             'username' =>
33 'required|min_length[5]|max_length[20]|is_unique[user.
34 username]',
35             'email' =>
36 'required|valid_email|is_unique[user.email]',
37             'password' => 'required|min_length[8]',
38         ])) {
39             return redirect()->back()->withInput()-
40 >with('errors', $this->validator->getErrors());
41         }
42
43         $data['password'] =
44 password_hash($data['password'], PASSWORD_DEFAULT);
45
46         $save = $this->model->save($data);
47
48         if ($save) {
49             session()->setFlashdata('success',
50 'Register Berhasil!');
51             return redirect()->to(base_url('login'));
52         }
53     }
54 }

```

|    |   |
|----|---|
| 45 | } else {  |
| 46 | session()->setFlashdata('error',    \$this->model->errors()); |
| 47 | return redirect()->back()->withInput();                       |
| 48 | }   |
| 49 | }   |
| 50 | }   |

## J. Pembahasan *RegsiterController.php*

Pada baris [1] [2] [3] [4] [5], syntax

```
<?php
```

```
namespace App\Controllers;
```

```
use App\Models\UserModel;
```

penggunaan baris-baris tersebut digunakan untuk mendefinisikan namespace App\Controllers, yang menunjukkan bahwa kelas RegisterController terletak di dalam direktori App\Controllers. Selanjutnya, kode mengimpor kelas UserModel yang berfungsi untuk berinteraksi dengan data pengguna atau users dalam database.

Pada baris [7], syntax `class RegisterController extends BaseController` digunakan untuk mendeklarasikan kelas RegisterController yang meng-extend atau mewarisi fungsionalitas dari BaseController.

Pada baris [8] [9] [10] [11] [12] [13] [14] [15], syntax

```
{
    protected $model;

    public function __construct()
    {
        $this->model = new UserModel();
    }
}
```

```

        $this->helpers = ['form', 'url'];
    }

```

pada baris-baris tersebut merupakan properti \$model, objek UserModel yang digunakan untuk melakukan operasi-operasi terkait database terhadap entitas pengguna (users). Konstruktor (\_\_construct()) bertanggung jawab atas inisialisasi objek \$model, memastikan bahwa kontroler dapat berinteraksi dengan data pengguna dengan efisien. Selain itu, konstruktor juga mengatur \$helpers untuk memuat helper form dan url, yang akan tersedia untuk digunakan dalam kontroler ini.

Pada abris [17] [18] [19] [20] [21] [22] [23] [24], syntax

```

public function index()
{
    $data = [
        'title' => 'Register'
    ];
    return view('register', $data);
}

```

Pada baris-baris tersebut merupakan metode index() yang menangani permintaan untuk menampilkan halaman registrasi dengan membuat array \$data yang berisi judul halaman ('title' => 'Register'). Selanjutnya, metode ini juga mengembalikan view 'register' dengan data \$data, yang akan menampilkan formulir registrasi dengan judul "Register".

Pada baris [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36][37] [38][39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50], syntax

```

public function store()
{
    $data = $this->request->getPost(['username', 'email',
    'password']);
}

```

```

        if (! $this->validate([

            'username'                                     =>
            'required|min_length[5]|max_length[20]|is_unique[user.username]',

            'email'                                         =>
            'required|valid_email|is_unique[user.email]',

            'password' => 'required|min_length[8]',

        ])) {

            return redirect()->back()->withInput()->with('errors', $this->validator->getErrors());

        }

        $data['password'] = password_hash($data['password'],
        PASSWORD_DEFAULT);

        $save = $this->model->save($data);

        if ($save) {

            session()->setFlashdata('success', 'Register
            Berhasil!');

            return redirect()->to(base_url('login'));

        } else {

            session()->setFlashdata('error', $this->model->errors());

```

```

        return redirect()->back()->withInput();
    }
}
}

```

pada baris-baris tersebut merupakan penggunaan metode `store()` yang menangani proses penyimpanan data registrasi pengguna setelah formulir registrasi dikirimkan. Data yang dikirimkan diambil menggunakan `$this->request->getPost()`, kemudian divalidasi dengan `$this->validate()` untuk memastikan bahwa username dan email unik serta memenuhi persyaratan lainnya seperti panjang minimum dan validitas email. Apabila validasi gagal, pengguna diarahkan kembali ke halaman sebelumnya dengan data input sebelumnya dan pesan kesalahan. Namun, apabila validasi berhasil, password di-hash menggunakan `password_hash()` sebelum disimpan ke database. Data kemudian disimpan menggunakan `$this->model->save($data)`. Saat penyimpanan berhasil, pengguna diarahkan ke halaman login dengan pesan sukses; dan apabila gagal, maka pengguna diarahkan kembali ke halaman registrasi dengan pesan kesalahan.

#### K. Source *UserController.php*

Table 6. Source Code Modul 6 *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' =>
$username])->orWhere(['email' => $username])->first();
21         if ($dataUser) {
22             if (password_verify($pw,
$dataUser['password'])) {
23                 session()->set([
24                     'username' =>
$dataUser['username'],
25                     'email' => $dataUser['email'],
26                     'logged_in' => true
27                 ]);
28
29                 return redirect()-
>to(base_url('buku'));
30             } else {
31                 session()->setFlashdata('pesan',
'Password salah');
32                 return redirect()-
>to(base_url('login'));
33             }
34         } else {
35             session()->setFlashdata('pesan', 'Username
atau email tidak ditemukan');

```

|    |   |
|----|---|
| 36 | return redirect()->to(base_url('login')); |
| 37 | }   |
| 38 | }   |
| 39 |   |
| 40 | public function logout()                  |
| 41 | {   |
| 42 | session()->destroy();                     |
| 43 | return redirect()->to(base_url('login')); |
| 44 | }   |
| 45 | }   |

#### **L. Pembahasan *UserController.php***

Pada Pada baris [1] [2] [3] [4] [5] [6], syntax

```
<?php

namespace App\Controllers;

use App\Controllers\BaseController;

use App\Models\UserModel;
```

pada baris tersebut digunakan untuk mendefinisikan namespace App\Controllers, yang menunjukkan bahwa kelas UserController berada di dalam direktori App\Controllers. Selain itu, kode tersebut juga digunakan untuk mengimpor kelas BaseController dan UserModel, dimana BaseController menyediakan fungsionalitas dasar untuk semua kontroler, dan UserModel digunakan untuk berinteraksi dengan tabel pengguna dalam database.

Pada baris [8], syntax `class UserController extends BaseController` digunakan untuk mendeklarasikan kelas UserController yang meng-extend BaseController.

Pada baris [9] [10] [11] [12] [13], syntax

```
{

    public function index()
```

```

{
    return view('login');
}

```

pada baris-baris tersebut merupakan Metode index() digunakan untuk menampilkan halaman login dengan mengembalikan view 'login', yang mungkin berisi formulir login.

Pada baris [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38], syntax

```

public function login()
{
    $user = new UserModel();

    $username = $this->request->getPost('username');

    $pw = $this->request->getPost('password');

    $dataUser = $user->where(['username' => $username])->orWhere(['email' => $username])->first();

    if ($dataUser) {
        if (password_verify($pw, $dataUser['password'])) {
            session()->set([
                'username' => $dataUser['username'],
                'email' => $dataUser['email'],
                'logged_in' => true
            ]);

            return redirect()->to(base_url('buku'));
        } else {

```



```

        session()->setFlashdata('pesan',      'Password
salah');

        return redirect()->to(base_url('login'));

    }

    } else {

        session()->setFlashdata('pesan',      'Username  atau
email tidak ditemukan');

        return redirect()->to(base_url('login'));

    }

}

```

pada baris-baris tersebut merupakan metode login(), yang digunakan untuk menangani proses autentikasi pengguna dengan mengambil data username dan password dari formulir login menggunakan `$this->request->getPost()`. Metode ini akan mencari pengguna berdasarkan username atau email menggunakan `UserModel`. Apabila pengguna ditemukan, verifikasi password dilakukan menggunakan `password_verify()`. Apabila verifikasi berhasil, data pengguna disimpan dalam sesi dan pengguna diarahkan ke halaman 'buku'. Apabila verifikasi gagal, pesan kesalahan disimpan dalam sesi dan pengguna diarahkan kembali ke halaman login. Lalu apabila pengguna tidak ditemukan, pesan kesalahan juga disimpan dalam sesi dan pengguna diarahkan kembali ke halaman login.

Pada baris [40] [41] [42] [43] [44] [45], syntax

```

public function logout()

{

    session()->destroy();

    return redirect()->to(base_url('login'));

}

}

```

pada baris-baris tersebut merupakan metode logout() digunakan untuk mengakhiri sesi pengguna yang sedang login dengan menghancurkan sesi pengguna menggunakan session()->destroy() dan kemudian mengarahkan pengguna kembali ke halaman login.

#### M. Source Code *AuthFilter.php*

Table 7. Source Code Modul 6 *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 | }  |

## N. Pembahasan *AuthFilter.php*

Pada baris [1] [2] [3] [4] [5] [6] [7], syntax

```
<?php

namespace App\Filters;

use CodeIgniter\Filters\FilterInterface;

use CodeIgniter\HTTP\RequestInterface;

use CodeIgniter\HTTP\ResponseInterface;
```

pada baris-baris tersebut digunakan untuk mendefinisikan namespace App\Filters, menunjukkan bahwa kelas AuthFilter berada di dalam direktori App\Filters. Kode-kode tersebut juga mengimpor antarmuka FilterInterface serta kelas RequestInterface dan ResponseInterface dari CodeIgniter. Penggunaan FilterInterface untuk menentukan metode yang harus diimplementasikan oleh filter.

Pada baris [9], syntax `class AuthFilter implements FilterInterface` digunakan untuk mendeklarasikan kelas AuthFilter yang mengimplementasikan FilterInterface.

Pada baris [10] [11] [12] [13] [14] [15] [16] [17], syntax

```
{

    public function before(RequestInterface $request,
        $arguments = null)

    {

        if(!session()->get('logged_in')){

            session()->setFlashdata('pesan', 'Login terlebih
dahulu!');

            return redirect()->to(base_url('login'));

        }

    }
```

```
}
```

pada baris-baris tersebut merupakan deklarasi metode `before()` yang dieksekusi sebelum permintaan HTTP mencapai kontroler, memeriksa apakah sesi pengguna memiliki nilai `logged_in` yang bernilai `true`. Apabila pengguna belum login, nilai `logged_in` tidak ada atau `false`, maka metode ini mengatur pesan flash menggunakan `session()->setFlashdata('pesan', 'Login terlebih dahulu!')` dan mengarahkan pengguna ke halaman login dengan `redirect()->to(base_url('login'))`.

Pada baris [19] [20] [21] [22] [23], syntax

```
public function after(RequestInterface $request,
ResponseInterface $response, $arguments = null)
{
    // Do something here
}
}
```

pada baris-baris tersebut merupakan deklarasi metode `after()` yang dieksekusi setelah permintaan HTTP diproses oleh kontroler dan sebelum respons dikirim ke klien. Dalam kode ini, metode `after()` tidak melakukan apa-apa, tetapi disediakan untuk menambahkan logika pasca-pemrosesan. Ini bisa berarti mengevaluasi hasil dari respons yang telah dikirim atau menambahkan informasi tambahan ke dalam respons sebelum dikirim kembali ke klien.

## O. Source Code *Create.php*

Table 8. Source Code Modul 6 *Create.php*

|   |                        |
|---|------------------------|
| 1 | <!DOCTYPE html>        |
| 2 | <html lang="en">       |
| 3 |                        |
| 4 | <head>                 |
| 5 | <meta charset="UTF-8"> |
| 6 |                        |

|   |  |
|---|--|
| 7 | <pre> &lt;meta                                http-equiv="X-UA-Compatible" content="IE=edge"&gt;     &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;     &lt;title&gt;Home&lt;/title&gt;     &lt;link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0- alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384- KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5IL y+dN9+nJOZ" crossorigin="anonymous"&gt;     &lt;style&gt;         .card-form {             max-width: 400px;         }     &lt;/style&gt; &lt;/head&gt; &lt;body&gt;     &lt;div class="container"&gt;         &lt;div class="row justify-content-center"&gt;             &lt;div class="col-12"&gt;                 &lt;center&gt;&lt;h2 class="mb-4"&gt;Tambah Data Buku&lt;/h2&gt;&lt;/center&gt;             &lt;/div&gt;             &lt;div class="col-6"&gt;                 &lt;div class="card card-form mx-auto"&gt;                     &lt;!-- Card view dan posisinya di tengah halaman --&gt; </pre> |
|---|--|

|    |   |
|----|---|
| 30 | <code>&lt;div class="card-body"&gt;</code>                            |
|    | <code>    &lt;form                    action="&lt;?=</code>           |
|    | <code>base_url('/buku/create')      ?&gt;"      method="post"</code>  |
| 31 | <code>onsubmit="return validateForm()"&gt;</code>                     |
| 32 | <code>        &lt;div class="mb-3"&gt;</code>                         |
|    | <code>            &lt;label          for="judul"</code>               |
| 33 | <code>class="form-label"&gt;Judul&lt;/label&gt;</code>                |
|    | <code>            &lt;input          type="text"</code>               |
| 34 | <code>class="form-control" id="judul" name="judul"&gt;</code>         |
|    | <code>            &lt;small  id="judulError"</code>                   |
| 35 | <code>class="text-danger"&gt;&lt;/small&gt;</code>                    |
| 36 | <code>        &lt;/div&gt;</code>                                     |
| 37 | <code>        &lt;div class="mb-3"&gt;</code>                         |
|    | <code>            &lt;label      for="penulis"</code>                 |
| 38 | <code>class="form-label"&gt;Penulis&lt;/label&gt;</code>              |
|    | <code>            &lt;input          type="text"</code>               |
| 39 | <code>class="form-control" id="penulis" name="penulis"&gt;</code>     |
|    | <code>            &lt;small</code>                                    |
| 40 | <code>id="penulisError" class="text-danger"&gt;&lt;/small&gt;</code>  |
| 41 | <code>        &lt;/div&gt;</code>                                     |
| 42 | <code>        &lt;div class="mb-3"&gt;</code>                         |
|    | <code>            &lt;label      for="penerbit"</code>                |
| 43 | <code>class="form-label"&gt;Penerbit&lt;/label&gt;</code>             |
|    | <code>            &lt;input          type="text"</code>               |
| 44 | <code>class="form-control" id="penerbit" name="penerbit"&gt;</code>   |
|    | <code>            &lt;small</code>                                    |
| 45 | <code>id="penerbitError" class="text-danger"&gt;&lt;/small&gt;</code> |
| 46 | <code>        &lt;/div&gt;</code>                                     |
| 47 | <code>        &lt;div class="mb-3"&gt;</code>                         |

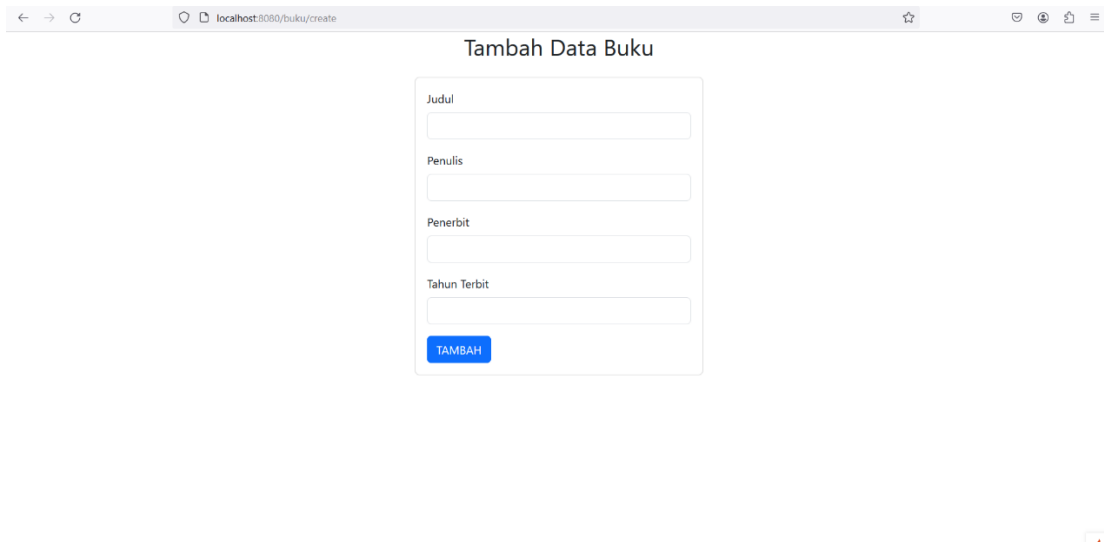


|    |   |                   |                |
|----|---|-------------------|----------------|
|    | var   | judulError        | =              |
| 70 | document.getElementById("judulError");        |                   |                |
|    | var   | penulisError      | =              |
| 71 | document.getElementById("penulisError");      |                   |                |
|    | var   | penerbitError     | =              |
| 72 | document.getElementById("penerbitError");     |                   |                |
| 73 | var   | tahun_terbitError | =              |
| 74 | document.getElementById("tahun_terbitError"); |                   |                |
| 75 |   |                   |                |
| 76 | var valid = true;                             |                   |                |
| 77 | if (judul === "") {                           |                   |                |
| 78 | judulError.innerHTML                          | =                 | "Judul harus   |
| 79 | diisi";                                       |                   |                |
| 80 | valid = false;                                |                   |                |
| 81 | } else {                                      |                   |                |
| 82 | judulError.innerHTML                          | =                 | "";            |
| 83 | }   |                   |                |
| 84 | if (penulis === "") {                         |                   |                |
| 85 | penulisError.innerHTML                        | =                 | "Penulis harus |
| 86 | diisi";                                       |                   |                |
| 87 | valid = false;                                |                   |                |
| 88 | } else {                                      |                   |                |
| 89 | penulisError.innerHTML                        | =                 | "";            |
| 90 | }   |                   |                |
| 91 | if (penerbit === "") {                        |                   |                |
| 92 | penerbitError.innerHTML                       | =                 | "Penerbit      |
| 93 | harus diisi";                                 |                   |                |
| 94 | valid = false;                                |                   |                |



|    |  |
|----|--|
| 95 | <pre>         } else {             penerbitError.innerText = "";         }          if (tahun_terbit === "") {             tahun_terbitError.innerText = "Tahun             terbit harus diisi";             valid = false;         } else if (isNaN(tahun_terbit)                tahun_terbit &lt; 1800    tahun_terbit &gt; 2024) {             tahun_terbitError.innerText = "Tahun             terbit harus berupa angka antara 1800 dan 2024";             valid = false;         } else {             tahun_terbitError.innerText = "";         }          return valid;     }     &lt;/script&gt;      &lt;script     src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-     alpha3/dist/js/bootstrap.bundle.min.js"     integrity="sha384-     ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7N     TKfAdVQSZe" crossorigin="anonymous"&gt;     &lt;/script&gt; &lt;/body&gt; &lt;/html&gt; </pre> |
|----|--|

**P. Ouput Program *create.php***



Gambar 1. Screenshot Hasil Jawaban Modul 6 *create.php*

## Q. Pembahasan *Create.php*

Pada baris [1] – [17], syntax

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Home</title>

    <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+n
JOZ" crossorigin="anonymous">
```

```

<style>

    .card-form {

        max-width: 400px;

    }

</style>

</head>

```

pada baris-baris tersebut merupakan bagian dari dokumen HTML yang mendefinisikan bagian <head> dari halaman web. Elemen <head> dimulai dengan deklarasi dokumen dan bahasa: <!DOCTYPE html> menunjukkan bahwa dokumen ini menggunakan HTML5, sementara <html lang="en"> menunjukkan bahwa bahasa utama dokumen ini adalah bahasa Inggris, yang penting untuk SEO dan aksesibilitas. Karakter encoding didefinisikan dengan <meta charset="UTF-8">, yang menetapkan encoding dokumen sebagai UTF-8 untuk mendukung berbagai karakter dari berbagai bahasa. Untuk kompatibilitas browser, <meta http-equiv="X-UA-Compatible" content="IE=edge"> bertugas memberitahu browser untuk menggunakan mesin rendering terbaru atau Edge mode. Elemen <meta name="viewport" content="width=device-width, initial-scale=1.0"> mengatur viewport agar sesuai dengan lebar perangkat dan skala awal 1.0, memastikan tampilan yang responsif dan cocok pada berbagai ukuran layar. Lalu untuk judul halaman ditetapkan dengan <title>Home</title>, yang penting untuk tampilan pada tab browser dan SEO. Pada kode tersebut juga terdapat proses mengimpor Bootstrap CSS yang dilakukan dengan <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">, dimana akan mengimpor file CSS dari Bootstrap versi 5.3.0-alpha3 dari CDN untuk membantu dalam membuat tata letak halaman yang responsif dan modern. Atribut integrity digunakan untuk keamanan, memverifikasi bahwa file yang diunduh tidak berubah, dan crossorigin="anonymous" menentukan bahwa permintaan sumber daya ini adalah permintaan cross-origin tanpa memerlukan kredensial. Pada kode tersebut juga terdapat penggunaan gaya CSS dengan menggunakan tag <style> untuk

menambahkan kelas `.card-form` yang membatasi lebar maksimal elemen dengan kelas tersebut hingga 400px, css digunakan untuk memodifikasi gaya khusus yang tidak disediakan oleh Bootstrap.

Pada baris [19] – [59], syntax

```
<body>

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-12">

                <center><h2          class="mb-4">Tambah          Data
Buku</h2></center>

            </div>

            <div class="col-6">

                <div class="card card-form mx-auto"> <!-- Card
view dan posisinya di tengah halaman -->

                    <div class="card-body">

                        <form                    action="<?=
base_url('/buku/create') ?>" method="post" onsubmit="return
validateForm()">

                            <div class="mb-3">

                                <label                    for="judul"
class="form-label">Judul</label>

                                    <input                    type="text"
class="form-control" id="judul" name="judul">

                                            <small                    id="judulError"
class="text-danger"></small>
```

```

        </div>

        <div class="mb-3">

            <label                for="penulis"
class="form-label">Penulis</label>

            <input                type="text"
class="form-control" id="penulis" name="penulis">

            <small                id="penulisError"
class="text-danger"></small>

        </div>

        <div class="mb-3">

            <label                for="penerbit"
class="form-label">Penerbit</label>

            <input                type="text"
class="form-control" id="penerbit" name="penerbit">

            <small                id="penerbitError"
class="text-danger"></small>

        </div>

        <div class="mb-3">

            <label                for="tahun_terbit"
class="form-label">Tahun Terbit</label>

            <input                type="text"
class="form-control" id="tahun_terbit" name="tahun_terbit">

            <small                id="tahun_terbitError"
class="text-danger"></small>

        </div>

```

```

        <button type="submit" class="btn
btn-primary">TAMBAH</button>

    </form>

</div>

</div>

</div>

</div>

</div>

```

pada baris-baris tersebut bertugas untuk mendefinisikan struktur dan tampilan sebuah halaman web menggunakan HTML dan Bootstrap untuk membuat form penambahan data buku. Pada kode tersebut dimulai dengan tag `<body>` sebagai bagian utama dari dokumen HTML yang menampung semua konten yang akan ditampilkan di halaman web, lalu `<div class="container">` yang menggunakan kelas container dari Bootstrap untuk memberikan margin dan padding. Selanjutnya, `<div class="row justify-content-center">` membuat baris grid dengan konten dipusatkan di dalamnya. Judul halaman ditampilkan menggunakan `<div class="col-12">` yang mengisi seluruh lebar kontainer, dan `<h2 class="mb-4">Tambah Data Buku</h2>` untuk teks judul dengan margin bawah. Formulir tambah data buku ditempatkan dalam `<div class="col-6">`, di dalamnya terdapat `<div class="card card-form mx-auto">` untuk membuat kartu dengan margin otomatis yang memposisikan kartu di tengah horizontal. Bagian `<div class="card-body">` mengelompokkan konten kartu, dan `<form action="<?= base_url('/buku/create') ?>" method="post" onsubmit="return validateForm()">` mengatur pengiriman data ke URL dengan metode POST, serta memanggil fungsi JavaScript `validateForm()` saat formulir dikirimkan untuk validasi. Selanjutnya tombol `<button type="submit" class="btn btn-primary">TAMBAH</button>` digunakan untuk mengirimkan formulir dengan gaya tombol biru dari Bootstrap. Setiap input dalam formulir mengikuti pola yang terdiri dari `<div class="mb-3">` untuk margin bawah, `<label for="judul" class="form-label">Judul</label>` untuk memberikan label pada input, `<input type="text" class="form-`

control" id="judul" name="judul"> untuk input teks, dan <small id="judulError" class="text-danger"></small> untuk menampilkan pesan error dengan gaya teks merah.

Pada baris [61] – [108], syntax

```
<script>

    function validateForm() {

        var                                judul                                =
document.getElementById("judul").value;

        var                                penulis                                =
document.getElementById("penulis").value;

        var                                penerbit                                =
document.getElementById("penerbit").value;

        var                                tahun_terbit                                =
document.getElementById("tahun_terbit").value;

        var                                judulError                                =
document.getElementById("judulError");

        var                                penulisError                                =
document.getElementById("penulisError");

        var                                penerbitError                                =
document.getElementById("penerbitError");

        var                                tahun_terbitError                                =
document.getElementById("tahun_terbitError");

        var valid = true;

        if (judul === "") {

            judulError.innerText = "Judul harus diisi";

            valid = false;
```

```

    } else {

        judulError.innerText = "";

    }

    if (penulis === "") {

        penulisError.innerText = "Penulis harus
diisi";

        valid = false;

    } else {

        penulisError.innerText = "";

    }

    if (penerbit === "") {

        penerbitError.innerText = "Penerbit harus
diisi";

        valid = false;

    } else {

        penerbitError.innerText = "";

    }

    if (tahun_terbit === "") {

        tahun_terbitError.innerText = "Tahun terbit
harus diisi";

        valid = false;

```



```

        } else if (isNaN(tahun_terbit) || tahun_terbit <
1800 || tahun_terbit > 2024) {

            tahun_terbitError.innerText = "Tahun terbit
harus berupa angka antara 1800 dan 2024";

            valid = false;

        } else {

            tahun_terbitError.innerText = "";

        }

        return valid;

    }

</script>

```

pada baris-baris tersebut merupakan kode javaScript yang digunakan untuk memvalidasi form input data buku sebelum form tersebut dikirimkan. Fungsi validateForm() bertugas memastikan semua input dalam form diisi dengan benar sebelum form dikirimkan. Fungsi ini dimulai dengan mendapatkan nilai dari input form: var judul = document.getElementById("judul").value, var penulis = document.getElementById("penulis").value, var penerbit = document.getElementById("penerbit").value, var tahun\_terbit = document.getElementById("tahun\_terbit").value. Kemudian, elemen-elemen kecil yang akan digunakan untuk menampilkan pesan error jika ada kesalahan input diambil: var judulError = document.getElementById("judulError"), var penulisError = document.getElementById("penulisError"), var penerbitError = document.getElementById("penerbitError"), var tahun\_terbitError = document.getElementById("tahun\_terbitError"). Variabel valid diinisialisasi sebagai true untuk melacak status validasi form dan akan diubah menjadi false jika ditemukan kesalahan input. Validasi dilakukan untuk setiap input: jika input judul kosong, tampilkan pesan error dan set valid ke false, jika tidak, kosongkan pesan error. Hal yang sama juga dilakukan untuk input penulis dan penerbit. Untuk input tahun\_terbit, jika kosong, bukan angka, atau tidak

dalam rentang 1800 hingga 2024, tampilkan pesan error dan set valid ke false, jika tidak, kosongkan pesan error. Fungsi tersebut akhirnya mengembalikan nilai valid yang menunjukkan apakah form valid atau tidak. Apabila false, form tidak akan dikirim.

Pada baris [110] – [114], syntax

```
<script      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"

            integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdvQ
SZe" crossorigin="anonymous">

    </script>

</body>

</html>
```

Pada baris-baris tersebut merupakan library Bootstrap dari CDN (Content Delivery Network) dengan versi 5.3.0-alpha3.

## R. Source Code *edit.php*

Table 9. Source Code Modul 6 *edit.php*

|   |   |
|---|---|
| 1 | <!DOCTYPE html>   |
| 2 | <html lang="en">  |
| 3 |   |
| 4 | <head>  |
| 5 | <meta charset="UTF-8">  |
| 6 | <meta                            http-equiv="X-UA-Compatible" |
|   | content="IE=edge">  |
| 7 | <meta name="viewport" content="width=device-width,            |
|   | initial-scale=1.0">   |
| 8 | <title>Edit Data</title>                                      |
| 9 |   |

|    |  |
|----|--|
| 10 | <pre> &lt;link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0- alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384- </pre> |
| 11 | <pre> KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5IL </pre>  |
| 12 | <pre> y+dN9+nJOZ" crossorigin="anonymous"&gt; </pre>   |
| 13 |  |
| 14 | <pre> &lt;/head&gt; </pre>   |
| 15 |  |
| 16 | <pre> &lt;body&gt; </pre>  |
| 17 | <pre> &lt;div class="container"&gt; </pre>   |
| 18 |  |
| 19 |  |
| 20 | <pre> &lt;div class="row"&gt; </pre>   |
| 21 | <pre> &lt;div class="col-12"&gt; </pre>  |
| 22 | <pre> &lt;center&gt;&lt;h2 class="mb-4"&gt;Edit Data Buku&lt;/h2&gt;&lt;/center&gt; </pre>   |
| 23 | <pre> &lt;/div&gt; </pre>  |
| 24 |  |
| 25 | <pre> &lt;div class="col-md-6 mx-auto"&gt; </pre>  |
| 26 | <pre> &lt;div class="card"&gt; </pre>  |
| 27 | <pre> &lt;div class="card-body"&gt; </pre>   |
| 28 | <pre> &lt;form action="&lt;= base_url('/buku/' . \$buku['id'] . '/edit') ?&gt;" method="post" onsubmit="return validateForm()"&gt; </pre>        |
| 29 | <pre> &lt;div class="mb-3"&gt; </pre>  |
| 30 | <pre> &lt;label for="judul" class="form-label"&gt;Judul&lt;/label&gt; </pre>   |

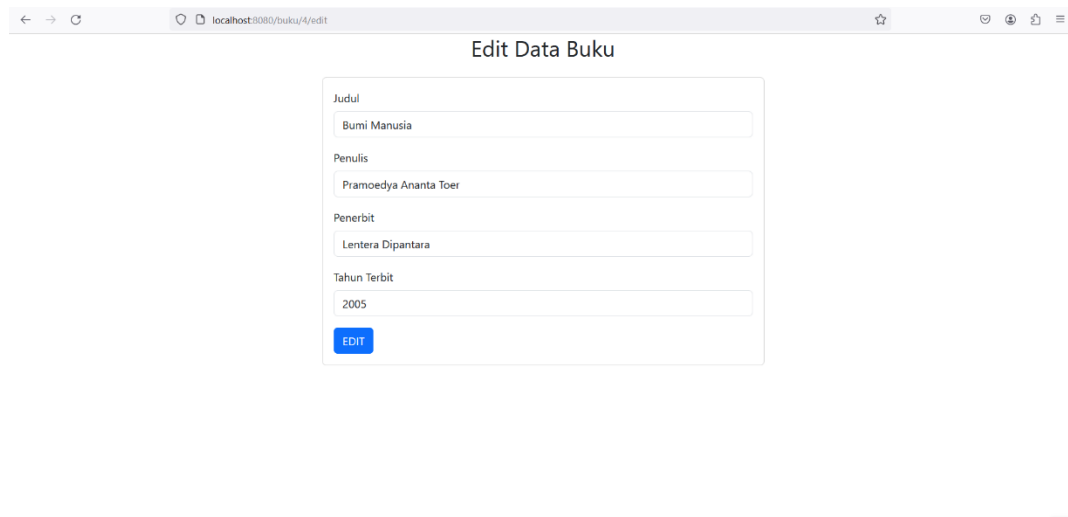
|    |   |
|----|---|
| 31 | <code>&lt;input type="text"</code>                                    |
|    | <code>class="form-control" id="judul" name="judul"</code>             |
| 32 | <code>value="&lt;?=\$buku['judul'] ?&gt;"&gt;</code>                  |
| 33 | <code>&lt;small id="judulError"</code>                                |
| 34 | <code>class="text-danger"&gt;&lt;/small&gt;</code>                    |
| 35 | <code>&lt;/div&gt;</code>   |
|    | <code>&lt;div class="mb-3"&gt;</code>                                 |
| 36 | <code>&lt;label for="penulis"</code>                                  |
|    | <code>class="form-label"&gt;Penulis&lt;/label&gt;</code>              |
| 37 | <code>&lt;input type="text"</code>                                    |
|    | <code>class="form-control" id="penulis" name="penulis"</code>         |
| 38 | <code>value="&lt;?=\$buku['penulis'] ?&gt;"&gt;</code>                |
| 39 | <code>&lt;small</code>  |
| 40 | <code>id="penulisError" class="text-danger"&gt;&lt;/small&gt;</code>  |
| 41 | <code>&lt;/div&gt;</code>   |
|    | <code>&lt;div class="mb-3"&gt;</code>                                 |
| 42 | <code>&lt;label for="penerbit"</code>                                 |
|    | <code>class="form-label"&gt;Penerbit&lt;/label&gt;</code>             |
| 43 | <code>&lt;input type="text"</code>                                    |
|    | <code>class="form-control" id="penerbit" name="penerbit"</code>       |
| 44 | <code>value="&lt;?=\$buku['penerbit'] ?&gt;"&gt;</code>               |
| 45 | <code>&lt;small</code>  |
| 46 | <code>id="penerbitError" class="text-danger"&gt;&lt;/small&gt;</code> |
| 47 | <code>&lt;/div&gt;</code>   |
|    | <code>&lt;div class="mb-3"&gt;</code>                                 |
|    | <code>&lt;label</code>  |
| 48 | <code>for="tahun_terbit" class="form-label"&gt;Tahun</code>           |
|    | <code>Terbit&lt;/label&gt;</code>                                     |

|    |   |  |                            |
|----|---|--|----------------------------|
|    | <code>&lt;input</code>                                      | <code>type="text"</code>                           |                            |
| 49 | <code>class="form-control"</code>                           | <code>id="tahun_terbit"</code>                     |                            |
|    | <code>name="tahun_terbit"</code>                            |  |                            |
| 50 |   | <code>value="&lt;?=</code>                         |                            |
|    | <code>\$buku['tahun_terbit'] ?&gt;"&gt;</code>              |  |                            |
| 51 |   | <code>&lt;small</code>                             |                            |
| 52 | <code>id="tahun_terbitError"</code>                         | <code>class="text-danger"&gt;&lt;/small&gt;</code> |                            |
| 53 |   | <code>&lt;/div&gt;</code>                          |                            |
| 54 |   | <code>&lt;button</code>                            | <code>type="submit"</code> |
| 55 | <code>class="btn btn-primary"&gt;EDIT&lt;/button&gt;</code> |  |                            |
| 56 |   | <code>&lt;/form&gt;</code>                         |                            |
| 57 |   | <code>&lt;/div&gt;</code>                          |                            |
| 58 |   | <code>&lt;/div&gt;</code>                          |                            |
| 59 |   | <code>&lt;/div&gt;</code>                          |                            |
| 60 |   |  |                            |
| 61 |   | <code>&lt;/div&gt;</code>                          |                            |
| 62 |   | <code>&lt;/div&gt;</code>                          |                            |
| 63 |   |  |                            |
| 64 | <code>&lt;script&gt;</code>                                 |  |                            |
|    | <code>function validateForm() {</code>                      |  |                            |
| 65 | <code>var</code>  | <code>judul</code>                                 | <code>=</code>             |
|    | <code>document.getElementById("judul").value;</code>        |  |                            |
| 66 | <code>var</code>  | <code>penulis</code>                               | <code>=</code>             |
|    | <code>document.getElementById("penulis").value;</code>      |  |                            |
| 67 | <code>var</code>  | <code>penerbit</code>                              | <code>=</code>             |
|    | <code>document.getElementById("penerbit").value;</code>     |  |                            |
| 68 | <code>var</code>  | <code>tahun_terbit</code>                          | <code>=</code>             |
| 69 | <code>document.getElementById("tahun_terbit").value;</code> |  |                            |
| 70 |   |  |                            |

|    |   |                   |                |
|----|---|-------------------|----------------|
|    | var   | judulError        | =              |
| 71 | document.getElementById("judulError");        |                   |                |
|    | var   | penulisError      | =              |
| 72 | document.getElementById("penulisError");      |                   |                |
|    | var   | penerbitError     | =              |
| 73 | document.getElementById("penerbitError");     |                   |                |
| 74 | var   | tahun_terbitError | =              |
| 75 | document.getElementById("tahun_terbitError"); |                   |                |
| 76 |   |                   |                |
| 77 | var valid = true;                             |                   |                |
| 78 | if (judul === "") {                           |                   |                |
| 79 | judulError.innerHTML                          | =                 | "Judul harus   |
| 80 | diisi";                                       |                   |                |
| 81 | valid = false;                                |                   |                |
| 82 | } else {                                      |                   |                |
| 83 | judulError.innerHTML                          | =                 | "";            |
| 84 | }   |                   |                |
| 85 | if (penulis === "") {                         |                   |                |
| 86 | penulisError.innerHTML                        | =                 | "Penulis harus |
| 87 | diisi";                                       |                   |                |
| 88 | valid = false;                                |                   |                |
| 89 | } else {                                      |                   |                |
| 90 | penulisError.innerHTML                        | =                 | "";            |
| 91 | }   |                   |                |
| 92 | if (penerbit === "") {                        |                   |                |
| 93 | penerbitError.innerHTML                       | =                 | "Penerbit      |
| 94 | harus diisi";                                 |                   |                |
| 95 | valid = false;                                |                   |                |

|     |   |
|-----|---|
| 96  | <pre>         } else { </pre>   |
| 97  | <pre>             penerbitError.innerText = ""; </pre>  |
| 98  | <pre>         } </pre>  |
| 99  | <pre>         if (tahun_terbit === "") { </pre>   |
| 100 | <pre>             tahun_terbitError.innerText = "Tahun terbit harus diisi"; </pre>  |
| 101 | <pre>             valid = false; </pre>   |
| 102 | <pre>         } else if (isNaN(tahun_terbit)    tahun_terbit &lt; 1800    tahun_terbit &gt; 2024) { </pre>  |
| 103 | <pre>             tahun_terbitError.innerText = "Tahun terbit harus berupa angka antara 1800 dan 2024"; </pre>  |
| 104 |   |
| 105 | <pre>             valid = false; </pre>   |
| 106 | <pre>         } else { </pre>   |
| 107 | <pre>             tahun_terbitError.innerText = ""; </pre>  |
| 108 | <pre>         } </pre>  |
| 109 |   |
| 110 | <pre>         return valid;     } &lt;/script&gt; </pre>  |
| 111 | <pre> &lt;script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0- alpha3/dist/js/bootstrap.bundle.min.js"     integrity="sha384- ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7N TKfAdv </pre> |

## S. Ouput Program *edit.php*



The screenshot shows a web browser window with the address bar displaying 'localhost:3080/tuku/4/edit'. The page title is 'Edit Data Buku'. The form contains the following fields:

- Judul: Bumi Manusia
- Penulis: Pramoedya Ananta Toer
- Penerbit: Lentera Dipantara
- Tahun Terbit: 2005

An 'EDIT' button is located at the bottom of the form.

Gambar 2. Screenshot Hasil Jawaban Modul 6 *edit.php*

## T. Pembahasan *edit.php*

Pada baris [1] -15, syntax

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
    <meta      name="viewport"      content="width=device-width,
initial-scale=1.0">
```

```
    <title>Edit Data</title>
```

```
    <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet"
```



```

        integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+n
JOZ" crossorigin="anonymous">
</head>

```

pada baris-baris tersebut merupakan deklarasi Dokumen `<!DOCTYPE html>` dan `<html lang="en">` yang bertugas untuk membantu browser memahami dan merender halaman dengan benar sesuai standar HTML5. Penggunaan metadata yaitu `<meta charset="UTF-8">`, `<meta http-equiv="X-UA-Compatible">`, dan `<meta name="viewport">` digunakan untuk memberikan informasi penting kepada browser tentang karakter set, mode rendering, dan cara halaman harus ditampilkan di berbagai perangkat. Selanjutnya terdapat tag `<title>` memberikan judul halaman yang muncul di tab browser, memudahkan pengguna mengidentifikasi halaman tersebut. Penggunaan `<link href="...">` untuk menghubungkan halaman HTML dengan file CSS Bootstrap dari CDN sehingga dapat dilakukan penggunaan gaya dan tata letak yang konsisten dari Bootstrap.

Pada baris [14] – [60], syntax

```

<body>

    <div class="container">

        <div class="row">

            <div class="col-12">

                <center><h2          class="mb-4">Edit          Data
Buku</h2></center>

            </div>

            <div class="col-md-6 mx-auto">

                <div class="card">

                    <div class="card-body">

```

```

        <form action="<?= base_url('/buku/' .
$buku['id'] . '/edit') ?>" method="post"

        onsubmit="return validateForm()">

        <div class="mb-3">

            <label                for="judul"
class="form-label">Judul</label>

            <input                type="text"
class="form-control" id="judul" name="judul"

            value="<?= $buku['judul']
?>">

            <small                id="judulError"
class="text-danger"></small>

        </div>

        <div class="mb-3">

            <label                for="penulis"
class="form-label">Penulis</label>

            <input                type="text"
class="form-control" id="penulis" name="penulis"

            value="<?=
$buku['penulis'] ?>">

            <small                id="penulisError"
class="text-danger"></small>

        </div>

        <div class="mb-3">

            <label                for="penerbit"
class="form-label">Penerbit</label>

```

```

                                <input                                type="text"
class="form-control" id="penerbit" name="penerbit"

                                value="<?=
$buku['penerbit'] ?>">

                                <small                                id="penerbitError"
class="text-danger"></small>

                                </div>

                                <div class="mb-3">

                                <label                                for="tahun_terbit"
class="form-label">Tahun Terbit</label>

                                <input                                type="text"
class="form-control" id="tahun_terbit" name="tahun_terbit"

                                value="<?=
$buku['tahun_terbit'] ?>">

                                <small id="tahun_terbitError"
class="text-danger"></small>

                                </div>

                                <button type="submit" class="btn
btn-primary">EDIT</button>

                                </form>

                                </div>

                                </div>

                                </div>

                                </div>

                                </div>

```

pada baris-baris tersebut merupakan bagian dari halaman HTML yang digunakan untuk mengedit data buku. <body> adalah elemen utama yang mengelilingi semua konten halaman, sementara .container digunakan untuk membatasi lebar konten halaman. .row digunakan bersama kelas Bootstrap untuk mengatur baris dalam halaman, yang dapat dibagi menjadi .col-12 untuk pengaturan penuh layar dan .col-md-6 untuk ukuran layar medium. .mx-auto diterapkan untuk menengahkan elemen secara horizontal di dalam parentnya. Elemen <center> digunakan untuk menengahkan teks, sementara <h2> menampilkan judul "Edit Data Buku". .card adalah komponen kartu Bootstrap yang mengelompokkan informasi dengan bingkai abu-abu yang sedikit dibulatkan. <form> digunakan untuk membuat formulir dengan action yang menentukan URL untuk pengeditan data buku dan method="post" untuk pengiriman data. Elemen <input> menerima input pengguna untuk judul, penulis, penerbit, dan tahun terbit, dengan class="form-control" untuk gaya Bootstrap. <small> menampilkan pesan error validasi, dan <button type="submit"> menyediakan tombol untuk mengirimkan formulir. Script JavaScript digunakan untuk validasi form, dan <script> eksternal memuat JavaScript Bootstrap dari CDN.

Pada baris [62] – [11], syntax

```
<script>

    function validateForm() {

        var                                judul                                =
document.getElementById("judul").value;

        var                                penulis                                =
document.getElementById("penulis").value;

        var                                penerbit                                =
document.getElementById("penerbit").value;

        var                                tahun_terbit                                =
document.getElementById("tahun_terbit").value;

        var                                judulError                                =
document.getElementById("judulError");
```

```

        var                penulisError                =
document.getElementById("penulisError");

        var                penerbitError                =
document.getElementById("penerbitError");

        var                tahun_terbitError                =
document.getElementById("tahun_terbitError");

var valid = true;

if (judul === "") {

    judulError.innerText = "Judul harus diisi";

    valid = false;

} else {

    judulError.innerText = "";

}

if (penulis === "") {

    penulisError.innerText = "Penulis harus
diisi";

    valid = false;

} else {

    penulisError.innerText = "";

}

if (penerbit === "") {

    penerbitError.innerText = "Penerbit harus
diisi";

    valid = false;

```

```

        } else {

            penerbitError.innerText = "";

        }

        if (tahun_terbit === "") {

            tahun_terbitError.innerText = "Tahun terbit
harus diisi";

            valid = false;

        } else if (isNaN(tahun_terbit) || tahun_terbit <
1800 || tahun_terbit > 2024) {

            tahun_terbitError.innerText = "Tahun terbit
harus berupa angka antara 1800 dan 2024";

            valid = false;

        } else {

            tahun_terbitError.innerText = "";

        }

        return valid;

    }

```

</script>

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"

```

```

    integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdv

```

pada baris-baris tersebut merupakan kode JavaScript yang berfungsi untuk melakukan validasi formulir sebelum data dikirim ke server. Fungsi `validateForm()` memanfaatkan

metode `document.getElementById("id_element").value` untuk mengambil nilai dari elemen input yang sesuai, seperti judul, penulis, penerbit, dan tahun terbit. Setiap elemen ini kemudian divalidasi: jika kosong, pesan kesalahan ditampilkan di elemen `<small>` yang sesuai, dan variabel `valid` diubah menjadi `false`. Tahun terbit juga diperiksa apakah berupa angka dan berada dalam rentang 1800-2024. `<script>` dengan `src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"` memuat file JavaScript Bootstrap dari CDN, yang menggunakan `integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe"` untuk memastikan keaslian versi yang dimuat, dan `crossorigin="anonymous"` untuk permintaan lintas asal yang aman.

#### U. Source Code *index.php*

Table 10. Source Code Modul 6 *index.php*

|    |  |
|----|--|
| 1  | <code>&lt;!DOCTYPE html&gt;</code>   |
| 2  | <code>&lt;html lang="en"&gt;</code>  |
| 3  | <code>&lt;head&gt;</code>  |
| 4  | <code>    &lt;meta charset="UTF-8"&gt;</code>  |
| 5  | <code>    &lt;meta http-equiv="X-UA-Compatible"</code><br><code>content="IE=edge"&gt;</code>                   |
| 6  | <code>    &lt;meta name="viewport" content="width=device-</code><br><code>width, initial-scale=1.0"&gt;</code> |
| 7  | <code>    &lt;title&gt;Home&lt;/title&gt;</code>   |
| 8  | <code>    &lt;style&gt;</code>   |
| 9  | <code>        body {</code>  |
| 10 | <code>            font-family: Arial, sans-serif;</code>   |
| 11 | <code>            background-color: #f8f9fa;</code>  |
| 12 | <code>        }</code>   |
| 13 |  |
| 14 | <code>        .container {</code>  |
| 15 | <code>            max-width: 900px;</code>   |
| 16 | <code>            margin: 50px auto;</code>  |

```
17         padding: 20px;
18         background-color: #fff;
19         border-radius: 10px;
20         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
21     }
22
23     .btn-logout {
24         margin-top: 20px;
25         float: right;
26     }
27
28     .btn-info {
29         margin-bottom: 10px;
30     }
31
32     .card {
33         margin-top: 20px;
34         border: 1px solid #dee2e6;
35         border-radius: .25rem;
36         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
37     }
38
39     .card-header {
40         background-color: #007bff;
41         color: #fff;
42         padding: 10px;
43         border-bottom: 1px solid rgba(0, 0, 0,
44         0.125);
45     }
46     .card-body {
```



```

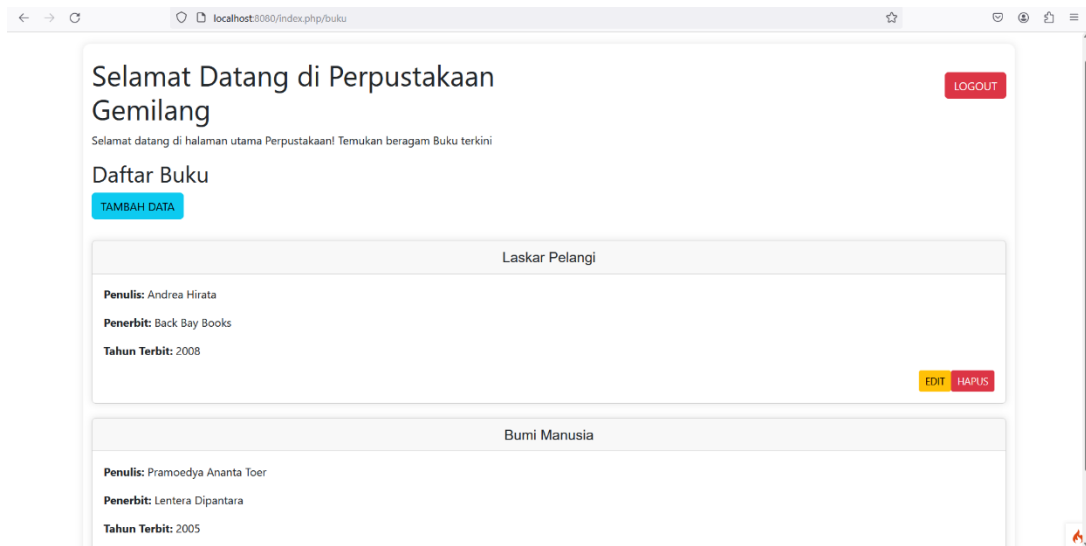
47         padding: 20px;
48     }
49
50     .table {
51         margin-top: 20px;
52     }
53 </style>
54 <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5I
Ly+dN9+nJOZ" crossorigin="anonymous">
55 </head>
56 <body>
57     <div class="container">
58         <div class="row">
59             <div class="col">
60                 <h1>Selamat Datang di Perpustakaan
Gemilang</h1>
61                 <p>Selamat datang di halaman utama
Perpustakaan! Temukan beragam Buku terkini</p>
62             </div>
63             <div class="col">
64                 <a href="<?= base_url('logout') ?>"
class="btn btn-danger btn-logout">LOGOUT</a>
65             </div>
66         </div>
67
68         <!-- create list of books -->
69         <div class="row">

```

|    |  |
|----|--|
| 70 | <code>&lt;div class="col"&gt;</code>   |
| 71 | <code>&lt;h2&gt;Daftar Buku&lt;/h2&gt;</code>                                      |
| 72 | <code>&lt;a class="btn btn-info" href="&lt;?=</code>                               |
|    | <code>base_url('/buku/create') ?&gt;"&gt;TAMBAH DATA&lt;/a&gt;</code>              |
| 73 | <code>&lt;/div&gt;</code>  |
| 74 | <code>&lt;/div&gt;</code>  |
| 75 |  |
| 76 | <code>&lt;div class="row"&gt;</code>   |
| 77 | <code>&lt;div class="col"&gt;</code>   |
| 78 | <code>&lt;?php foreach(\$buku as \$key =&gt;</code>                                |
|    | <code>\$value): ?&gt;</code>   |
| 79 | <code>&lt;div class="card"&gt;</code>  |
| 80 | <code>&lt;div class="card-header"</code>   |
|    | <code>style="font-family: Arial; font-size: 20px;"</code>                          |
| 81 | <code>&lt;center&gt;&lt;?= \$value['judul']</code>                                 |
|    | <code>?&gt;&lt;/center&gt;</code>  |
| 82 | <code>&lt;/div&gt;</code>  |
| 83 | <code>&lt;div class="card-body"&gt;</code>   |
| 84 | <code>&lt;p&gt;&lt;strong&gt;Penulis:&lt;/strong&gt;</code>                        |
|    | <code>&lt;?= \$value['penulis'] ?&gt;&lt;/p&gt;</code>                             |
| 85 | <code>&lt;p&gt;&lt;strong&gt;Penerbit:&lt;/strong&gt;</code>                       |
|    | <code>&lt;?= \$value['penerbit'] ?&gt;&lt;/p&gt;</code>                            |
| 86 | <code>&lt;p&gt;&lt;strong&gt;Tahun</code>  |
|    | <code>Terbit:&lt;/strong&gt; &lt;?= \$value['tahun_terbit'] ?&gt;&lt;/p&gt;</code> |
| 87 | <code>&lt;div class="d-flex justify-</code>  |
|    | <code>content-end"&gt;</code>  |
| 88 | <code>&lt;!-- hapus button --&gt;</code>   |
| 89 | <code>&lt;a href="&lt;=</code>   |
|    | <code>base_url('/buku/' . \$value['id'] . '/edit') ?&gt;"</code>                   |
|    | <code>class="btn btn-warning btn-sm"&gt;EDIT&lt;/a&gt;</code>                      |
| 90 |  |

|     |  |
|-----|--|
|     | <code>&lt;form action="&lt;?=</code>                                 |
|     | <code>base_url('/buku/' . \$value['id'] . '/delete') ?&gt;"</code>   |
| 91  | <code>method="POST" style="display: inline-block;"&gt;</code>        |
|     | <code>&lt;input type="hidden"</code>                                 |
| 92  | <code>name="_method" value="DELETE"&gt;</code>                       |
|     | <code>&lt;button type="submit"</code>                                |
|     | <code>onclick="return confirm('Are you sure you want to</code>       |
|     | <code>delete this item?')" class="btn btn-danger btn-</code>         |
| 93  | <code>sm"&gt;HAPUS&lt;/button&gt;</code>                             |
| 94  | <code>&lt;/form&gt;</code>   |
| 95  | <code>&lt;/div&gt;</code>  |
| 96  | <code>&lt;/div&gt;</code>  |
| 97  | <code>&lt;/div&gt;</code>  |
| 98  | <code>&lt;?php endforeach; ?&gt;</code>                              |
| 99  | <code>&lt;/div&gt;</code>  |
| 100 | <code>&lt;/div&gt;</code>  |
| 101 | <code>&lt;/div&gt;</code>  |
|     | <code>&lt;script</code>  |
|     | <code>src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-</code>      |
|     | <code>alpha3/dist/js/bootstrap.bundle.min.js"</code>                 |
|     | <code>integrity="sha384-</code>                                      |
|     | <code>ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7</code>   |
| 102 | <code>NTKfAdVQSZe" crossorigin="anonymous"&gt;&lt;/script&gt;</code> |
| 103 | <code>&lt;/body&gt;</code>   |
|     | <code>&lt;/html&gt;</code>   |

## V. Output Program *index.php*



Gambar 3. Screenshot Hasil Jawaban Modul 6 *index.php*

## W. Pembahasan *index.php*

Pada baris [1] – [55], syntax

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
    <meta      name="viewport"      content="width=device-width,
initial-scale=1.0">
```

```
    <title>Home</title>
```

```
    <style>
```

```
        body {
```

```
            font-family: Arial, sans-serif;
```

```

        background-color: #f8f9fa;
    }

    .container {
        max-width: 900px;
        margin: 50px auto;
        padding: 20px;
        background-color: #fff;
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    .btn-logout {
        margin-top: 20px;
        float: right;
    }

    .btn-info {
        margin-bottom: 10px;
    }

    .card {
        margin-top: 20px;
        border: 1px solid #dee2e6;
        border-radius: .25rem;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

```

```

        .card-header {
            background-color: #007bff;
            color: #fff;
            padding: 10px;
            border-bottom: 1px solid rgba(0, 0, 0, 0.125);
        }

        .card-body {
            padding: 20px;
        }

        .table {
            margin-top: 20px;
        }

</style>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+n
JOZ" crossorigin="anonymous">

</head>

```

pada baris-baris tersebut merupakan kode HTML dan CSS yang ditunjukkan untuk memberikan gambaran tentang halaman web yang menggunakan Bootstrap 5 dengan gaya tambahan lainnya. Deklarasi dokumen (<!DOCTYPE html> dan <html lang="en">) menetapkan jenis dokumen dan bahasa yang digunakan. Bagian <head> mencakup meta tags untuk karakter set, kompatibilitas IE, dan responsif, serta judul halaman. Penggunaan internal styles menetapkan gaya CSS langsung, sementara stylesheet eksternal mengimpor Bootstrap dari CDN untuk pengaturan gaya tambahan. Lalu penggunaan CSS untuk mengatur body

dengan font dan latar belakang, container untuk tampilan terpusat dengan lebar maksimum dan bayangan kotak, dan tombol logout (btn-logout) ditempatkan di sudut kanan atas. Selain itu, gaya kartu dan tabel ditentukan untuk margin, bayangan, dan warna latar belakang yang konsisten.

Pada baris [56] – [73], syntax

```
<body>

    <div class="container">

        <div class="row">

            <div class="col">

                <h1>Selamat    Datang    di    Perpustakaan
Gemilang</h1>

                <p>Selamat    datang    di    halaman    utama
Perpustakaan! Temukan beragam Buku terkini</p>

            </div>

            <div class="col">

                <a href="{%= base_url('logout') %}" class="btn
btn-danger btn-logout">LOGOUT</a>

            </div>

        </div>

        <div class="row">

            <div class="col">

                <h2>Daftar Buku</h2>

                <a    class="btn    btn-info"    href="{%=
base_url('/buku/create') %}">TAMBAH DATA</a>
```

```
</div>
```

```
</div>
```

pada baris-baris tersebut merupakan penggunaan dari container `<div class="container">`) yang digunakan untuk membatasi konten agar tetap rapi di tengah halaman dengan lebar maksimum 900px, dan secara otomatis menyesuaikan jarak tepi (padding) dari layar. Grid system Bootstrap, seperti `<div class="row">` dan `<div class="col">`, memungkinkan pembagian halaman menjadi bagian-bagian yang responsif dan mudah diatur. Setiap `<div class="row">` digunakan untuk menandai baris baru di dalam container, sementara `<div class="col">` digunakan untuk membuat kolom di dalam baris, dengan lebar yang merata secara default. Pada halaman tersebut, elemen-elemen seperti header utama untuk menampilkan judul perpustakaan dan deskripsi singkat, tombol logout dengan gaya Bootstrap yang menonjol, judul untuk daftar buku, dan tombol tambah data untuk mengarahkan pengguna ke halaman tambah buku (`/buku/create`) membantu dalam navigasi dan tampilan yang terorganisir.

Pada baris [75] – [99], syntax

```
<div class="row">
```

```
<div class="col">
```

```
<?php foreach($buku as $key => $value): ?>
```

```
<div class="card">
```

```
<div class="card-header" style="font-family: Arial; font-size: 20px;">
```

```
<center><?=$value['judul']
?></center>
```

```
</div>
```

```
<div class="card-body">
```

```
<p><strong>Penulis:</strong><?=$value['penulis'] ?></p>
```



```

                <p><strong>Penerbit:</strong>      <?=
$value['penerbit'] ?></p>

                <p><strong>Tahun Terbit:</strong> <?=
$value['tahun_terbit'] ?></p>

                <div class="d-flex justify-content-
end">

                    <a href="<?= base_url('/buku/' .
$value['id'] . '/edit') ?>" class="btn btn-warning btn-
sm">EDIT</a>

                    <form                                action="<?=
base_url('/buku/' . $value['id'] . '/delete') ?>"
method="POST" style="display: inline-block;">

                        <input                                type="hidden"
name="_method" value="DELETE">

                        <button                                type="submit"
onclick="return confirm('Are you sure you want to delete this
item?')>" class="btn btn-danger btn-sm">HAPUS</button>

                    </form>

                </div>

            </div>

        </div>

    </div>

    <?php endforeach; ?>

</div>

</div>

</div>

```

pada baris-baris tersebut merupakan bagian dari HTML yang di-generate oleh PHP untuk menampilkan daftar buku dari variabel \$buku. Penggunaan konstruksi PHP `foreach($buku as $key => $value)` digunakan untuk mengiterasi setiap elemen dalam array \$buku, di mana setiap elemen direpresentasikan sebagai sebuah kartu (`<div class="card">`). Setiap kartu ini menampilkan informasi buku seperti judul, penulis, penerbit, dan tahun terbit. Bagian header kartu (`<div class="card-header">`) menampilkan judul buku dengan gaya tertentu, sedangkan dalam badan kartu (`<div class="card-body">`), informasi seperti penulis, penerbit, dan tahun terbit ditampilkan dalam paragraf (`<p>`). Setiap kartu juga dilengkapi dengan tombol EDIT (`<a>`) untuk mengarahkan pengguna ke halaman edit buku berdasarkan ID buku, serta form HAPUS (`<form>`) dengan tombol HAPUS untuk mengonfirmasi dan menghapus buku dari daftar.

Pada baris [100] – [102], syntax

```
<script      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"      integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdvQ
SZe" crossorigin="anonymous"></script>

</body>

</html>
```

pada baris tersebut merupakan tag `<script>` yang digunakan untuk menyematkan kode JavaScript ke dalam halaman web. Pada kode tersebut, atribut `src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"` menunjukkan bahwa browser harus mengunduh file JavaScript Bootstrap terbaru dari CDN JSDelivr. Atribut `integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdvQSZe"` digunakan untuk memastikan integritas file yang diunduh dengan memverifikasi hash SHA-384 dari file tersebut. Hal ini membantu mencegah pengambilan file yang dimodifikasi secara tidak sah. Atribut `crossorigin="anonymous"` menetapkan bahwa permintaan untuk file JavaScript tidak akan mengirimkan kredensial pengguna

## X. Source Code *login.php*

Table 11. Source Code Modul 6 *login.php*

|    |   |
|----|---|
| 1  | <!DOCTYPE html>   |
| 2  | <html lang="en">  |
| 3  |   |
| 4  | <head>  |
| 5  | <meta charset="UTF-8">                                  |
| 6  | <meta http-equiv="X-UA-Compatible" content="IE=edge">   |
| 7  | <meta name="viewport" content="width=device-width,      |
| 8  | initial-scale=1.0">                                     |
| 9  | <title>Login</title>                                    |
| 10 | <link   |
|    | href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-     |
|    | alpha3/dist/css/bootstrap.min.css" rel="stylesheet"     |
|    | integrity="sha384-                                      |
|    | KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy |
|    | +dN9+nJOZ" crossorigin="anonymous">                     |
| 11 |   |
| 12 | <style>   |
| 13 | body {  |
| 14 | background-color: #f8f9fa;                              |
| 15 | }   |
| 16 |   |
| 17 | .card-view {  |
| 18 | max-width: 400px;                                       |
| 19 | margin: auto;   |
| 20 | margin-top: 100px;                                      |
| 21 | padding: 20px;  |
| 22 | border-radius: 10px;                                    |
| 23 | box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);                |
| 24 | background-color: #fff;                                 |

```
25 }
26
27 .card-view .card-title {
28   font-size: 24px;
29   font-weight: bold;
30   text-align: center;
31   margin-bottom: 20px;
32 }
33
34 .card-view form {
35   margin-bottom: 0;
36 }
37
38 .card-view .form-group {
39   margin-bottom: 20px;
40 }
41
42 .card-view .form-group label {
43   font-weight: bold;
44 }
45
46 .card-view .btn-primary {
47   width: 100%;
48 }
49
50 .card-view .mt-3 {
51   text-align: center;
52 }
53
54 </style>
55 </head>
```

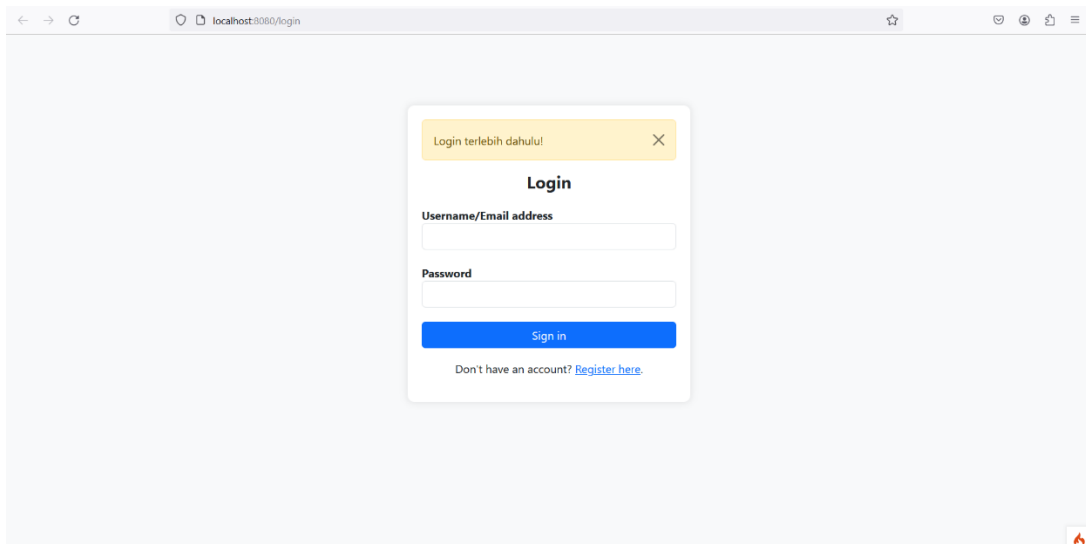
|    |  |
|----|--|
| 56 |  |
| 57 | <body>   |
| 58 | <div class="container">  |
| 59 | <div class="card-view">  |
| 60 | <?php if (!empty(session()->getFlashdata('pesan'))) :<br>?>  |
| 61 | <div class="alert alert-warning alert-dismissible fade<br>show" role="alert">                          |
| 62 | <?= session()->getFlashdata('pesan'); ?>   |
| 63 | <button type="button" class="btn-close" data-bs-<br>dismiss="alert" aria-label="Close"></button>       |
| 64 | </div>   |
| 65 | <?php endif; ?>  |
| 66 | <h5 class="card-title">Login</h5>  |
| 67 | <form action="<?= base_url('/login') ?>" method="post">  |
| 68 | <div class="form-group">   |
| 69 | <label for="email">Username/Email address</label>  |
| 70 | <input type="text" class="form-control" id="email"<br>name="username" value="<?= old('username') ?>">  |
| 71 | </div>   |
| 72 | <div class="form-group">   |
| 73 | <label for="password">Password</label>   |
| 74 | <input type="password" class="form-control"<br>id="password" name="password">                          |
| 75 | </div>   |
| 76 | <button type="submit" class="btn btn-primary">Sign<br>in</button>                                      |
| 77 | <p class="mt-3">Don't have an account? <a href="<?=<br>base_url('register') ?>">Register here</a>.</p> |
| 78 | </form>  |
| 79 | </div>   |

```

80 </div>
81 <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
    alpha3/dist/js/bootstrap.bundle.min.js"
82 integrity="sha384-
    ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NT
    KfAdVQSZe" crossorigin="anonymous">
83 </script>
84 </body>
85 </html>

```

## Y. Output Program *login.php*



Gambar 4. Screenshot Hasil Jawaban Modul 6 *login.php*

## Z. Pembahasan *login.php*

Pada baris [1] – [55], syntax

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta      name="viewport"      content="width=device-width,
initial-scale=1.0">

<title>Login</title>

<link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet"

      integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+n
JOZ" crossorigin="anonymous">

<style>

  body {

    background-color: #f8f9fa;

  }

  .card-view {

    max-width: 400px;

    margin: auto;

    margin-top: 100px;

    padding: 20px;

    border-radius: 10px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    background-color: #fff;

  }

  .card-view .card-title {

```

```
        font-size: 24px;

        font-weight: bold;

        text-align: center;

        margin-bottom: 20px;
    }

    .card-view form {

        margin-bottom: 0;
    }

    .card-view .form-group {

        margin-bottom: 20px;
    }

    .card-view .form-group label {

        font-weight: bold;
    }

    .card-view .btn-primary {

        width: 100%;
    }

    .card-view .mt-3 {

        text-align: center;
    }

</style>

</head>
```



pada baris-baris tersebut merupakan kode HTML yang diberikan digunakan untuk membuat halaman login yang sederhana namun efektif dengan bantuan Bootstrap, sebuah framework CSS populer. Struktur dasar HTML termasuk deklarasi `<!DOCTYPE html>` untuk menunjukkan bahwa dokumen ini adalah HTML5, elemen `<html lang="en">` untuk menetapkan bahasa utama dokumen, serta elemen meta untuk mengatur karakter encoding ke UTF-8, kompatibilitas dengan Internet Explorer, dan skala tampilan yang responsif. Judul halaman ditetapkan dengan `<title>Login</title>` dan CSS Bootstrap diimpor dari CDN untuk menggunakan gaya dan komponen yang sudah ada. Gaya kustom CSS diterapkan pada body untuk mengatur warna latar belakang (`#f8f9fa`), dan pada elemen div dengan kelas `.card-view` untuk menampilkan kartu login yang terpusat dengan lebar maksimum 400px, margin otomatis, padding, border-radius, box-shadow, dan warna latar belakang putih. Gaya tambahan untuk elemen di dalam `.card-view` termasuk pengaturan judul kartu, form, form-group, label, tombol utama, dan elemen dengan kelas `.mt-3`. Di dalam `<body>`, elemen `<div class="container">` digunakan untuk membungkus konten, sementara elemen `.card-view` menampilkan kartu login dengan pesan flash jika ada, judul "Login", dan form login yang mencakup input untuk username/email, password, tombol "Sign in", dan tautan ke halaman pendaftaran. Selanjutnya, tag `<script>` menghubungkan ke file JavaScript Bootstrap dari CDN, yang diperlukan untuk menjalankan komponen dinamis Bootstrap seperti alert yang dapat ditutup.

Pada baris [57] – [80], syntax

```
<body>

    <div class="container">

        <div class="card-view">

            <?php                if                (!empty(session()-
>getFlashdata('pesan'))): ?>

                <div class="alert alert-warning alert-
dismissible fade show" role="alert">

                    <?= session()->getFlashdata('pesan'); ?>
```

```

        <button type="button" class="btn-close"
data-bs-dismiss="alert" aria-label="Close"></button>

    </div>

    <?php endif; ?>

    <h5 class="card-title">Login</h5>

    <form action="<?= base_url('/login') ?>"
method="post">

        <div class="form-group">

            <label for="email">Username/Email
address</label>

            <input type="text" class="form-control"
id="email" name="username" value="<?= old('username') ?>">

        </div>

        <div class="form-group">

            <label for="password">Password</label>

            <input type="password" class="form-
control" id="password" name="password">

        </div>

        <button type="submit" class="btn btn-
primary">Sign in</button>

        <p class="mt-3">Don't have an account? <a
href="<?= base_url('register') ?>">Register here</a>.</p>

    </form>

</div>

</div>

```

pada baris-baris tersebut merupakan bagian dari halaman login yang menggunakan PHP dan Bootstrap untuk memberikan tampilan yang bersih dan responsif serta menangani masukan pengguna secara efisien. Struktur utamanya dimulai dengan tag `<body>` yang membungkus seluruh konten halaman, diikuti oleh elemen `<div class="container">` yang merupakan container Bootstrap untuk memberikan padding dan margin otomatis, serta elemen `<div class="card-view">` yang digunakan untuk menampilkan formulir login dengan gaya khusus yang telah didefinisikan. Di dalam card-view, jika ada pesan flash yang disimpan dalam sesi, akan ditampilkan elemen `<div class="alert alert-warning alert-dismissible fade show" role="alert">` yang menampilkan pesan flash sebagai alert Bootstrap dengan opsi untuk menutup alert menggunakan tombol `<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>`. Judul formulir ditampilkan menggunakan `<h5 class="card-title">Login</h5>` yang memberikan tampilan tebal dan terpusat pada judul "Login". Formulir login dimulai dengan `<form action="<?= base_url('/login') ?>" method="post">` yang mengirimkan data login menggunakan metode POST ke endpoint /login. Di dalam formulir, terdapat elemen `<div class="form-group">` yang mengelompokkan label dan input untuk username/email dan password dengan gaya Bootstrap. Input username/email diwakili oleh `<input type="text" class="form-control" id="email" name="username" value="<?= old('username') ?>">` dan input password diwakili oleh `<input type="password" class="form-control" id="password" name="password">`. Formulir ditutup dengan tombol `<button type="submit" class="btn btn-primary">Sign in</button>` yang mengirimkan formulir, serta paragraf `<p class="mt-3">Don't have an account? <a href="<?= base_url('register') ?>">Register here</a></p>` yang memberikan tautan ke halaman pendaftaran bagi pengguna yang belum memiliki akun.

Pada baris [81] – [85], syntax

```
<script      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"

            integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdvQ
SZe" crossorigin="anonymous">

</script>
```



|    |                                     |
|----|-------------------------------------|
| 12 | margin: 0;                          |
| 13 | padding: 0;                         |
| 14 | display: flex;                      |
| 15 | justify-content: center;            |
| 16 | align-items: center;                |
| 17 | height: 100vh;                      |
| 18 | }                                   |
| 19 |                                     |
| 20 | .card {                             |
| 21 | width: 500px;                       |
| 22 | padding: 30px;                      |
| 23 | background-color: #fff;             |
| 24 | border-radius: 10px;                |
| 25 | box-shadow: 0px 0px 10px            |
|    | rgba(0,0,0,0.1);                    |
| 26 | }                                   |
| 27 |                                     |
| 28 | .form-group {                       |
| 29 | margin-bottom: 30px;                |
| 30 | }                                   |
| 31 |                                     |
| 32 | .form-control {                     |
| 33 | width: 100%;                        |
| 34 | padding: 10px;                      |
| 35 | border: 1px solid #ced4da;          |
| 36 | border-radius: 5px;                 |
| 37 | transition: border-color 0.3s ease; |
| 38 | font-size: 16px;                    |
| 39 | }                                   |
| 40 |                                     |
| 41 | .form-control:focus {               |

|    |   |
|----|---|
| 42 | border-color: #007bff;                    |
| 43 | }   |
| 44 |   |
| 45 | .form-label {                             |
| 46 | color: #000;                              |
| 47 | transition: all 0.3s ease;                |
| 48 | margin-bottom: 5px;                       |
| 49 | display: block;                           |
| 50 | }   |
| 51 |   |
| 52 | .form-control:focus ~ .form-label, .form- |
|    | control:not(:placeholder-shown).form-     |
|    | control:not(:focus) ~ .form-label {       |
| 53 | top: -10px;                               |
| 54 | font-size: 12px;                          |
| 55 | color: #007bff;                           |
| 56 | }   |
| 57 |   |
| 58 | .btn-primary {                            |
| 59 | background-color: #007bff;                |
| 60 | border: none;                             |
| 61 | border-radius: 5px;                       |
| 62 | width: 100%;                              |
| 63 | padding: 15px;                            |
| 64 | color: #fff;                              |
| 65 | cursor: pointer;                          |
| 66 | transition: background-color 0.3s ease;   |
| 67 | }   |
| 68 |   |
| 69 | .btn-primary:hover {                      |
| 70 | background-color: #0056b3;                |

|    |   |
|----|---|
| 71 | }   |
| 72 |   |
| 73 | .alert {  |
| 74 | border-radius: 5px;                             |
| 75 | padding: 15px;                                  |
| 76 | margin-bottom: 30px;                            |
| 77 | }   |
| 78 |   |
| 79 | .alert-success {                                |
| 80 | background-color: #d4edda;                      |
| 81 | color: #155724;                                 |
| 82 | border-color: #c3e6cb;                          |
| 83 | }   |
| 84 |   |
| 85 | .alert-danger {                                 |
| 86 | background-color: #f8d7da;                      |
| 87 | color: #721c24;                                 |
| 88 | border-color: #f5c6cb;                          |
| 89 | }   |
| 90 | </style>  |
| 91 | </head>   |
| 92 | <body>  |
| 93 | <div class="card">                              |
| 94 | <?php if (session()->getFlashdata('success')) : |
|    | ?>  |
| 95 | <div class="alert alert-success"><?=            |
|    | session()->getFlashdata('success') ?></div>     |
| 96 | <?php endif; ?>                                 |
| 97 |   |
| 98 | <?php if (session()->getFlashdata('errors')) :  |
|    | ?>  |

|     |   |
|-----|---|
| 99  | <div class="alert alert-danger">                        |
|     | <?php        foreach        (session()-                 |
| 100 | >getFlashdata('errors') as \$error) : ?>                |
|     | <p><?= \$error ?></p>                                   |
| 101 | <?php endforeach; ?>                                    |
| 102 | </div>  |
| 103 | <?php endif; ?>   |
| 104 |   |
| 105 | <form  action="<?=  base_url('/register')  ?>"          |
| 106 | method="post">  |
|     | <div class="form-group">                                |
| 107 | <label                    class="form-label"            |
| 108 | for="username">Username</label>                         |
|     | <input    id="username"    class="form-                 |
| 109 | control" name="username" value="<?= old('username') ?>" |
|     | required />   |
|     | </div>  |
| 110 |   |
| 111 | <div class="form-group">                                |
| 112 | <label                    class="form-label"            |
| 113 | for="email">Email address</label>                       |
|     | <input  id="email"  class="form-control"                |
| 114 | name="email" value="<?= old('email') ?>" required />    |
|     | </div>  |
| 115 |   |
| 116 | <div class="form-group">                                |
| 117 | <label                    class="form-label"            |
| 118 | for="password">Password</label>                         |
|     | <input  type="password"  id="password"                  |
| 119 | name="password" class="form-control" required />        |
|     | </div>  |

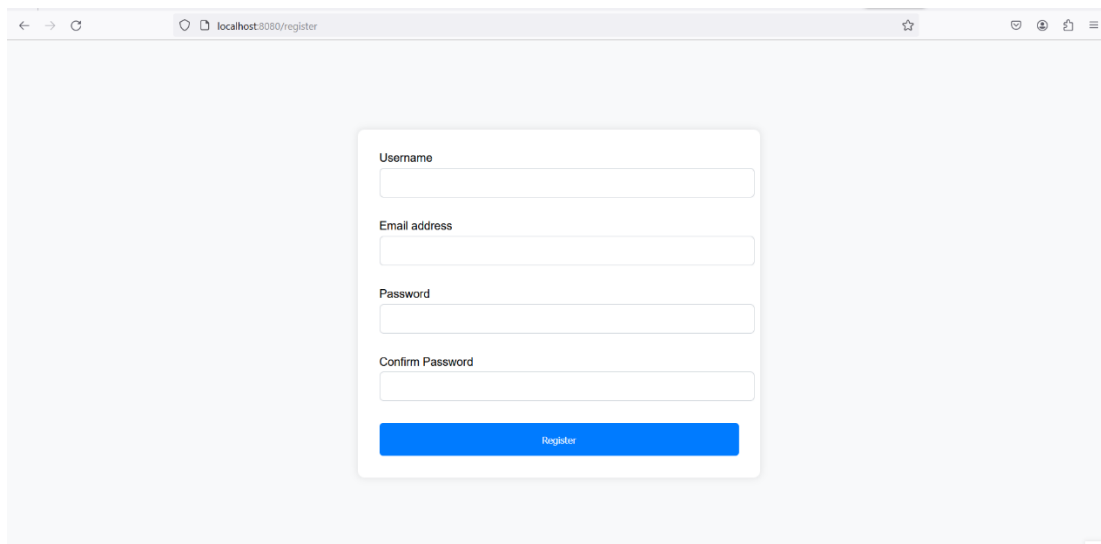


```

120
121         <div class="form-group">
122             <label                class="form-label"
123 for="confirm_password">Confirm Password</label>
124             <input                type="password"
id="confirm_password"        name="confirm_password"
class="form-control" required />
125             </div>
126         <button type="submit" class="btn btn-
127 primary">Register</button>
128     </form>
129 </div>
130 </body>
131 </html>

```

**BB. Ouput Program *register.php***



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/register'. The main content area features a registration form with the following elements:

- Username:** A text input field.
- Email address:** A text input field.
- Password:** A text input field.
- Confirm Password:** A text input field.
- Register:** A blue button with white text.

Gambar 5. Screenshot Hasil Jawaban Modul 6 *register.php*

## CC. Pembahasan *register.php*

Pada baris [1] – [91], syntax

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta      name="viewport"      content="width=device-width,
initial-scale=1.0">

    <title>Register</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f8f9fa;

            margin: 0;

            padding: 0;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

        }

        .card {

            width: 500px;
```

```

padding: 30px;

background-color: #fff;

border-radius: 10px;

box-shadow: 0px 0px 10px 0px rgba(0,0,0,0.1);
}

.form-group {

margin-bottom: 30px;
}

.form-control {

width: 100%;

padding: 10px;

border: 1px solid #ced4da;

border-radius: 5px;

transition: border-color 0.3s ease;

font-size: 16px;
}

.form-control:focus {

border-color: #007bff;
}

.form-label {

color: #000;

transition: all 0.3s ease;

margin-bottom: 5px;

```

```

        display: block;
    }

    .form-control:focus ~ .form-label,
    .form-control:not(:placeholder-shown) .form-control:not(:focus) ~
    .form-label {

        top: -10px;

        font-size: 12px;

        color: #007bff;
    }

    .btn-primary {

        background-color: #007bff;

        border: none;

        border-radius: 5px;

        width: 100%;

        padding: 15px;

        color: #fff;

        cursor: pointer;

        transition: background-color 0.3s ease;
    }

    .btn-primary:hover {

        background-color: #0056b3;
    }

    .alert {

```

```

        border-radius: 5px;

        padding: 15px;

        margin-bottom: 30px;
    }

    .alert-success {

        background-color: #d4edda;

        color: #155724;

        border-color: #c3e6cb;
    }

    .alert-danger {

        background-color: #f8d7da;

        color: #721c24;

        border-color: #f5c6cb;
    }

</style>

</head>

```

pada baris-baris tersebut merupakan bagian HTML dimulai dengan deklarasi dokumen HTML5 dan menetapkan bahasa halaman menjadi bahasa Inggris. Metadata seperti pengaturan karakter encoding UTF-8, kompatibilitas mode rendering untuk Internet Explorer, dan viewport yang responsif juga ditetapkan. Tag <title> menentukan judul halaman yang akan ditampilkan di tab browser. Terdapat juga penggunaan bagian CSS mendefinisikan tampilan halaman, dimulai dari properti global untuk seluruh body seperti jenis font, warna latar belakang, dan display flex untuk penataan elemen. .card mengatur kotak formulir dengan properti seperti lebar, padding, latar belakang, border-radius, dan box-shadow. Pengaturan juga diberikan untuk elemen-elemen seperti .form-group, .form-control

untuk input, .form-label untuk label, .btn-primary untuk tombol "Register", serta .alert, .alert-success, dan .alert-danger untuk pesan peringatan dengan warna dan gaya yang sesuai.

Pada baris [92] – [131], syntax

```
<body>

    <div class="card">

        <?php if (session()->getFlashdata('success')) : ?>

            <div class="alert alert-success"><?= session()-
>getFlashdata('success') ?></div>

        <?php endif; ?>

        <?php if (session()->getFlashdata('errors')) : ?>

            <div class="alert alert-danger">

                <?php          foreach          (session()-
>getFlashdata('errors') as $error) : ?>

                    <p><?= $error ?></p>

                <?php endforeach; ?>

            </div>

        <?php endif; ?>

        <form      action="<?=      base_url('/register')      ?>"
method="post">

            <div class="form-group">

                <label              class="form-label"
for="username">Username</label>

                <input  id="username"  class="form-control"
name="username" value="<?= old('username') ?>" required />

            </div>
```

```

        <div class="form-group">

            <label class="form-label" for="email">Email
address</label>

            <input id="email" class="form-control"
name="email" value="<?= old('email') ?>" required />

        </div>

        <div class="form-group">

            <label class="form-label"
for="password">Password</label>

            <input type="password" id="password"
name="password" class="form-control" required />

        </div>

        <div class="form-group">

            <label class="form-label"
for="confirm_password">Confirm Password</label>

            <input type="password" id="confirm_password"
name="confirm_password" class="form-control" required />

        </div>

        <button type="submit" class="btn btn-
primary">Register</button>

    </form>

</div>

</body>

</html>

```

pada baris-baris tersebut merupakan bagian penting dari halaman HTML untuk menampilkan formulir pendaftaran. Terdapat juga penggunaan PHP untuk menangani pesan-pesan sukses atau error setelah pengguna mengirimkan formulir. Misalnya, jika ada pesan 'success', akan ditampilkan dalam sebuah div dengan kelas 'alert alert-success'. Begitu juga dengan pesan 'errors', di mana setiap kesalahan ditampilkan sebagai paragraf terpisah di dalam div dengan kelas 'alert alert-danger'. Semua pesan ini disimpan dalam session menggunakan framework PHP seperti CodeIgniter. Formulir pendaftaran tersebut terdiri dari empat bagian utama: input untuk username, email, password, dan konfirmasi password. Setiap input dilengkapi dengan label yang jelas, yaitu "Username" dan "Email address" untuk memudahkan pengguna saat mengisi formulir. Input untuk password diatur dengan tipe 'password' untuk mengamankan masukan pengguna. Selanjutnya terdapat tombol "Register" digunakan untuk mengirimkan data formulir ke endpoint '/register' menggunakan metode POST, yang berarti data akan dikirimkan secara aman ke server untuk diproses lebih lanjut.

#### **DD. Source Code *welcome\_mesaage.php***

Table 13.Source Code Modul 6 *welcome\_message.php*

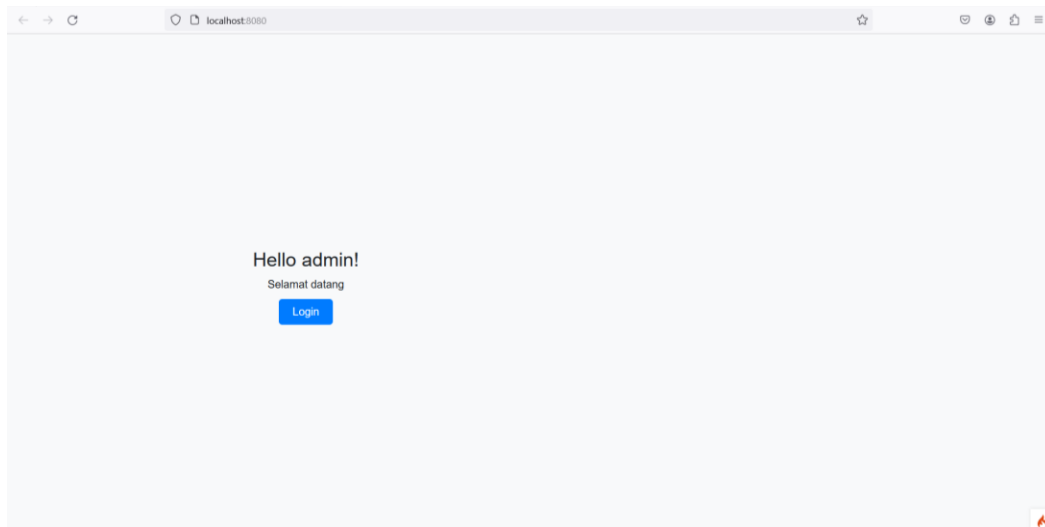
|    |  |
|----|--|
| 1  | <!DOCTYPE html>  |
| 2  | <html lang="en">   |
| 3  |  |
| 4  | <head>   |
| 5  | <meta charset="UTF-8">   |
| 6  | <title>PRAKTIKUM 6</title>   |
| 7  | <meta name="description" content="The small<br>framework with powerful features">                                      |
| 8  | <meta name="viewport" content="width=device-<br>width, initial-scale=1.0">   |
| 9  | <link rel="shortcut icon" type="image/png"<br>href="/favicon.ico">   |
| 10 | <link<br>href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0-<br>alpha3/dist/css/bootstrap.min.css" rel="stylesheet" |



|    |  |
|----|--|
|    | integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous"> |
| 11 |  |
| 12 | <style>  |
| 13 | body {   |
| 14 | font-family: Arial, sans-serif;  |
| 15 | background-color: #f8f9fa;   |
| 16 | margin: 0;   |
| 17 | padding: 0;  |
| 18 | display: flex;   |
| 19 | justify-content: center;   |
| 20 | align-items: center;   |
| 21 | height: 100vh;   |
| 22 | }  |
| 23 |  |
| 24 | .container {   |
| 25 | text-align: center;  |
| 26 | }  |
| 27 |  |
| 28 | .btn-primary {   |
| 29 | background-color: #007bff;   |
| 30 | color: #fff;   |
| 31 | padding: 10px 20px;  |
| 32 | font-size: 16px;   |
| 33 | border: none;  |
| 34 | border-radius: 5px;  |
| 35 | cursor: pointer;   |
| 36 | text-decoration: none;   |
| 37 | transition: background-color 0.3s ease;  |
| 38 | }  |

|    |  |
|----|--|
| 39 |  |
| 40 | .btn-primary:hover {                     |
| 41 | background-color: #0056b3;               |
| 42 | }  |
| 43 | </style>                                 |
| 44 | </head>                                  |
| 45 |  |
| 46 | <body>                                   |
| 47 | <div class="container">                  |
| 48 | <div class="row">                        |
| 49 | <div class="col-md-6">                   |
| 50 | <h3 class="h3">Hello <?php if            |
|    | (session()->get('logged_in') === TRUE) { |
|    | echo session()-                          |
| 51 | >get('username');                        |
|    | } else {                                 |
| 52 | echo "Guest";                            |
| 53 | } ?>!/h3>                                |
| 54 | <p>Selamat datang</p>                    |
| 55 | <a href="<?= base_url('login')           |
| 56 | ?>" class="btn-primary">Login</a>        |
|    | </div>                                   |
| 57 | </div>                                   |
| 58 | </div>                                   |
| 59 | </body>                                  |
| 60 | </html>                                  |
| 61 |  |

**EE. Output Program *welcome\_mesaage.php***



Gambar 6. Screenshot Hasil Jawaban Modul 6 *welcome\_message.php*

### **FF.Pembahasan *welcome\_mesaage.php***

Pada baris [1] – [44], syntax

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>PRAKTIKUM 6</title>
```

```
    <meta name="description" content="The small framework with  
powerful features">
```

```
    <meta      name="viewport"      content="width=device-width,  
initial-scale=1.0">
```

```
    <link      rel="shortcut      icon"      type="image/png"  
href="/favicon.ico">
```

```
    <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0-  
alpha3/dist/css/bootstrap.min.css"      rel="stylesheet"  
integrity="sha384-
```

KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+n  
JOZ" crossorigin="anonymous">

<style>

```
body {  
  
    font-family: Arial, sans-serif;  
  
    background-color: #f8f9fa;  
  
    margin: 0;  
  
    padding: 0;  
  
    display: flex;  
  
    justify-content: center;  
  
    align-items: center;  
  
    height: 100vh;  
  
}  
  
.container {  
  
    text-align: center;  
  
}  
  
.btn-primary {  
  
    background-color: #007bff;  
  
    color: #fff;  
  
    padding: 10px 20px;  
  
    font-size: 16px;  
  
    border: none;  
  
    border-radius: 5px;
```

```

        cursor: pointer;

        text-decoration: none;

        transition: background-color 0.3s ease;
    }

    .btn-primary:hover {

        background-color: #0056b3;

    }

</style>

</head>

```

pada baris-baris tersebut merupakan kode untuk sebuah halaman HTML yang mencakup deklarasi dokumen dan meta tags untuk mengatur karakter set, judul halaman, deskripsi singkat, serta pengaturan viewport untuk responsifitas. Selain itu, terdapat link untuk favicon dan penggunaan Bootstrap melalui CDN untuk mengambil file CSS Bootstrap versi 5.3.0-alpha3. Bagian CSS internalnya menentukan gaya halaman dengan font Arial atau sans-serif, latar belakang abu-abu muda, dan konten yang terpusat secara vertikal dan horizontal di tengah halaman. Class .container digunakan untuk meratakan teks ke tengah, sementara tombol .btn-primary diberi warna latar belakang biru dengan transisi warna saat digulirkan untuk efek visual yang lebih halus.

Pada baris [46] – [61], syntax

```

<body>

    <div class="container">

        <div class="row">

            <div class="col-md-6">

                <h3 class="h3">Hello <?php if (session()-
>get('logged_in') === TRUE) {

```

```

        echo session()->get('username');

    } else {

        echo "Guest";

    } ?>!

```

pada baris-baris tersebut merupakan halaman HTML yang menampilkan pesan sambutan dinamis berdasarkan status login pengguna dan menyediakan tombol untuk login. Struktur dasar HTML menggunakan elemen .container dari Bootstrap, dengan satu baris dan satu kolom besar di dalamnya untuk menampilkan konten. Di dalam elemen <h3>, terdapat kondisi PHP untuk menampilkan nama pengguna jika sudah login, atau menampilkan "Guest" jika belum. Elemen <p> menampilkan pesan "Selamat datang" sebagai tambahan informasi setelah pesan sambutan. Tombol login (<a>) dengan class btn-primary mengarahkan pengguna ke halaman login.

## GG. Source Code BukuModel.php

Table 14. Source Code Modul 6 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 | }                                  |

## HH. Pembahasan BukuModel.php

Pada baris [1] [2] [3] [4] [5], syntax

```
<?php
namespace App\Models;
use CodeIgniter\Model;
```

pada baris-baris tersebut mendefinisikan namespace App\Models dan mengimpor kelas Model dari framework CodeIgniter.

Pada baris [7], syntax `class BukuModel extends Model` baris tersebut mendefinisikan kelas BukuModel merupakan turunan dari kelas Model dari CodeIgniter.

Pada baris [8] -[21], syntax

```
{
    protected $DBGroup          = 'default';
    protected $table             = 'buku';
    protected $primaryKey        = 'id';
    protected $useAutoIncrement = true;
```



```

protected $returnType      = 'array';

protected $useSoftDeletes  = false;

protected $protectFields   = true;

protected $allowedFields   = [

    'judul',

    'penulis',

    'penerbit',

    'tahun_terbit'

];

```

pada baris-baris tersebut digunakan untuk mengkonfigurasi perilaku model dalam berinteraksi dengan tabel database buku. Hal tersebut termasuk pengaturan seperti koneksi basis data (\$DBGroup), nama tabel (\$table), primary key (\$primaryKey), penggunaan auto increment untuk primary key (\$useAutoIncrement), tipe data yang dikembalikan (\$returnType), dan sebagainya.

Pada baris [24] – [28], syntax

```

protected $useTimestamps = false;

protected $dateFormat    = 'datetime';

protected $createdField   = 'created_at';

protected $updatedField   = 'updated_at';

protected $deletedField   = 'deleted_at';

```

pada baris-baris tersebut merupakan bagian yang mengatur apakah model akan menggunakan timestamps, misalnya, created\_at dan updated\_at. Penggunaan \$useTimestamps akan mengontrol penggunaan timestamps, sedangkan \$dateFormat menentukan format tanggal yang digunakan.

Pada baris [31] – [34], syntax

```

protected $validationRules      = [];

    protected $validationMessages = [];

    protected $skipValidation     = false;

    protected $cleanValidationRules = true;

```

pada abris-baris tersebut merupakan properti-properti yang digunakan untuk mengatur aturan validasi data. \$validationRules berisi aturan-aturan validasi, \$validationMessages berisi pesan-pesan kesalahan yang terkait dengan validasi, dan \$skipValidation menentukan apakah validasi harus dilewati atau tidak.

Pada baris [37] – [46], syntax

```

protected $allowCallbacks = true;

    protected $beforeInsert    = [];
    protected $afterInsert     = [];
    protected $beforeUpdate    = [];
    protected $afterUpdate     = [];
    protected $beforeFind      = [];
    protected $afterFind       = [];
    protected $beforeDelete    = [];
    protected $afterDelete     = [];

}

```

Pada baris-baris tersebut merupakan properti-properti yang digunakan untuk mendefinisikan callback-callback yang akan dieksekusi sebelum atau sesudah operasi-insert, update, find (query), atau delete. Callback-callback ini memungkinkan untuk mengeksekusi kode tambahan sebelum atau sesudah melakukan operasi terkait pada data.

## II. Source Code *UserModel.php*

Table 15. Source Code Modul 6 *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     = ['username', 'email',
17     'password'];
18
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     protected $validationRules = [
26         'username' =>
27         'required|min_length[3]|max_length[20]|is_unique[user.username]',
28         'email'    =>
29         'required|valid_email|is_unique[user.email]',
30         'password' => 'required|min_length[6]',
31     ];

```

```

29     protected $validationMessages = [
30         'username' => [
31             'required'    => 'Username is required',
32             'min_length' => 'Username must be at least 3
characters long',
33             'max_length' => 'Username cannot exceed 20
characters',
34             'is_unique'  => 'Username already exists',
35         ],
36         'email' => [
37             'required'    => 'Email is required',
38             'valid_email' => 'You must provide a valid email
address',
39             'is_unique'  => 'Email already exists',
40         ],
41         'password' => [
42             'required'    => 'Password is required',
43             'min_length' => 'Password must be at least 6
characters long',
44         ],
45     ];
46     protected $skipValidation          = false;
47     protected $cleanValidationRules    = true;
48
49     protected $allowCallbacks = true;
50     protected $beforeInsert    = [];
51     protected $afterInsert     = [];
52     protected $beforeUpdate    = [];
53     protected $afterUpdate     = [];
54     protected $beforeFind      = [];
55     protected $afterFind       = [];

```

|    |                          |       |
|----|--------------------------|-------|
| 56 | protected \$beforeDelete | = []; |
| 57 | protected \$afterDelete  | = []; |
| 58 | }                        |       |

## JJ. Pembahasan *UserModel.php*

Pada baris [1] – [5], syntax

```
<?php

namespace App\Models;

use CodeIgniter\Model;
```

pada baris-baris tersebut men deklarasikan namespace App\Models. Hal tersebut menandakan bahwa kelas UserModel berada di direktori App\Models dalam struktur proyek CodeIgniter.

Pada baris [7], syntax `class UserModel extends Model` baris tersebut merupakan kelas UserModel yang mewarisi semua fitur dan metode dari kelas Model dari CodeIgniter.

Pada baris [8] – [16], syntax

```
{

    protected $DBGroup          = 'default';

    protected $table            = 'user';

    protected $primaryKey       = 'id';

    protected $useAutoIncrement = true;

    protected $returnType       = 'array';

    protected $useSoftDeletes   = false;

    protected $protectFields    = true;

    protected $allowedFields     = ['username', 'email',
'password'];
```

pada baris-baris tersebut merupakan konfigurasi utama untuk sebuah model dalam framework PHP seperti CodeIgniter. Properti `$DBGroup` menentukan grup koneksi basis data yang digunakan, dengan 'default' menunjukkan penggunaan koneksi standar. `$table` menentukan nama tabel dalam basis data yang digunakan oleh model ini, dalam kasus ini adalah 'user'. `$primaryKey` menentukan kolom yang berfungsi sebagai primary key dalam tabel ('id'), sementara `$useAutoIncrement` mengatur apakah primary key menggunakan increment otomatis. `$returnType` menentukan jenis data yang dikembalikan oleh model, dalam hal ini 'array'. `$useSoftDeletes` mengatur apakah model mendukung penghapusan lembut (soft deletes), sedangkan `$protectFields` mengontrol apakah field-field tertentu dilindungi dari asignasi secara massal. Selanjutnya, `$allowedFields` menentukan field-field yang diizinkan untuk asignasi secara massal, yang dalam kode tersebut mencakup 'username', 'email', dan 'password'.

Pada baris [18] – [22], syntax

```
protected $useTimestamps = false;

protected $dateFormat      = 'datetime';

protected $createdField    = 'created_at';

protected $updatedField    = 'updated_at';

protected $deletedField    = 'deleted_at';
```

pada abris-baris tersebut bertugas untuk mengatur pengelolaan timestamps dalam sebuah model dalam framework PHP seperti CodeIgniter. Penggunaan `$useTimestamps` untuk mengontrol apakah timestamps seperti `created_at` dan `updated_at` digunakan pada kode tersbeut diatur sebagai false. `$dateFormat` menentukan format tanggal yang digunakan untuk timestamps, dengan nilai 'datetime'. Selanjutnya, `$createdField`, `$updatedField`, dan `$deletedField` menentukan nama kolom dalam tabel basis data yang digunakan untuk masing-masing timestamps, yaitu 'created\_at', 'updated\_at', dan 'deleted\_at'.

Pada baris [24] – [47], syntax

```
protected $validationRules = [
```

```

        'username' =>
'required|min_length[3]|max_length[20]|is_unique[user.username]',

        'email' =>
'required|valid_email|is_unique[user.email]',

        'password' => 'required|min_length[6]',

    ];

    protected $validationMessages = [

        'username' => [

            'required' => 'Username is required',

            'min_length' => 'Username must be at least 3
characters long',

            'max_length' => 'Username cannot exceed 20
characters',

            'is_unique' => 'Username already exists',

        ],

        'email' => [

            'required' => 'Email is required',

            'valid_email' => 'You must provide a valid email
address',

            'is_unique' => 'Email already exists',

        ],

        'password' => [

            'required' => 'Password is required',

```

```

        'min_length' => 'Password must be at least 6
characters long',

    ],

];

protected $skipValidation      = false;

protected $cleanValidationRules = true;

```

pada baris-baris tersebut digunakan untuk mengatur validasi data dalam sebuah model dalam framework PHP seperti CodeIgniter. Penggunaan `$validationRules` untuk menentukan aturan validasi untuk masing-masing field seperti username, email, dan password, termasuk keharusan (required), panjang minimal (min\_length), panjang maksimal (max\_length), dan unik (is\_unique untuk memastikan nilai unik dalam basis data). Lalu penggunaan `$validationMessages` untuk menentukan pesan kesalahan yang ditampilkan jika validasi gagal untuk setiap aturan. `$skipValidation` mengontrol apakah validasi akan dilewati atau tidak (diatur sebagai false untuk melakukan validasi). Serta `$cleanValidationRules` mengontrol apakah aturan validasi harus dibersihkan sebelum digunakan.

Pada baris [49] – [58], syntax

```

protected $allowCallbacks = true;

    protected $beforeInsert    = [];
    protected $afterInsert     = [];
    protected $beforeUpdate    = [];
    protected $afterUpdate     = [];
    protected $beforeFind      = [];
    protected $afterFind       = [];
    protected $beforeDelete    = [];
    protected $afterDelete     = [];

```



```
}
```

pada baris-baris tersebut digunakan untuk mengatur penggunaan callback dalam sebuah model dalam framework PHP seperti CodeIgniter. Penggunaan \$allowCallbacks untuk menentukan apakah fungsi callback seperti sebelum insert atau setelah delete diizinkan, dimana pada kode tersebut diatur menjadi true. Penggunaan properti-properti seperti \$beforeInsert, \$afterInsert, \$beforeUpdate, \$afterUpdate, \$beforeFind, \$afterFind, \$beforeDelete, dan \$afterDelete digunakan untuk menentukan tindakan tertentu yang akan dijalankan sebelum atau setelah operasi dasar pada data, seperti penyisipan, pembaruan, pencarian, dan penghapusan.

#### **KK. Source Code *UserSeeder.php***

Table 16. Source Code Modul 6 *UserSeeder.php*

|    |   |
|----|---|
| 1  | <?php                                   |
| 2  |   |
| 3  | namespace App\Database\Seeds;           |
| 4  |   |
| 5  | use CodeIgniter\Database\Seeder;        |
| 6  |   |
| 7  | class UserSeeder extends Seeder         |
| 8  | {                                       |
| 9  | public function run()                   |
| 10 | {                                       |
| 11 | \$this->db->table('user')->insert([     |
| 12 | 'username' => 'admin',                  |
| 13 | 'email' => 'admin@mail.com',            |
| 14 | 'password' => password_hash('12345678', |
|    | PASSWORD_DEFAULT),                      |
| 15 | ]);                                     |
| 16 | }                                       |
| 17 | }                                       |

## **LL.      Pembahasan *UserSeeder.php***

Pada baris [1] – [5], syntax

```
<?php

namespace App\Database\Seeds;

use CodeIgniter\Database\Seeder;
```

pada baris-baris tersebut merupakan deklarasi namespace App\Database\Seeds, yang menunjukkan lokasi dari kelas UserSeeder di dalam struktur direktori proyek.

Pada abirs [7], syntax `class UserSeeder extends Seeder` baris tersebut merupakan kelas UserSeeder mewarisi semua fungsi dari kelas Seeder CodeIgniter, yang menyediakan metode untuk memasukkan data ke dalam basis data.

Pada baris [8] – [17], syntax

```
{

    public function run()

    {

        $this->db->table('user')->insert([

            'username' => 'admin',

            'email' => 'admin@mail.com',

            'password'      =>      password_hash('12345678',

PASSWORD_DEFAULT),

        ]);

    }

}
```

pada baris-baris tersebut merupakan method run(), method khusus yang digunakan dalam proses seeding data pada basis data. Ketika method ini dieksekusi, kode yang terdapat di dalamnya akan menjalankan operasi untuk memasukkan data ke dalam tabel user. Pada kode

tersebut, terdapat pemanggilan `$this->db->table('user')->insert(...)`, yang berfungsi untuk memasukkan sebuah array asosiatif ke dalam tabel user. Array tersebut berisi nilai untuk kolom-kolom seperti 'username' dengan nilai 'admin', 'email' dengan nilai 'admin@mail.com', dan 'password' dengan hasil dari fungsi `password_hash('12345678', PASSWORD_DEFAULT)`. Fungsi `password_hash()` digunakan untuk mengamankan dan mengenkripsi password '12345678' menggunakan algoritma yang aman sebelum disimpan ke dalam basis data.

#### MM. Source Code 2024-06-12-202936\_Buku

Table 17. Source Code Modul 6 2024-06-12-202936\_Buku

|    |                                     |
|----|-------------------------------------|
| 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 | }                                       |

## **NN. Pembahasan 2024-06-12-202936\_Buku**

Pada baris [1] – [5], syntax

?php

```
namespace App\Database\Migrations;
```

```
use CodeIgniter\Database\Migration;
```

pada baris-baris tersebut merupakan deklarasi namespace App\Database\Migrations, yang menunjukkan lokasi dari kelas Buku di dalam struktur direktori proyek.

Pada baris [7], syntax `class Buku extends Migration` baris tersebut merupakan kelas Buku mewarisi semua fungsi dari kelas Migration CodeIgniter, yang menyediakan metode untuk mengatur skema basis data.

Pada baris [8] – [37], syntax

```
{  
  
    public function up()  
  
    {  
  
        $this->forge->addField([  
  
            'id' => [  
  
                'type' => 'INT',  
  
                'constraint' => 5,  
  
                'unsigned' => true,  
  
                'auto_increment' => true  
  
            ],  
  
            'judul' => [  
  
                'type' => 'VARCHAR',  
  
                'constraint' => 100,  
  
            ],  
  
            'penulis' => [  
  
                'type' => 'VARCHAR',  
  
                'constraint' => 100,  
  
            ],  
  
            'penerbit' => [  

```

```

        'type' => 'VARCHAR',

        'constraint' => 100,

    ],

    'tahun_terbit' => [

        'type' => 'YEAR',

    ],

]);

$this->forge->addKey('id', true);

$this->forge->createTable('buku');

}

```

pada baris-baris tersebut merupakan method `up()`, sebuah method dalam kode migrasi yang bertanggung jawab untuk membuat struktur tabel baru dalam basis data. Pada kode tersebut, method ini digunakan untuk membuat tabel buku dengan menggunakan fungsi `$this->forge->addField([...])` untuk menambahkan definisi kolom ke dalam tabel. Di dalam method ini, terdapat definisi kolom seperti 'id' dengan tipe INT, panjang maksimum 5 digit, dan memiliki atribut unsigned serta `auto_increment` untuk menghasilkan nilai yang otomatis bertambah. Selain itu, terdapat juga kolom 'judul', 'penulis', 'penerbit', dan 'tahun\_terbit' dengan tipe VARCHAR dan YEAR yang masing-masing memiliki constraint panjang maksimum dan tipe data yang sesuai. Metode `$this->forge->addKey('id', true)` digunakan untuk menetapkan kolom 'id' sebagai primary key, sedangkan `$this->forge->createTable('buku')` digunakan untuk mengeksekusi pembuatan tabel 'buku' berdasarkan definisi kolom yang telah ditentukan sebelumnya.

Pada baris [39] – [41], syntax

```

public function down()

{

```

```

        $this->forge->dropTable('buku', true);
    }
}

```

pada baris tersebut merupakan method down() yang digunakan dalam kode migrasi untuk melakukan rollback, khususnya untuk menghapus tabel yang telah dibuat sebelumnya. Dalam implementasinya, method ini menggunakan \$this->forge->dropTable('buku', true) untuk menghapus tabel 'buku' dari basis data. Parameter kedua yang disetel sebagai true mengindikasikan agar SQL yang dihasilkan akan mencakup IF EXISTS, sehingga mengamankan proses penghapusan jika tabel tidak ada dalam basis data saat rollback dilakukan.

#### OO. Source Code 2024-06-12-202940\_User

Table 18. Source Code Modul 6 2024-06-12-202940\_User

|    |                                     |
|----|-------------------------------------|
| 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 | } |

## **PP. Pembahasan 2024-06-12-202940\_User**

Pada baris [1] – [5], syntax

```
<?php
namespace App\Database\Migrations;
use CodeIgniter\Database\Migration;
```

pada baris tersebut digunakan untuk mendeklarasi namespace App\Database\Migrations, yang menunjukkan lokasi dari kelas User di dalam struktur direktori proyek.

Pada baris [7], syntax `class User extends Migration` baris tersebut merupakan kelas User yang mewarisi semua fungsi dari kelas Migration CodeIgniter, serta menyediakan metode untuk mengatur skema basis data.

Pada baris [8] – [43], syntax

```
{
    public function up()
    {
        $this->forge->addField([
            'id'=> [
                'type' => 'INT',
                'constraint' => 5,
                'unsigned' => true,
                'auto_increment' => true
            ],
```

```

        'username' => [
            'type' => 'VARCHAR',
            'constraint' => 100,
        ],
        'email' => [
            'type' => 'VARCHAR',
            'constraint' => 100,
        ],
        'password' => [
            'type' => 'TEXT',

        ],
        'created_at' => [
            'type' => 'DATETIME',
        ],
        'updated_at' => [
            'type' => 'DATETIME',
        ],
        'deleted_at' => [
            'type' => 'DATETIME',
        ],
    ]);

$this->forge->addKey('id', true);

```

```

        $this->forge->createTable('user', true);

    }

```

pada baris-baris tersebut merupakan method up() yang berisi kode untuk membuat tabel user dengan struktur dan properti kolom yang didefinisikan menggunakan \$this->forge->addField([...]). \$this->forge->addField([...]) menambahkan definisi kolom ke dalam tabel, di mana 'id' => [...], 'username' => [...], 'email' => [...], 'password' => [...], 'created\_at' => [...], 'updated\_at' => [...], dan 'deleted\_at' => [...] mendefinisikan kolom-kolom tabel beserta tipe data, panjang maksimum, dan atribut lainnya seperti auto\_increment untuk kolom id dan tipe data DATETIME untuk created\_at, updated\_at, dan deleted\_at. \$this->forge->addKey('id', true) menetapkan kolom id sebagai primary key. \$this->forge->createTable('user', true) membuat tabel dengan nama user berdasarkan definisi yang telah ditentukan, di mana parameter kedua true mengindikasikan untuk menambahkan IF NOT EXISTS pada perintah SQL yang dihasilkan.

Pada baris [45] – [49], syntax

```

public function down()

    {

        $this->forge->dropTable('user', true);

    }

}

```

pada baris-baris tersebut merupakan method down() yang digunakan untuk melakukan rollback, dalam hal ini menghapus tabel yang telah dibuat. \$this->forge->dropTable('user', true) menghapus tabel user dari basis data, di mana parameter kedua true mengindikasikan untuk menambahkan IF EXISTS pada perintah SQL yang dihasilkan.

## QQ. Source Code buku.sql

Table 19.Source Code Modul 6 buku.sql

|   |   |
|---|---|
| 1 | SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO"; |
| 2 | START TRANSACTION;                      |

|    |  |
|----|--|
| 3  | SET time_zone = "+00:00";                          |
| 4  |  |
| 5  |  |
| 6  | CREATE TABLE `buku` (                              |
| 7  | `id` INT(5) UNSIGNED NOT NULL,                     |
| 8  | `judul` VARCHAR(100) NOT NULL,                     |
| 9  | `penulis` VARCHAR(100) NOT NULL,                   |
| 10 | `penerbit` VARCHAR(100) NOT NULL,                  |
| 11 | `tahun_terbit` YEAR(4) NOT NULL                    |
| 12 | ) ENGINE=INNODB DEFAULT CHARSET=utf8               |
|    | COLLATE=utf8_general_ci;                           |
| 13 |  |
| 14 |  |
| 15 | CREATE TABLE `crud` (                              |
| 16 | `id` INT(5) UNSIGNED NOT NULL,                     |
| 17 | `nama` VARCHAR(50) NOT NULL,                       |
| 18 | `nim` VARCHAR(13) NOT NULL,                        |
| 19 | `alamat` TEXT NOT NULL                             |
| 20 | ) ENGINE=INNODB DEFAULT CHARSET=utf8               |
|    | COLLATE=utf8_general_ci;                           |
| 21 |  |
| 22 | INSERT INTO `crud` (`id`, `nama`, `nim`, `alamat`) |
| 23 | VALUES   |
| 24 | (1, 'syahna', '2210817120007', 'Banjarmasin'),     |
| 25 | (4, 'puteri', '2210817120007', 'Rantau');          |
| 26 |  |
| 27 | CREATE TABLE `migrations` (                        |
| 28 | `id` BIGINT(20) UNSIGNED NOT NULL,                 |
| 29 | `version` VARCHAR(255) NOT NULL,                   |
| 30 | `class` VARCHAR(255) NOT NULL,                     |
| 31 | `group` VARCHAR(255) NOT NULL,                     |

|    |   |
|----|---|
| 32 | <code>`namespace` VARCHAR(255) NOT NULL,</code>                   |
| 33 | <code>`time` INT(11) NOT NULL,</code>                             |
| 34 | <code>`batch` INT(11) UNSIGNED NOT NULL</code>                    |
| 35 | <code>) ENGINE=INNODB DEFAULT CHARSET=utf8</code>                 |
|    | <code>COLLATE=utf8_general_ci;</code>                             |
| 36 |   |
| 37 | <code>INSERT INTO `migrations` (`id`, `version`, `class`,</code>  |
|    | <code>`group`, `namespace`, `time`, `batch`) VALUES</code>        |
| 38 | <code>(1, '2024-06-12-080538',</code>                             |
|    | <code>'App\\Database\\Migrations\\Crud', 'default', 'App',</code> |
|    | <code>1684830849, 1),</code>                                      |
| 39 | <code>(2, '2024-06-12-090624',</code>                             |
|    | <code>'App\\Database\\Migrations\\User', 'default', 'App',</code> |
|    | <code>1685438559, 2),</code>                                      |
| 40 | <code>(3, '2024-06-12-015133',</code>                             |
|    | <code>'App\\Database\\Migrations\\Buku', 'default', 'App',</code> |
|    | <code>1686193775, 3),</code>                                      |
| 41 | <code>(4, '2024-06-12-114209',</code>                             |
|    | <code>'App\\Database\\Migrations\\User', 'default', 'App',</code> |
|    | <code>1686193775, 3);</code>                                      |
| 42 |   |
| 43 | <code>CREATE TABLE `user` (</code>                                |
| 44 | <code>`id` INT(5) UNSIGNED NOT NULL,</code>                       |
| 45 | <code>`username` VARCHAR(100) NOT NULL,</code>                    |
| 46 | <code>`email` VARCHAR(100) NOT NULL,</code>                       |
| 47 | <code>`password` TEXT NOT NULL</code>                             |
| 48 | <code>) ENGINE=INNODB DEFAULT CHARSET=utf8</code>                 |
|    | <code>COLLATE=utf8_general_ci;</code>                             |
| 49 |   |
| 50 |   |

|    |  |
|----|--|
| 51 | INSERT INTO `user` (`id`, `username`, `email`,             |
| 52 | `password`) VALUES   |
| 53 | (1, 'user', 'user@gmail.com',                              |
|    | '\$2y\$10\$uikyKAFYu5peatxETHuGq.t6wItNJidIoEy0P5pkBQyyMJD |
|    | qRV39O'),  |
| 54 | (2, 'admin', 'admin@mail.com',                             |
|    | '\$2y\$10\$owPcuOliltHxCooZMtYEhOkEnURD39DSJ1/OQcdTKnvaBYJ |
|    | bwT1D2');  |
| 55 |  |
| 56 | ALTER TABLE `buku`   |
| 57 | ADD PRIMARY KEY (`id`);                                    |
| 58 |  |
| 59 | ALTER TABLE `crud`   |
| 60 | ADD PRIMARY KEY (`id`);                                    |
| 61 |  |
| 62 | ALTER TABLE `migrations`                                   |
| 63 | ADD PRIMARY KEY (`id`);                                    |
| 64 |  |
| 65 | ALTER TABLE `user`   |
| 66 | ADD PRIMARY KEY (`id`);                                    |
| 67 |  |
| 68 | ALTER TABLE `buku`   |
| 69 | MODIFY `id` INT(5) UNSIGNED NOT NULL AUTO_INCREMENT;       |
| 70 |  |
| 71 | ALTER TABLE `crud`   |
| 72 | MODIFY `id` INT(5) UNSIGNED NOT NULL AUTO_INCREMENT,       |
|    | AUTO_INCREMENT=5;  |
| 73 |  |
| 74 | ALTER TABLE `migrations`                                   |
| 75 | MODIFY `id` BIGINT(20) UNSIGNED NOT NULL                   |
|    | AUTO_INCREMENT, AUTO_INCREMENT=5;                          |

|    |  |
|----|--|
| 76 |  |
| 77 | ALTER TABLE `user`                                   |
| 78 | MODIFY `id` INT(5) UNSIGNED NOT NULL AUTO_INCREMENT, |
|    | AUTO_INCREMENT=3;                                    |
| 79 | COMMIT;  |

## **RR. Pembahasan buku.sql**

Pada database yang dibuat oleh skrip sql tersebut digunakan untuk keperluan web. Setiap tabel dalam struktur database tersebut memiliki tujuan untuk mendukung fungsi keseluruhan sistem. Pada skrip tersebut terdapat pembuatan tabel buku yang berfungsi sebagai repositori untuk menyimpan informasi terkait buku, yaitu judul, penulis, penerbit, dan tahun terbit. Lalu terdapat tabel crud untuk menyimpan informasi dasar tentang pengguna atau mahasiswa, dengan kolom yang mencakup ID, nama, NIM (Nomor Induk Mahasiswa), dan alamat. Tabel migrations digunakan dalam mencatat dan melacak setiap perubahan struktur database. Informasi yang tersimpan di sini mencakup versi migrasi, kelas migrasi, grup, namespace, waktu migrasi, dan batch. Pada skrip tersebut juga terdapat pembuatan tabel user yang bertanggung jawab untuk penyimpanan data pengguna sistem, termasuk ID pengguna, nama pengguna (username), email, dan password yang tersimpan dengan metode enkripsi. Database tersebut dibuat untuk mendukung berbagai fungsi pada aplikasi web, seperti manajemen data buku, pengelolaan informasi pengguna, pelacakan perubahan pada struktur database, serta autentikasi dan manajemen keamanan pengguna.

## **SS. Tautan Git**

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/syahputerii/PRAK601-WEB2.git>