

## Praktikum 13: Framework Lanjutan (Modul Login)

### Tujuan

1. Mahasiswa mampu memahami konsep dasar Auth dan Filter.
2. Mahasiswa mampu memahami konsep dasar Login System.
3. Mahasiswa mampu membuat modul login menggunakan Framework Codeigniter 4.

### Instruksi Praktikum

1. Persiapkan text editor misalnya **VSCode**.
2. Buka kembali folder dengan nama **lab11\_php\_ci** pada docroot webserver (**htdocs**)
3. Ikuti langkah-langkah praktikum yang akan dijelaskan berikutnya.

### Langkah-langkah Praktikum

#### Persiapan.

Untuk memulai membuat modul Login, yang perlu disiapkan adalah database server menggunakan MySQL. Pastikan MySQL Server sudah dapat dijalankan melalui XAMPP.

#### Membuat Tabel: User Login

Field	Type Data	Ukuran	Keterangan
<b>id</b>	INT	11	PRIMARY KEY, auto_increment
<b>username</b>	VARCHAR	200	
<b>useremail</b>	VARCHAR	200	
<b>userpassword</b>	VARCHAR	200	

#### Membuat Tabel User

```
CREATE TABLE user (
  id INT(11) auto_increment,
  username VARCHAR(200) NOT NULL,
  useremail VARCHAR(200),
  userpassword VARCHAR(200),
  PRIMARY KEY(id)
);
```

#### Membuat Model User

Selanjutnya adalah membuat Model untuk memproses data Login. Buat file baru pada direktori **app/Models** dengan nama **UserModel.php**

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class UserModel extends Model
```

```
{  
    protected $table = 'user';  
    protected $primaryKey = 'id';  
    protected $useAutoIncrement = true;  
    protected $allowedFields = ['username', 'useremail', 'userpassword'];  
}
```

### Membuat Controller User

Buat Controller baru dengan nama **User.php** pada direktori **app/Controllers**.  
Kemudian tambahkan method **index()** untuk menampilkan daftar user, dan method **login()** untuk proses login.

```
<?php  
  
namespace App\Controllers;  
  
use App\Models\UserModel;  
  
class User extends BaseController  
{  
    public function index()  
    {  
        $title = 'Daftar User';  
        $model = new UserModel();  
        $users = $model->findAll();  
        return view('user/index', compact('users', 'title'));  
    }  
  
    public function login()  
    {  
        helper(['form']);  
        $email = $this->request->getPost('email');  
        $password = $this->request->getPost('password');  
        if (!$email)  
        {  
            return view('user/login');  
        }  
  
        $session = session();  
        $model = new UserModel();  
        $login = $model->where('useremail', $email)->first();  
        if ($login)  
        {  
            $pass = $login['userpassword'];  
            if (password_verify($password, $pass))  
            {  
                $login_data = [  

```

```

        'user_id' => $login['id'],
        'user_name' => $login['username'],
        'user_email' => $login['useremail'],
        'logged_in' => TRUE,
    ];
    $session->set($login_data);
    return redirect('admin/artikel');
}
else
{
    $session->setFlashdata("flash_msg", "Password salah.");
    return redirect()->to('/user/login');
}
}
else
{
    $session->setFlashdata("flash_msg", "email tidak terdaftar.");
    return redirect()->to('/user/login');
}
}
}

```

### Membuat View Login

Buat direktori baru dengan nama **user** pada direktori **app/views**, kemudian buat file baru dengan nama **login.php**.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
    <link rel="stylesheet" href="<?= base_url('/style.css');?>">
</head>
<body>
    <div id="login-wrapper">
        <h1>Sign In</h1>
        <?php if(session()->getFlashdata('flash_msg')):?>
            <div class="alert alert-danger"><?=
session()->getFlashdata('flash_msg') ?></div>
        <?php endif;?>
        <form action="" method="post">
            <div class="mb-3">
                <label for="InputForEmail" class="form-label">Email
address</label>
                <input type="email" name="email" class="form-control"
id="InputForEmail" value="<?= set_value('email') ?>">
            </div>

```

```
        <div class="mb-3">
            <label for="InputForPassword"
class="form-label">Password</label>
            <input type="password" name="password"
class="form-control" id="InputForPassword">
        </div>
        <button type="submit" class="btn
btn-primary">Login</button>
    </form>
</div>
</body>
</html>
```

### Membuat Database Seeder

Database seeder digunakan untuk membuat data dummy. Untuk keperluan ujicoba modul login, kita perlu memasukkan data user dan password ke dalam database. Untuk itu buat database seeder untuk tabel user. Buka CLI, kemudian tulis perintah berikut:

```
php spark make:seeder UserSeeder
```

Selanjutnya, buka file **UserSeeder.php** yang berada di lokasi direktori **/app/Database/Seeds/UserSeeder.php** kemudian isi dengan kode berikut:

```
<?php

namespace App\Database\Seeds;

use CodeIgniter\Database\Seeder;

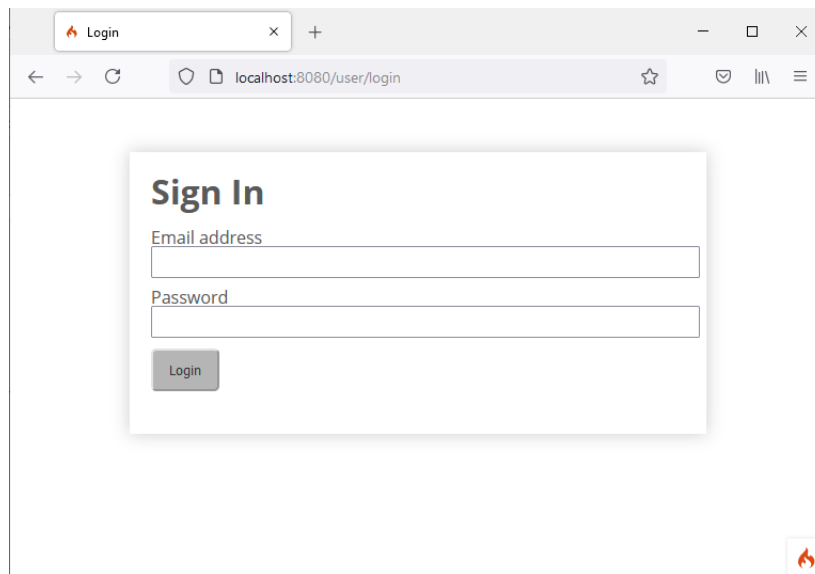
class UserSeeder extends Seeder
{
    public function run()
    {
        $model = model('UserModel');
        $model->insert([
            'username' => 'admin',
            'useremail' => 'admin@email.com',
            'userpassword' => password_hash('admin123', PASSWORD_DEFAULT),
        ]);
    }
}
```

Selanjutnya buka kembali CLI dan ketik perintah berikut:

```
php spark db:seed UserSeeder
```

### Uji Coba Login

Selanjutnya buka url <http://localhost:8080/user/login> seperti berikut:



Gambar 13.1 Login Form

### Menambahkan Auth Filter

Selanjutnya membuat filer untuk halaman admin. Buat file baru dengan nama **Auth.php** pada direktori **app/Filters**.

```
<?php namespace App\Filters;

use CodeIgniter\HTTP\RequestInterface;
use CodeIgniter\HTTP\ResponseInterface;
use CodeIgniter\Filters\FilterInterface;

class Auth implements FilterInterface
{
    public function before(RequestInterface $request, $arguments = null)
    {
        // jika user belum login
        if(! session()->get('logged_in')){
            // maka redirct ke halaman login
            return redirect()->to('/user/login');
        }
    }

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

Selanjutnya buka file **app/Config/Filters.php** tambahkan kode berikut:

```
'auth' => App\Filters\Auth::class
```

```
17      */
18      public $aliases = [
19          'csrf'      => CSRF::class,
20          'toolbar'   => DebugToolbar::class,
21          'honeypot' => Honeypot::class,
22          'auth'      => App\Filters\Auth::class,
23      ];
24
```

Gambar 13.2 Config Filters

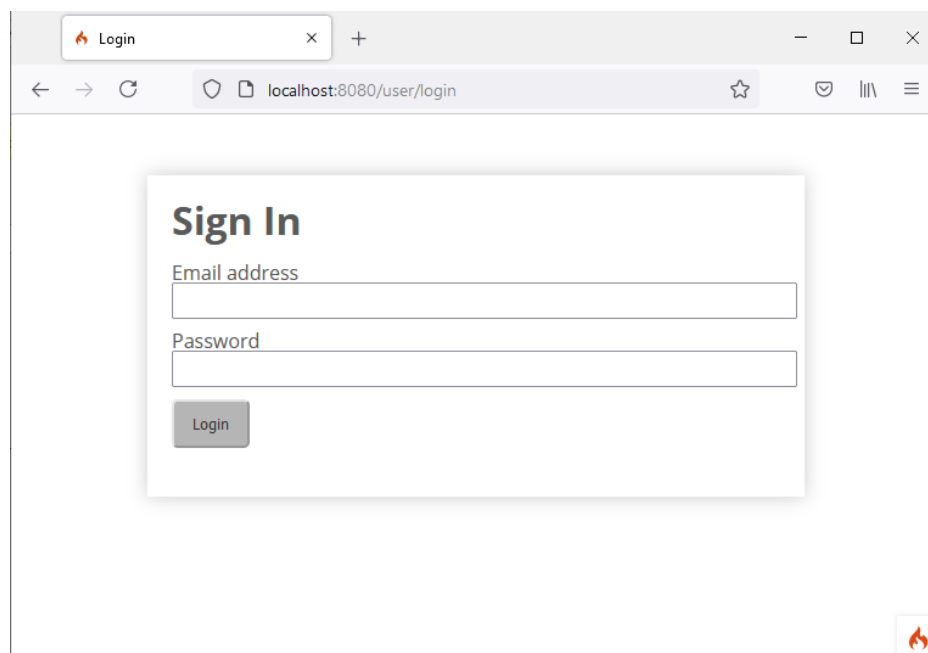
Selanjutnya buka file **app/Config/Routes.php** dan sesuaikan kodenya.

```
39
40 $routes->get('/artikel/(:any)', 'Artikel::view/$1');
41 $routes->group('admin', ['filter' => 'auth'], function($routes) {
42     $routes->get('artikel', 'Artikel::admin_index');
43     $routes->add('artikel/add', 'Artikel::add');
44     $routes->add('artikel/edit/(:any)', 'Artikel::edit/$1');
45     $routes->get('artikel/delete/(:any)', 'Artikel::delete/$1');
46 });
47
```

Gambar 13.3 Config Routes

### Percobaan Akses Menu Admin

Buka url dengan alamat <http://localhost:8080/admin/artikel> ketika alamat tersebut diakses maka, akan dimunculkan halaman login.



Gambar 13.4 Login Admin

### Fungsi Logout

Tambahkan method logout pada Controller User seperti berikut:

```
public function logout()
{
    session()->destroy();
    return redirect()->to('/user/login');
}
```

### Pertanyaan dan Tugas

Selesaikan programnya sesuai Langkah-langkah yang ada. Anda boleh melakukan improvisasi.

### Laporan Praktikum

1. Melanjutkan praktikum sebelumnya pada **repository** dengan nama **Lab11Web**.
2. Kerjakan semua latihan yang diberikan sesuai urutannya.
3. Screenshot setiap perubahannya.
4. Update file **README.md** dan tuliskan penjelasan dari setiap langkah praktikum beserta screenshotnya.
5. **Commit** hasilnya pada **repository** masing-masing.
6. Kirim **URL repository** pada **e-learning** ecampus