

## MODUL 12 Web Framework

---

### 12.1. Deskripsi Singkat

Web framework adalah suatu software yang dirancang untuk mendukung pengembangan suatu aplikasi web. Pada web framework disediakan suatu standar mekanisme utk membangun (*build*) dan memasang (*deploy*) aplikasi web di internet. Fitur-fitur yang biasanya tersedia diantaranya, library untuk mengakses database, templating framework, manajemen session, security, dan lain-lain. Web framework biasanya mempromosikan code reuse untuk memudahkan proses pengembangan aplikasi web.

Secara umum web framework application terbagi menjadi frontend framework dan backend framework. Front end framework fokus pada elemen visual website atau aplikasi di sisi client yang berinteraksi dengan user, meliputi CSS dan Javascript framework. Sedangkan backend framework fokus pada sisi server, yaitu sisi website yg tidak terlihat oleh user, diantaranya PHP framework.

### 12.2. Tujuan Praktikum

Setelah menyelesaikan praktikum pada modul ini, mahasiswa diharapkan dapat mengenal bagaimana menggunakan web framework application untuk mempermudah pengembangan aplikasi web.

### 12.3. Material Praktikum

Praktikum menggunakan editor teks seperti Notepad atau Notepad++ atau aplikasi editor teks lainnya yang tersedia untuk menulis HTML dan program PHP. Pada praktikum ini juga menggunakan web server Apache dan program PHP.

### 12.4. Kegiatan Praktikum

#### 12.4.1. Front-end framework: Bootstrap

Bootstrap merupakan salah satu front-end framework yang sering digunakan. Saat modul ini dibuat, versi terbaru dari Bootstrap adalah Bootstrap 5.2. Untuk melihat dokumentasi cara memulai membuat aplikasi web dengan Bootstrap 5.2, silakan kunjungi halaman berikut.

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Selanjutnya, Anda akan membuat halaman web menggunakan Bootstrap.

1. Buat folder project web dengan nama `webbs`. Kemudian buat file `index.html` dengan kode berikut.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Bootstrap Demo</title>
  </head>
  <body>
  </body>
</html>
```

2. Kemudian sisipkan bootstrap CSS pada tag `head` dan bootstrap JS pada akhir tag `body`.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFB6D0itfPri4tjfhXaWutUpFmBp4vmVor"
crossorigin="anonymous">
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"
crossorigin="anonymous"></script>
```

Selain menggunakan link CSS dan JS eksternal, dapat juga menggunakan link file direktori di server dengan menyimpan file CSS dan JS ke folder project web.

3. Kemudian buat layout halaman web yang terdiri dari header dan 3 kolom artikel dengan menambahkan kode berikut di tag `body`.

```
<div class="container-fluid p-5 bg-primary text-white text-
center">
  <h1>Halaman Web dengan Bootstrap</h1>
  <p>Resize halaman responsive ini untuk melihat efeknya!</p>
</div>

<div class="container mt-5">
  <div class="row">
    <div class="col-sm-4">
      <h3>Kolom 1</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit...</p>
      <p>Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris...</p>
    </div>
    <div class="col-sm-4">
```

```

        <h3>Kolom 2</h3>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit...</p>
        <p>Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris...</p>
    </div>
    <div class="col-sm-4">
        <h3>Kolom 3</h3>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit...</p>
        <p>Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris...</p>
    </div>
</div>
</div>

```

4. Akses halaman web tersebut melalui browser dan resize untuk melihat efek responsifnya. Baca dokumentasi mengenai layout container pada halaman berikut:

<https://getbootstrap.com/docs/5.2/layout/containers/>

5. Kemudian ubah warna header menjadi warna orange gradient dengan text berwarna hitam. Baca dokumentasi mengenai background pada halaman berikut:

<https://getbootstrap.com/docs/5.2/utilities/background/>

6. Selanjutnya, Anda dapat melihat contoh kode untuk halaman-halaman yang dirancang menggunakan bootstrap pada halaman berikut:

<https://getbootstrap.com/docs/5.2/examples/>

Download example untuk halaman sign-in. Lalu copy file index.html dan signin.css ke dalam folder webbs\_signin. Jika anda buka halaman index.html tersebut pada browser, maka anda akan melihat tampilannya tidak serupa dengan yang anda lihat pada halaman getbootstrap di atas.

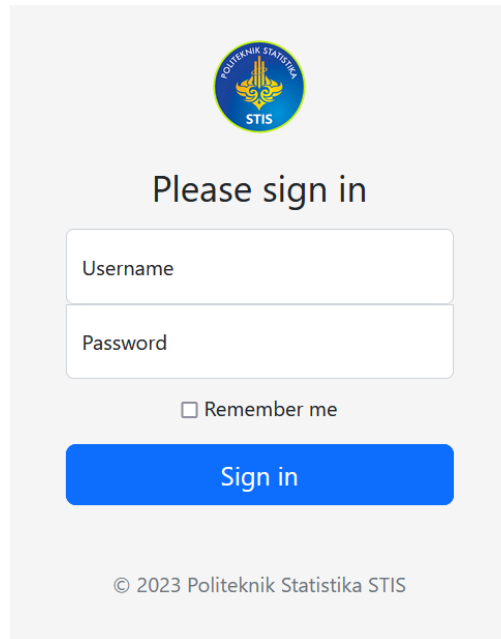
7. Buka dan pelajari script yang terdapat pada index.html. Anda akan menemukan baris-baris yang merujuk ke file external, diantaranya adalah

```
<link href="../../assets/dist/css/bootstrap.min.css" rel="stylesheet">
```

Download source code pada halaman getbootstrap. Cari file yang dimaksud pada folder dist/css, lalu copy dan paste file tersebut pada folder webbs\_signin. Sesuaikan path href pada potongan program di atas. Muat

ulang halaman `index.html`. Seharusnya tampilannya sekarang sudah serupa dengan yang ada pada halaman `getbootstrap`.

8. Masih terdapat kekurangan pada halaman `index.html` tersebut, yaitu logo. Cari baris program pada `index.html` yang merujuk ke file logo. Ganti logo tersebut dengan logo Politeknik Statistika STIS. Sesuaikan tampilan halaman sign-in tersebut sehingga index menjadi seperti berikut ini.



#### 12.4.2. Back-end framework: CodeIgniter

CodeIgniter merupakan salah satu back-end framework berbasis PHP. Selanjutnya, ikuti langkah-langkah berikut untuk membuat project web menggunakan CodeIgniter.

1. Terdapat beberapa cara yang dapat dipilih untuk menginstal CodeIgniter. Silakan membaca lebih detail cara instalasinya pada halaman berikut.

[https://codeigniter.com/user\\_guide/installation/](https://codeigniter.com/user_guide/installation/)

Pada praktikum ini, Anda akan mencoba melakukan instalasi dengan menggunakan Composer. Silakan baca dokumentasinya untuk mengetahui kelebihan instalasi dengan Composer. Jika Anda belum pernah menginstal Composer, kunjungi halaman berikut untuk mengunduh file instalasi Composer dan lakukan instalasi.

<https://getcomposer.org/download/>

Pastikan composer sudah terinstall dengan baik. Lakukan pengecekan melalui command prompt dengan mengetikkan:

```
composer --version
```

Jika berhasil akan muncul informasi versi dari composer Anda.

```
composer version 2.7.4 2024-04-22 21:17:03  
PHP version 8.2.12 (C:\xampp\php\php.exe)
```

Jika belum berhasil, tambahkan path composer ke dalam PATH di Windows Anda. Buka Environment Variables, lalu di bagian "System Variables" atau "User Variables", cari variabel bernama "Path" dan klik pada tombol "Edit". Di dalam kotak dialog "Edit Environment Variable", klik pada tombol "New" dan tambahkan jalur ke direktori instalasi Composer. Secara default, jalur instalasi Composer biasanya di C:\ProgramData\ComposerSetup\bin kemudian simpan dan Keluar. Restart Apache server melalui xampp control panel.

2. Selanjutnya buka command prompt, pindahkan path ke direktori htdocs, dan jalankan perintah berikut.

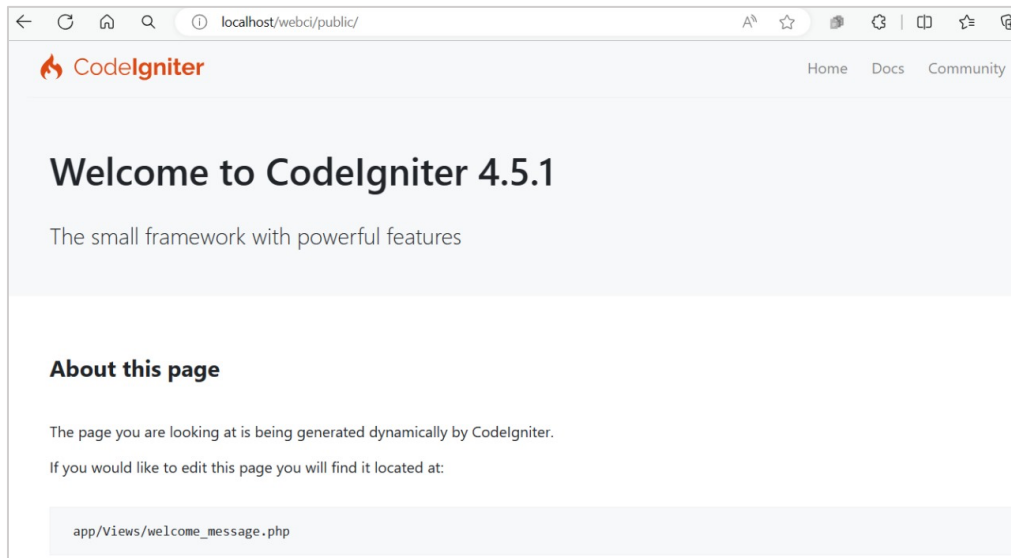
```
composer create-project codeigniter4/appstarter webci
```

Jika instalasi gagal dijalankan, buka `php.ini`, aktifkan `extension=intl` dengan menghapus tanda `;` di depannya, lalu simpan kembali file `php.ini`.

3. Jika instalasi sudah selesai, buka direktori tempat Anda membuat project tersebut. Pastikan sudah ada folder baru bernama `webci`. Di dalam folder tersebut terdapat folder dan file yang dibutuhkan untuk membuat project web dengan CodeIgniter.
4. Akses web tersebut dari browser. Jika Anda menggunakan Apache sebagai server, maka halaman web tersebut dapat dikunjungi dengan menambahkan `"public"` di akhir link url, contohnya:

```
http://localhost/webci/public/
```

Jika berhasil, maka akan muncul tampilan sebagai berikut.



Karena Codelgniter sudah memiliki internal server yang dapat digunakan untuk mempermudah menjalankan halaman web pada tahap development, maka selanjutnya kita akan menggunakan internal server ini. Untuk menjalankan internal server di Codelgniter, pada command prompt pindahkan path ke folder webci lalu ketikkan perintah berikut.

```
php spark serve
```

Secara default aplikasi Codelgniter akan berjalan pada port 8080 dengan alamat:

```
http://localhost:8080/
```

Selanjutnya, project `webci` dapat diakses melalui link tersebut. Jika Anda ingin mengganti port defaultnya, tambahkan argument pada `--port` seperti contoh berikut.

```
php spark serve --port 3000
```

5. Perhatikan file dan folder yang ada di dalam folder project `webci`. Di dalam folder tersebut terdapat file bernama `env`. Duplicate file tersebut, dan rename file barunya menjadi `.env`. Lalu buka file `.env`, dan ubah baris berikut:

```
# CI_ENVIRONMENT = production
```

menjadi:

```
CI_ENVIRONMENT = development
```

Buka project `webci` di browser pada tab baru, dan perhatikan apakah ada perubahan pada footernya? Baca dokumentasi mengenai hal ini pada link berikut.

[https://codeigniter.com/user\\_guide/general/configuration.html](https://codeigniter.com/user_guide/general/configuration.html)  
[https://codeigniter.com/user\\_guide/general/environments.html](https://codeigniter.com/user_guide/general/environments.html)

6. Buka kembali file `.env`, dan ubah baris berikut:

```
# app.baseURL = ''
```

menjadi:

```
app.baseURL = 'http://localhost:8080/'
```

Isi dari `baseURL` disesuaikan dengan link server yang dijalankan pada setiap environment. Karena pada saat ini Anda sedang menggunakan internal server CodeIgniter, maka gunakan `baseURL` `'http://localhost:8080/'`.

7. Folder inti dari project CodeIgniter yaitu folder `app`. Perhatikan isi dari folder `app` tersebut. Folder `Models`, `Views`, dan `Controllers` di dalam folder `app` digunakan untuk menerapkan MVC pattern, yaitu dengan memisahkan kode untuk model, view, dan controller. Silakan buka file `Home.php` pada folder `Controllers` dan perhatikan isi file tersebut, terdapat fungsi `index` sebagai berikut.

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

View atau halaman yang dipanggil pada fungsi `index` tersebut bernama `welcome_message`. Buka file `welcome_message` pada folder `Views` dan perhatikan isinya.

8. Selanjutnya Anda akan mengganti view `welcome_message` pada fungsi `index` dengan halaman sign-in yang telah anda buat pada kegiatan 12.4.1. Copy file `index.html` yang merupakan halaman html dari form sign-in menggunakan bootstrap dari project `web_signin`, paste ke folder `Views` pada project `webci`, ganti ekstensi filenya dari `index.html` menjadi `index.php`.

9. File resources yang dapat diakses publik seperti image, css, dan js dapat disimpan di folder `public`. Pada project `webci`, buka folder `public`. Buat sebuah folder baru di dalam folder `public` tersebut, beri nama `css`.
10. Copy file lainnya selain `index.html` pada folder `web_signin` ke dalam folder `public/css` yang baru saja anda buat.
11. Kemudian buka file `index.php` di folder `Views`, ubah referensi pada tag link css menjadi:

```
<link href="/css/bootstrap.min.css" rel="stylesheet">
<link href="/css/signin.css" rel="stylesheet">
```

Ubah juga path untuk memanggil logo Politeknik Statistika STIS.

12. Buka kembali file `Home.php` pada folder `Controllers` dan ubah view pada fungsi `index` dari `"welcome_message"` menjadi `"index"`. Kemudian akses `webci` dari browser, pastikan yang tampil adalah halaman sign-in. Jika anda melihat pesan error `"Can't find a route for ..."`, lakukan langkah-langkah berikut:

- Buka `Routes.php` pada folder **`app/config`**
- Tambahkan script berikut pada bagian `Router Setup`

```
$routes->setAutoRoute(true);
```

13. Selanjutnya Anda akan membuat model untuk mengakses tabel `user` yang Anda sudah buat pada praktikum sebelumnya. Ubah terlebih dahulu konfigurasi database pada `webci` dengan mengubah baris berikut di file `.env`

```
# database.default.hostname = localhost
# database.default.database = ci4
# database.default.username = root
# database.default.password = root
# database.default.DBDriver = MySQLi
```

menjadi:

```
database.default.hostname = localhost
database.default.database = <nama_database>
database.default.username = <username>
database.default.password = <password>
database.default.DBDriver = MySQLi
```

Sesuaikan isiannya dengan nama database, username, dan password yang Anda gunakan.



Kemudian buat file `UserModel.php` pada folder `Models` dengan isi kode sebagai berikut.

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class UserModel extends Model
{
    protected $table          = 'user';
    protected $primaryKey     = 'username';
}
```

14. Kemudian buat fungsi untuk sign in di Controller. Tambahkan kode fungsi berikut di `Home.php` pada folder `Controllers`.

```
public function signInAuth()
{
    $session = session();
    $userModel = new UserModel();
    $username = $this->request->getVar('username');
    $password = $this->request->getVar('password');

    $data = $userModel->where('username', $username)->first();

    if($data){
        $pass = $data['password'];
        if($password=== $pass){
            $ses_data = [
                'username' => $data['username'],
                'isSignedIn' => TRUE
            ];
            $session->set($ses_data);
            return redirect()->to('/Home/meeting');

        }else{
            $session->setFlashdata('msg', 'Password is
incorrect.');
```

Jangan lupa tambahkan kode berikut sebelum pendefinisian Class pada `Home.php` agar dapat mengakses `UserModel.php` tanpa menuliskan lokasinya.

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

15. Buat halaman view bernama `meeting.php` yang akan ditampilkan setelah user berhasil sign in.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <meta name="description" content="">
    <link href="/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <header class="navbar navbar-dark sticky-top bg-dark flex-md-
nowrap p-0 shadow">
      <div class="navbar-nav">
        <div class="nav-item text-nowrap">
          <a class="nav-link px-3" href="Signout">Sign out</a>
        </div>
      </div>
    </header>
    <div class="container-fluid">
      Hello, <?=$username;?>
      <h1>Data Meeting</h1>
    </div>
  </body>
</html>
```

Lalu buat controller untuk menampilkan view tersebut. Tambahkan fungsi `meeting` pada controller `Home.php`.

```
public function meeting()
{
    $session = session();
    $data = [
        'username' => $session->get('username')
    ];
    return view('meeting', $data);
}
```

16. Akses kembali project `webci` melalui browser. Pastikan halaman `sign in` sudah berfungsi dengan baik, menampilkan halaman `meeting` setelah berhasil sign in.

17. Selanjutnya, tambahkan pula fungsi `signout` pada controller `Home.php` yang berisi perintah untuk menghapus session dan kembali ke halaman sign in.

```
public function signout()
{
    $session = session();
    $session->destroy();
    return redirect()->to('/Home');
}
```

### 12.5. Responsi

Kerjakan sesuai dengan yang dijelaskan pada bagian Kegiatan Praktikum. Simpan tangkapan layar hasil pekerjaan Anda dalam file docx. Simpan ulang file tersebut dalam format pdf, dan beri nama dengan format `<<nim>>_modul12.pdf`, contoh: `192191234_modul12.pdf`. Kumpulkan file tersebut sebagai responsi melalui Google Classroom.