

Praktikum 1: PHP Framework (Codeigniter)

Tujuan

1. Mahasiswa mampu memahami konsep dasar Framework.
2. Mahasiswa mampu memahami konsep dasar MVC.
3. Mahasiswa mampu membuat program sederhana menggunakan Framework Codeigniter4.

Instruksi Praktikum

1. Persiapkan text editor misalnya **VSCode**.
2. Buat folder baru dengan nama **lab11_php_ci** pada docroot webserver (**htdocs**)
3. Ikuti langkah-langkah praktikum yang akan dijelaskan berikutnya.

Langkah-langkah Praktikum

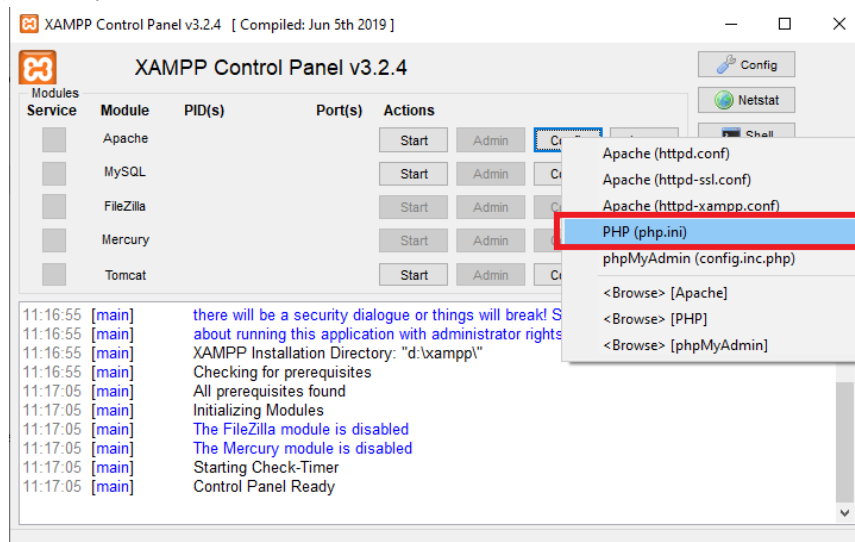
Persiapan

Sebelum memulai menggunakan Framework Codeigniter, perlu dilakukan konfigurasi pada webserver. Beberapa ekstensi PHP perlu diaktifkan untuk kebutuhan pengembangan Codeigniter 4.

Berikut beberapa ekstensi yang perlu diaktifkan:

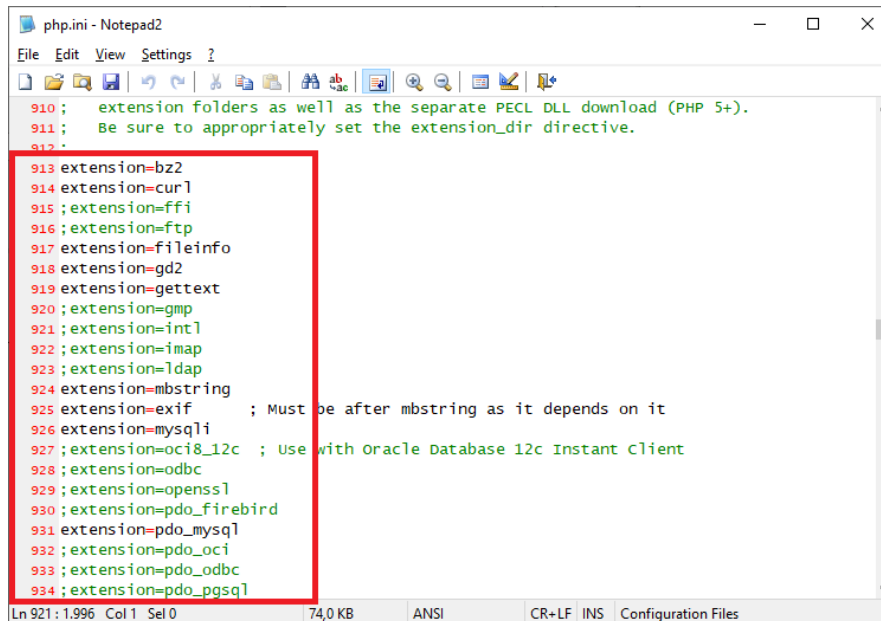
- **php-json** ekstension untuk bekerja dengan JSON;
- **php-mysqlnd** native driver untuk MySQL;
- **php-xml** ekstension untuk bekerja dengan XML;
- **php-intl** ekstensi untuk membuat aplikasi multibahasa;
- **libcurl** (opsional), jika ingin pakai Curl.

Untuk mengaktifkan ekstensi tersebut, melalui **XAMPP Control Panel**, pada bagian Apache klik **Config -> PHP.ini**



Gambar 11.1 Konfigurasi PHP

Pada bagian extention, hilangkan tanda ; (titik koma) pada ekstensi yang akan diaktifkan. Kemudian simpan kembali filenya dan restart Apache web server.

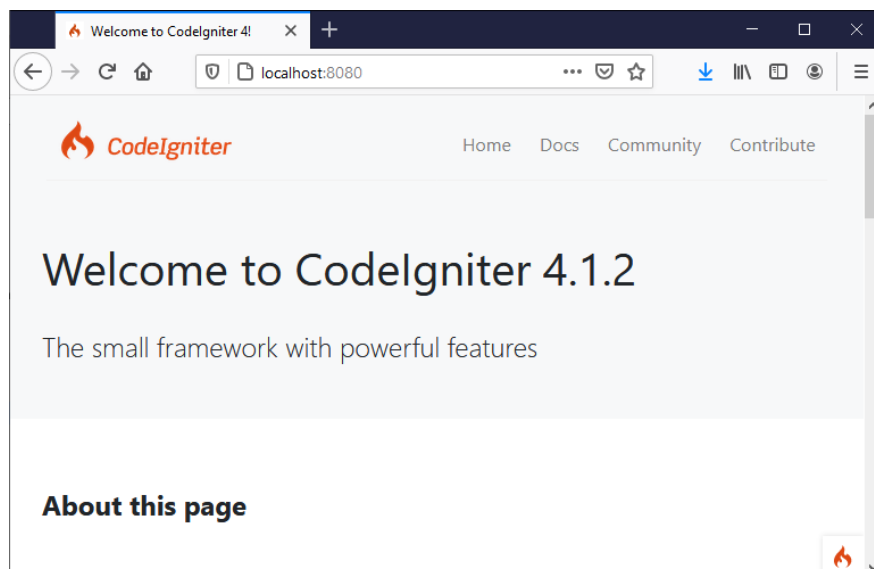


Gambar 11.2 Ekstensi PHP

Instalasi Codeigniter 4

Untuk melakukan instalasi Codeigniter 4 dapat dilakukan dengan dua cara, yaitu cara manual dan menggunakan *composer*. Pada praktikum ini kita menggunakan cara manual.

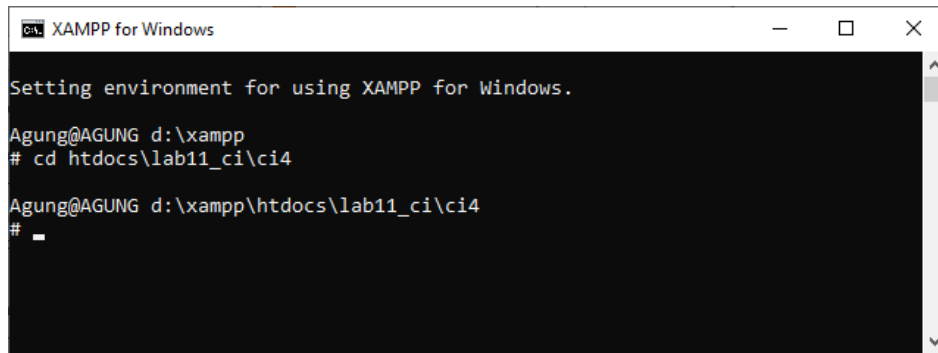
- Unduh **Codeigniter** dari website <https://codeigniter.com/download>
- Ekstrak file zip Codeigniter ke direktori **htdocs/lab11_ci**.
- Ubah nama direktori **framework-4.x.xx** menjadi **ci4**.
- Buka browser dengan alamat http://localhost/lab11_ci/ci4/public/



Gambar 11.3 Tampilan Codeigniter 4.

Menjalankan CLI (Command Line Interface)

Codeigniter 4 menyediakan CLI untuk mempermudah proses development. Untuk mengakses CLI buka terminal/command prompt.



```
Setting environment for using XAMPP for Windows.

Agung@AGUNG d:\xampp
# cd htdocs\lab11_ci\ci4

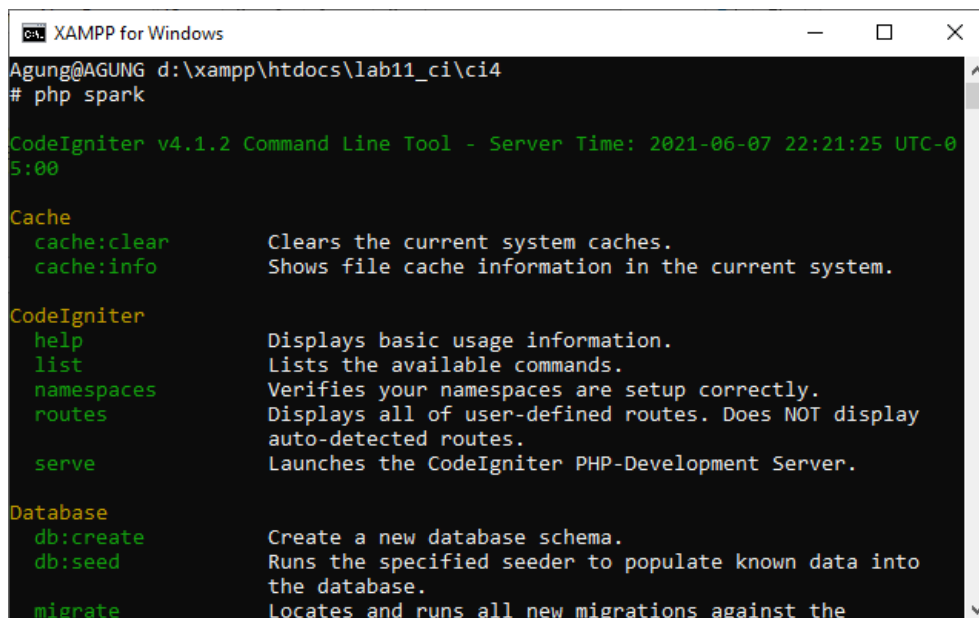
Agung@AGUNG d:\xampp\htdocs\lab11_ci\ci4
#
```

Gambar 11.4 Tampilan Command Prompt

Arahkan lokasi direktori sesuai dengan direktori kerja project dibuat (xampp/htdocs/lab11_ci/ci4/)

Perintah yang dapat dijalankan untuk memanggil CLI Codeigniter adalah:

```
php spark
```



```
Agung@AGUNG d:\xampp\htdocs\lab11_ci\ci4
# php spark

CodeIgniter v4.1.2 Command Line Tool - Server Time: 2021-06-07 22:21:25 UTC-05:00

Cache
  cache:clear      Clears the current system caches.
  cache:info       Shows file cache information in the current system.

CodeIgniter
  help             Displays basic usage information.
  list             Lists the available commands.
  namespaces       Verifies your namespaces are setup correctly.
  routes           Displays all of user-defined routes. Does NOT display auto-detected routes.
  serve            Launches the CodeIgniter PHP-Development Server.

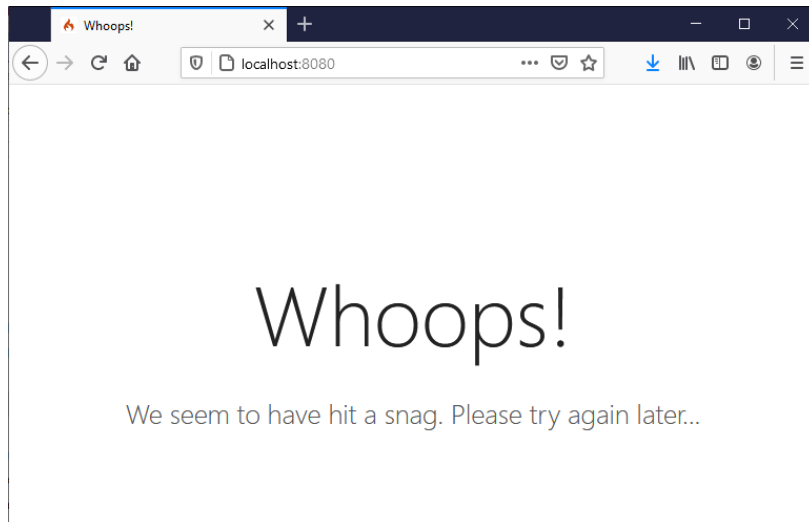
Database
  db:create        Create a new database schema.
  db:seed          Runs the specified seeder to populate known data into the database.
  migrate          Locates and runs all new migrations against the
```

Gambar 11.5 Perintah CLI

Mengaktifkan Mode Debugging

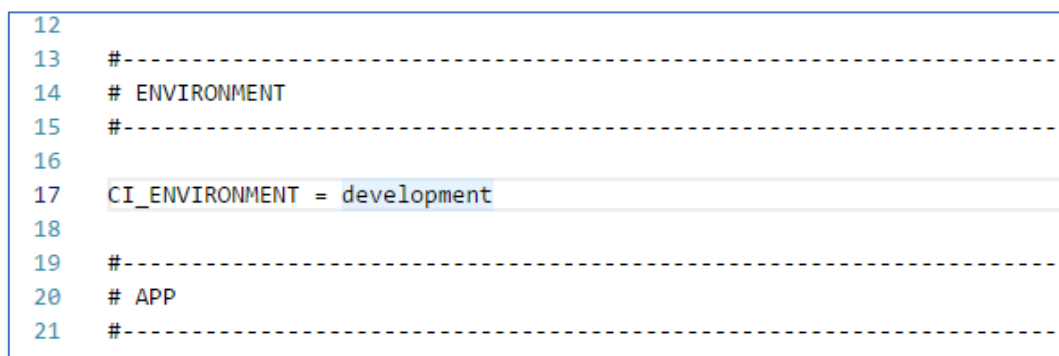
Codeigniter 4 menyediakan fitur **debugging** untuk memudahkan developer untuk mengetahui pesan error apabila terjadi kesalahan dalam membuat kode program.

Secara default fitur ini belum aktif. Ketika terjadi error pada aplikasi akan ditampilkan pesan kesalahan seperti berikut.



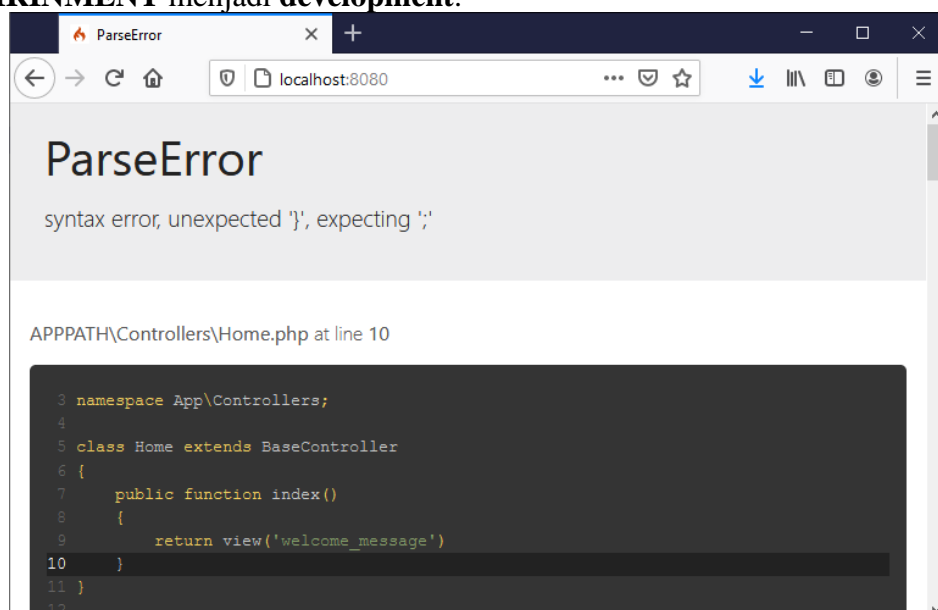
Gambar 11.6 CI Error

Semua jenis error akan ditampilkan sama. Untuk memudahkan mengetahui jenis errornya, maka perlu diaktifkan mode debugging dengan mengubah nilai konfigurasi pada environment variable **CI_ENVIRINMENT** menjadi **development**.



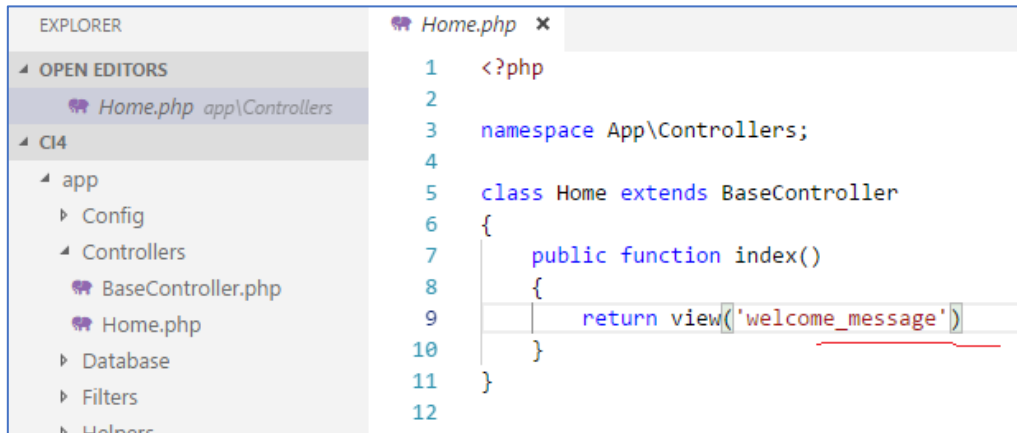
Gambar 11.7 Konfigurasi CI

Ubah nama file **env** menjadi **.env** kemudian buka file tersebut dan ubah nilai variable **CI_ENVIRONMENT** menjadi **development**.



Gambar 11.8 Error

Contoh error yang terjadi. Untuk mencoba error tersebut, ubah kode pada file **app/Controller/Home.php** hilangkan titik koma pada akhir kode.



Gambar 11.9 Kode Home

Struktur Direktori

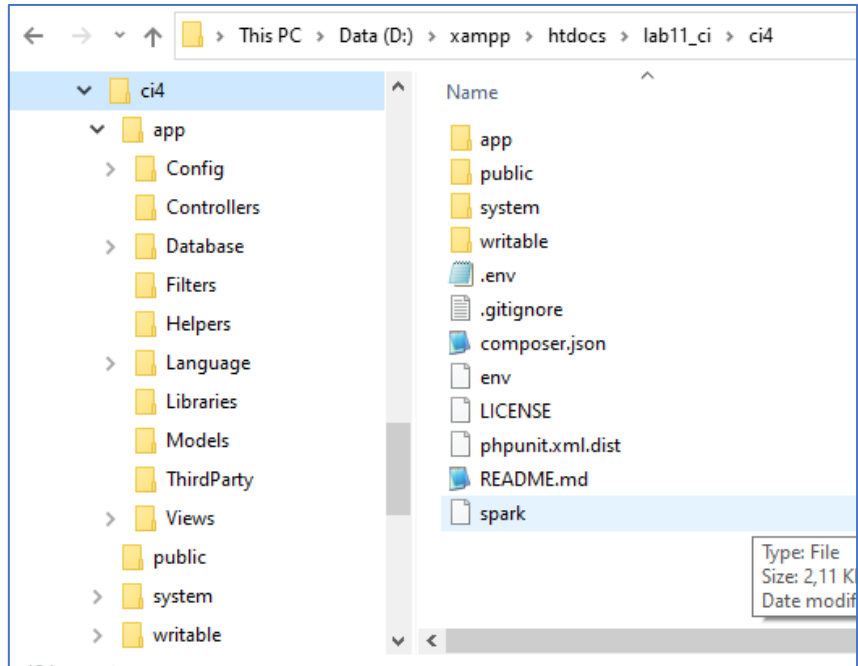
Untuk lebih memahami Framework Codeigniter 4 perlu mengetahui struktur direktori dan file yang ada. Buka pada **Windows Explorer** atau dari **Visual Studio Code -> Open Folder**.

Terdapat beberapa direktori dan file yang perlu dipahami fungsi dan kegunaannya.

- **.github** folder ini kita butuhkan untuk konfigurasi repo github, seperti konfigurasi untuk build dengan github action;
- **app** folder ini akan berisi kode dari aplikasi yang kita kembangkan;
- **public** folder ini berisi file yang bisa diakses oleh publik, seperti file index.php, robots.txt, favicon.ico, ads.txt, dll;
- **tests** folder ini berisi kode untuk melakukan testing dengan PHPUnit;
- **vendor** folder ini berisi library yang dibutuhkan oleh aplikasi, isinya juga termasuk kode core dari system CI.
- **writable** folder ini berisi file yang ditulis oleh aplikasi. Nantinya, kita bisa pakai untuk menyimpan file yang di-upload, logs, session, dll.

Sedangkan file-file yang berada pada root direktori CI sebagai berikut.

- **.env** adalah file yang berisi variabel environment yang dibutuhkan oleh aplikasi.
- **.gitignore** adalah file yang berisi daftar nama file dan folder yang akan diabaikan oleh Git.
- **build** adalah script untuk mengubah versi codeigniter yang digunakan. Ada versi release (stabil) dan development (labil).
- **composer.json** adalah file JSON yang berisi informasi tentang proyek dan daftar library yang dibutuhkannya. File ini digunakan oleh Composer sebagai acuan.
- **composer.lock** adalah file yang berisi informasi versi dari library yang digunakan aplikasi.
- **license.txt** adalah file yang berisi penjelasan tentang lisensi Codeigniter;
- **phpunit.xml.dist** adalah file XML yang berisi konfigurasi untuk PHPUnit.
- **README.md** adalah file keterangan tentang codebase CI. Ini biasanya akan dibutuhkan pada repo github atau gitlab.
- **spark** adalah program atau script yang berfungsi untuk menjalankan server, generate kode, dll.



Gambar 11.10 Struktur Direktori CI4

Fokus kita pada folder **app**, dimana folder tersebut adalah area kerja kita untuk membuat aplikasi. Dan folder **public** untuk menyimpan aset web seperti css, gambar, javascript, dll.

Memahami Konsep MVC

Codeigniter menggunakan konsep MVC. MVC merupakan singkatan dari *Model-View-Controller*. MVC merupakan konsep arsitektur yang umum digunakan dalam pengembangan aplikasi. Konsep MVC adalah memisahkan kode program berdasarkan logic proses, data, dan tampilan. Untuk logic proses diletakkan pada direktori Contoller, Objek data diletakkan pada direktori Model, dan desain tampilan diletakkan pada direktori View.

Codeigniter menggunakan konsep pemrograman berorientasi objek dalam mengimplementasikan konsep MVC.

Model merupakan kode program yang berisi pemodelan data. Data dapat berupa database ataupun sumber lainnya.

View merupakan kode program yang berisi bagian yang menangani terkait tampilan user interface sebuah aplikasi. didalam aplikasi web biasanya pasti akan berhubungan dengan html dan css.

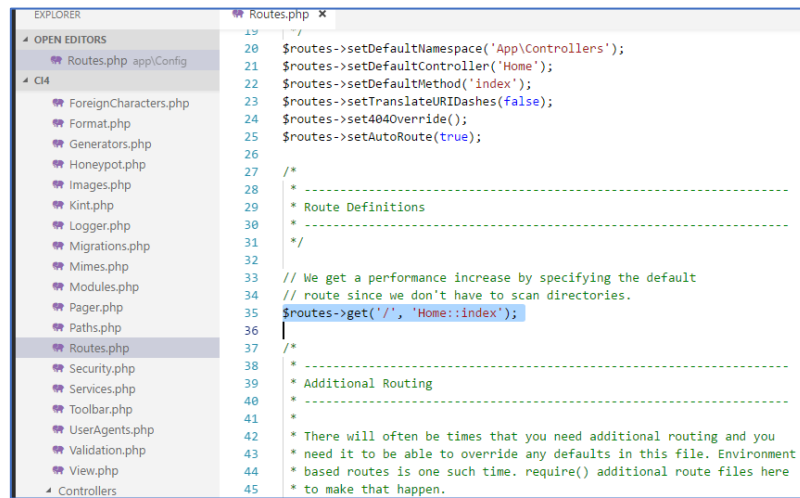
Controller merupakan kode program yang berkaitan dengan logic proses yang menghubungkan antara view dan model. Controller berfungsi untuk menerima request dan data dari user kemudian diproses dengan menghubungkan bagian model dan view.

Routing dan Controller

Routing merupakan proses yang mengatur arah atau rute dari request untuk menentukan fungsi/bagian mana yang akan memproses request tersebut. Pada framework CI4, routing bertujuan untuk menentukan Controller mana yang harus merespon sebuah request. Controller adalah class atau script yang bertanggung jawab merespon sebuah request.

Pada Codeigniter, request yang diterima oleh file index.php akan diarahkan ke Router untuk kemudian oleh router tersebut diarahkan ke Controller.

Router terletak pada file **app/config/Routes.php**



Gambar 11.11 Routes.php

Pada file tersebut kita dapat mendefinisikan route untuk aplikasi yang kita buat.

Contoh:

```
$routes->get('/', 'Home::index');
```

Kode tersebut akan mengarahkan rute untuk halaman home.

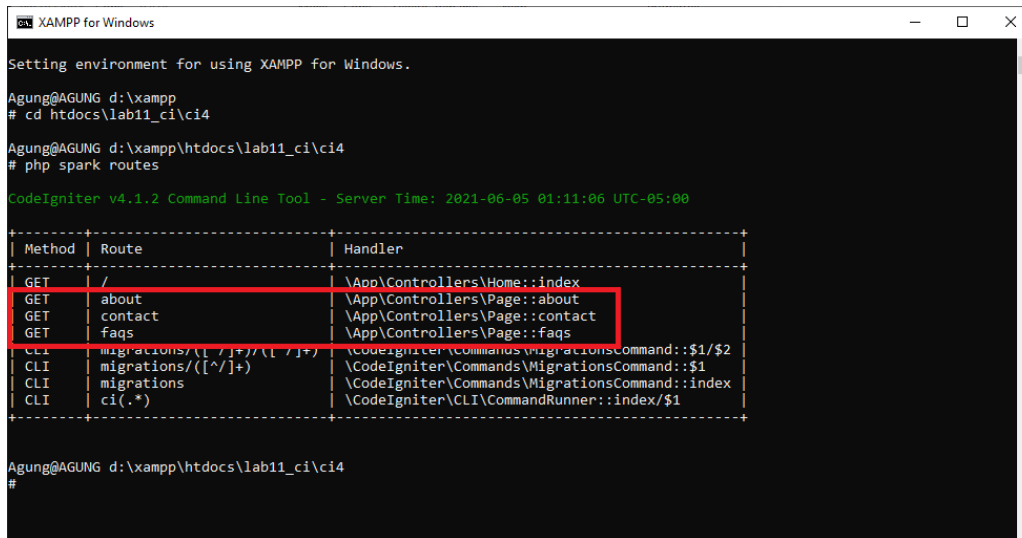
Membuat Route Baru.

Tambahkan kode berikut di dalam **Routes.php**

```
$routes->get('/about', 'Page::about');  
$routes->get('/contact', 'Page::contact');  
$routes->get('/faqs', 'Page::faqs');
```

Untuk mengetahui route yang ditambahkan sudah benar, buka CLI dan jalankan perintah berikut.

```
php spark routes
```



```
Setting environment for using XAMPP for Windows.
Agung@AGUNG d:\xampp
# cd htdocs\lab11_ci\ci4
Agung@AGUNG d:\xampp\htdocs\lab11_ci\ci4
# php spark routes

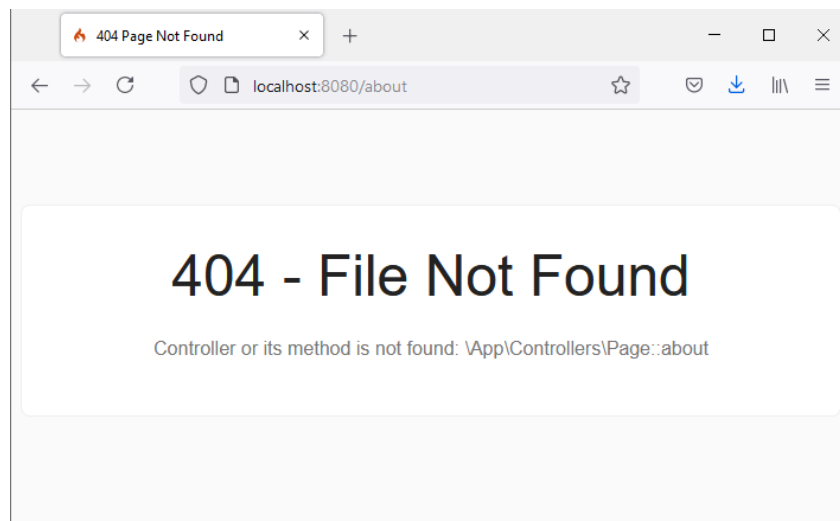
CodeIgniter v4.1.2 Command Line Tool - Server Time: 2021-06-05 01:11:06 UTC-05:00

+-----+-----+-----+
| Method | Route | Handler |
+-----+-----+-----+
| GET | / | \App\Controllers\Home::index |
| GET | about | \App\Controllers\Page::about |
| GET | contact | \App\Controllers\Page::contact |
| GET | faqs | \App\Controllers\Page::faqs |
| CLI | migrations/([ / ]+)([ / ]+) | \CodeIgniter\Commands\MigrationsCommand::$1/$2 |
| CLI | migrations/([ / ]+) | \CodeIgniter\Commands\MigrationsCommand::$1 |
| CLI | migrations | \CodeIgniter\Commands\MigrationsCommand::index |
| CLI | ci(.*) | \CodeIgniter\CLI\CommandRunner::index/$1 |
+-----+-----+-----+

Agung@AGUNG d:\xampp\htdocs\lab11_ci\ci4
#
```

Gambar 11.12 Tampilan CLI

Selanjutnya coba akses route yang telah dibuat dengan mengakses alamat url <http://localhost:8080/about>



Gambar 11.13 Tampilan error page.

Ketika diakses akan muncul tampilan error 404 file not found, itu artinya file/page tersebut tidak ada. Untuk dapat mengakses halaman tersebut, harus dibuat terlebih dahulu Controller yang sesuai dengan routing yang dibuat yaitu Controller Page.

Membuat Controller

Selanjutnya adalah membuat Controller Page. Buat file baru dengan nama `page.php` pada direktori Controller kemudian isi kodenya seperti berikut.

```
<?php

namespace App\Controllers;

class Page extends BaseController
{
    public function about()
```

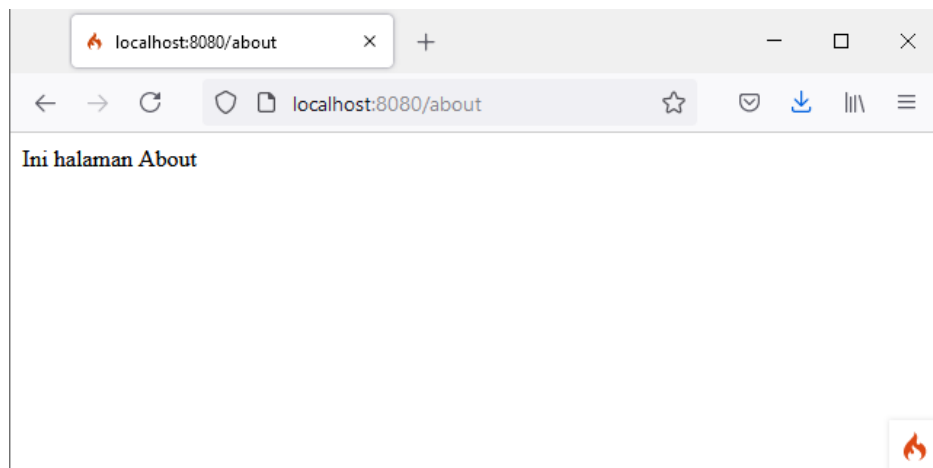


```
{
    echo "Ini halaman About";
}

public function contact()
{
    echo "Ini halaman Contact";
}

public function faqs()
{
    echo "Ini halaman FAQ";
}
}
```

Selanjutnya refresh Kembali browser, maka akan ditampilkan hasilnya yaotu halaman sudah dapat diakses.



Gambar 11.14 Tampilan Halaman About

Auto Routing

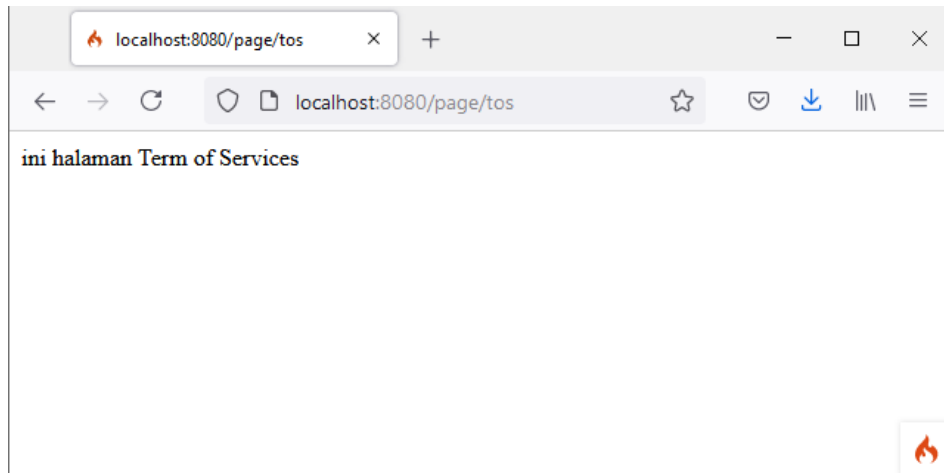
Secara default fitur *autoroute* pada Codeigniter sudah aktif. Untuk mengubah status autoroute dapat mengubah nilai variabelnya. Untuk menonaktifkan ubah nilai **true** menjadi **false**.

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

Tambahkan method baru pada **Controller Page** seperti berikut.

```
public function tos()
{
    echo "ini halaman Term of Services";
}
```

Method ini belum ada pada **routing**, sehingga cara mengaksesnya dengan menggunakan alamat: <http://localhost:8080/page/tos>



Gambar 11.15 Tampilan autoroute.

Membuat View

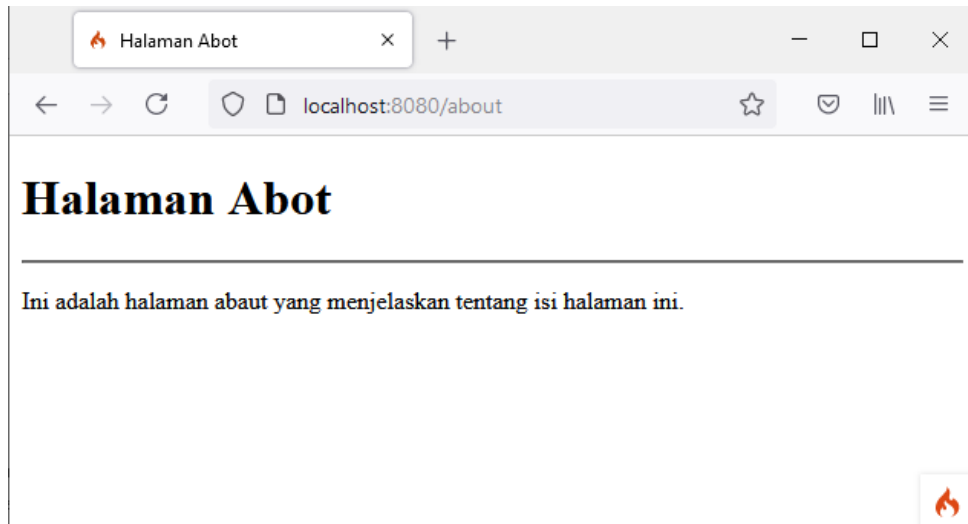
Selanjutnya adalah membuat view untuk tampilan web agar lebih menarik. Buat file baru dengan nama `about.php` pada direktori view (**app/view/about.php**) kemudian isi kodenya seperti berikut.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title><?= $title; ?></title>
</head>
<body>
    <h1><?= $title; ?></h1>
    <hr>
    <p><?= $content; ?></p>
</body>
</html>
```

Ubah **method about** pada class **Controller Page** menjadi seperti berikut:

```
public function about()
{
    return view('about', [
        'title' => 'Halaman Abot',
        'content' => 'Ini adalah halaman about yang menjelaskan tentang isi
halaman ini.'
    ]);
}
```

Kemudian lakukan refresh pada halaman tersebut.

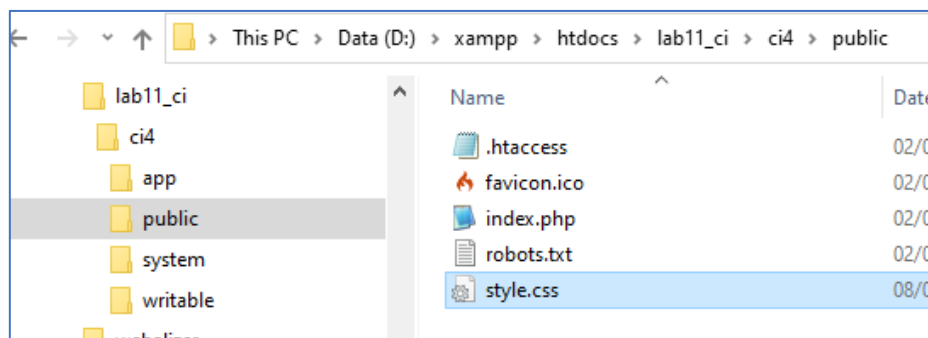


Gambar 11.16 Halaman about.

Membuat Layout Web dengan CSS

Pada dasarnya layout web dengan css dapat diimplmentasikan dengan mudah pada codeigniter. Yang perlu diketahui adalah, pada Codeigniter 4 file yang menyimpan asset css dan javascript terletak pada direktori **public**.

Buat file css pada direktori **public** dengan nama **style.css** (copy file dari praktikum **lab4_layout**). Kita akan gunakan layout yang pernah dibuat pada praktikum 4.



Gambar 11.17 Direktori asset

Kemudian buat folder **template** pada direktori **view** kemudian buat file **header.php** dan **footer.php**

File `app/view/template/header.php`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title><?= $title; ?></title>
  <link rel="stylesheet" href="<?= base_url('/style.css'); ?>">
</head>
<body>
  <div id="container">
    <header>
      <h1>Layout Sederhana</h1>
```

```
</header>
<nav>
  <a href="<?= base_url('/');?>" class="active">Home</a>
  <a href="<?= base_url('/artikel');?>">Artikel</a>
  <a href="<?= base_url('/about');?>">About</a>
  <a href="<?= base_url('/contact');?>">Kontak</a>
</nav>
<section id="wrapper">
  <section id="main">
```

File **app/view/template/footer.php**

```
</section>
<aside id="sidebar">
  <div class="widget-box">
    <h3 class="title">Widget Header</h3>
    <ul>
      <li><a href="#">Widget Link</a></li>
      <li><a href="#">Widget Link</a></li>
    </ul>
  </div>
  <div class="widget-box">
    <h3 class="title">Widget Text</h3>
    <p>Vestibulum lorem elit, iaculis in nisl volutpat,
malesuada tincidunt arcu. Proin in leo fringilla, vestibulum mi porta,
faucibus felis. Integer pharetra est nunc, nec pretium nunc pretium ac.</p>
  </div>
</aside>
</section>
<footer>
  <p>&copy; 2021 - Universitas Pelita Bangsa</p>
</footer>
</div>
</body>
</html>
```

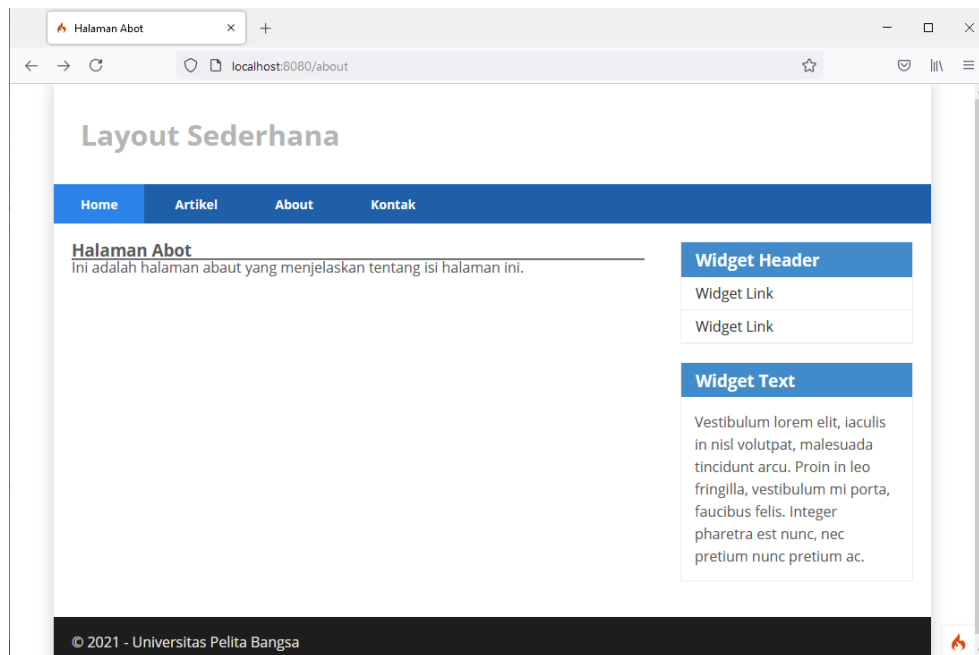
Kemudian ubah file **app/view/about.php** seperti berikut.

```
<?= $this->include('template/header'); ?>

<h1><?= $title; ?></h1>
<hr>
<p><?= $content; ?></p>

<?= $this->include('template/footer'); ?>
```

Selanjutnya refresh tampilan pada alamat <http://localhost:8080/about>



Gambar 11.18

Pertanyaan dan Tugas

Lengkapi kode program untuk menu lainnya yang ada pada Controller Page, sehingga semua link pada navigasi header dapat menampilkan tampilan dengan layout yang sama.

Laporan Praktikum

1. Buatlah **repository** baru dengan nama **Lab7Web**.
2. Kerjakan semua latihan yang diberikan sesuai urutannya.
3. Screenshot setiap perubahannya.
4. Buatlah 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