

## Pertemuan 6

# Intro - Codeigniter 3

# Codeigniter - Prerequisite



**Prerequisite:** Sebelum menginstall Codeigniter, kita membutuhkan beberapa software yang harus dipersiapkan.

## Steps:

1. Download and Install XAMPP: [Link](#)
2. Download and Install Text Editor: [VSCode Text Editor](#)
3. Start XAMPP (Apache and Mysql)
4. Access [localhost](#) from browser

# Codeigniter - Prerequisite



# Codeigniter - Installation

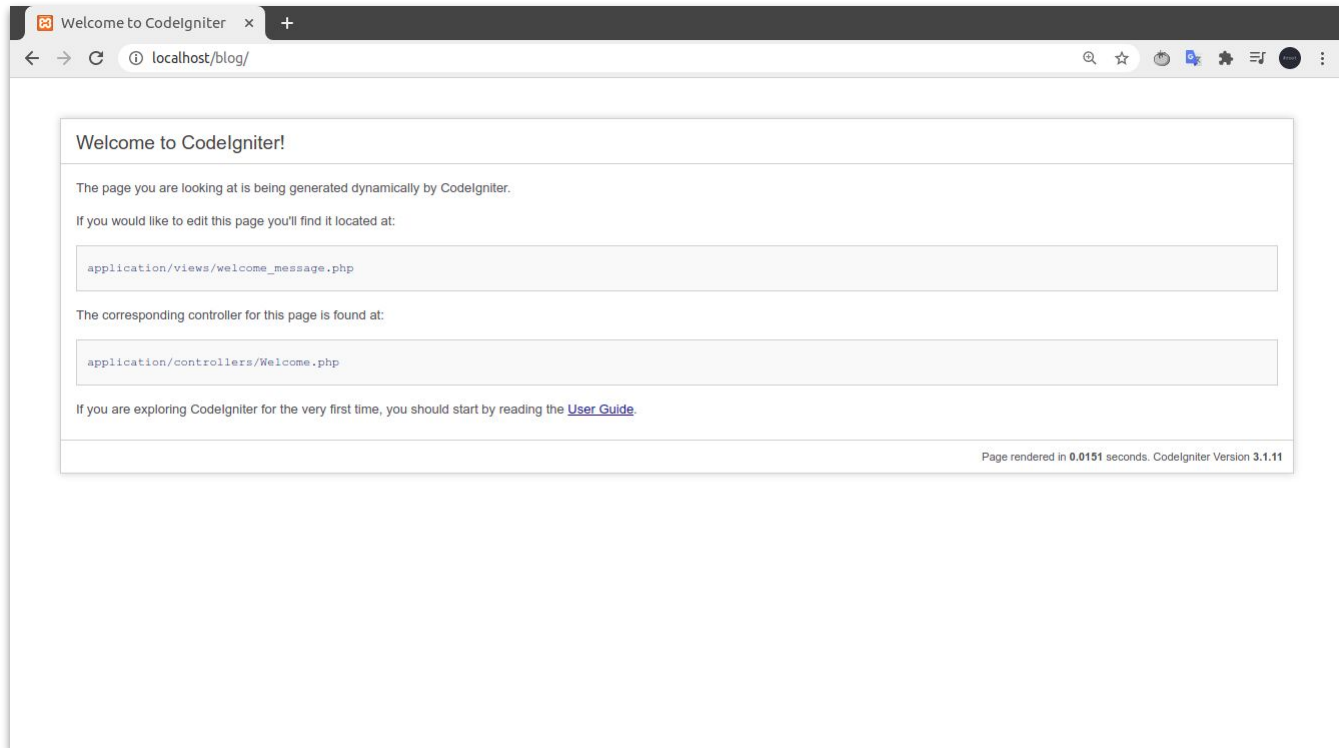


**Installation:** Codeigniter merupakan sebuah framework berbasis PHP, untuk menggunakannya kita harus mendownload dan menginstall terlebih dahulu.

## Steps:

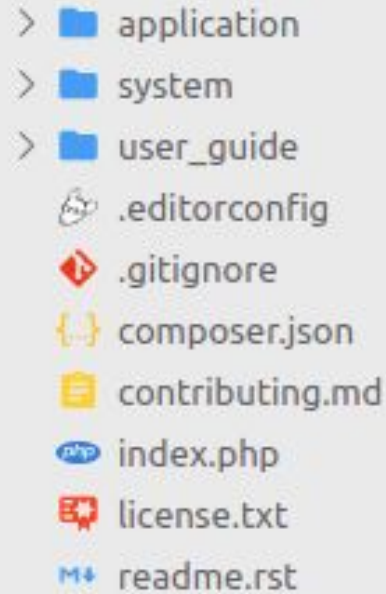
1. Download Codeigniter 3: [Link](#)
2. Extract file zip Codeigniter 3
3. Rename folder as you wish, example **blog**
4. Move the folder (blog) to **htdocs** xampp
5. Access **localhost/blog** from browser

# Codeigniter - Installation



# Codeigniter - Folder Structure

---



A screenshot of a file explorer window showing the root directory of a Codeigniter project. The items are listed as follows:

- > application (folder)
- > system (folder)
- > user\_guide (folder)
- .editorconfig (file)
- .gitignore (file)
- composer.json (file)
- contributing.md (file)
- index.php (file)
- license.txt (file)
- readme.rst (file)

# Codeigniter - Folder Structure



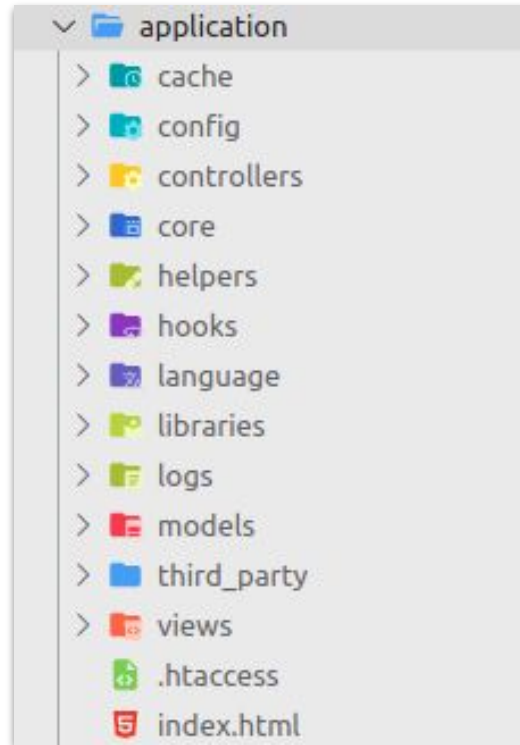
**Folder Structure:** Codeigniter berisi banyak file dan folder yang dikelola atau terstruktur berdasarkan tujuannya.

## Folder Structure:

1. **application:** berisi code aplikasi, kita akan menulis kode di sini.
2. **system:** berisi kode inti atau sistem dari Codeigniter.
3. **user\_guide:** berisi dokumentasi.
4. **index.php:** file pertama yang diakses.
5. **etc:** .gitignore, composer.json, contributing.md, license.txt, readme.rst

# Codeigniter - application folder

---





# Codeigniter - application folder



**application:** folder ini berisi kode yang berkaitan development codeigniter. Kita akan melakukan pengembangan di dalam folder ini.

**application:**

1. **controller:** mengatur proses atau logic aplikasi
2. **models:** mengatur terkait data atau database
3. **views:** mengatur dan membangun tampilan aplikasi
4. etc: cache, config, core, helpers, hooks, language, libraries

# Codeigniter - Routing



**Routing:** Routing merupakan konsep yang digunakan untuk mengatur jalur aplikasi. Routing di CodeIgniter berdasarkan nama class, method, dan parameter.

# Codeigniter - Routing



localhost/blog/index.php/welcome/index/ID

welcome: merepresentasikan class

index: merepresentasikan function

ID: merepresentasikan parameter

# Codeigniter - Routing

---



```
# buat method about di Controller Welcome  
public function about() {  
    echo "Halaman About";  
}
```

# Codeigniter - Controller



## Studi Kasus

1. Ingin membuat fitur blog
2. Fitur blog dapat menampilkan blog, menambahkan blog baru, dan mengedit blog.

## Problem

1. Kita tidak tepat menangani hal tersebut di Controller Welcome karena berbeda business logic.

## Solusi

1. Kita dapat membuat controller baru untuk mengatur Fitur blog (Business Logic)

# Codeigniter - Controller




**Controller:** Controller digunakan untuk mengatur proses atau logika dari aplikasi (Business Logic).

## Steps:

1. Buat Controller Blog
2. Buat method index dan add
3. Pada method index, tampilkan output “Ini Halaman Blog”
4. Pada method add, tampilkan output “Halaman Tambah Blog”
5. Akses halaman blog: `localhost/blog/index.php/blog`
6. Akses halaman tambah blog: `localhost/blog/index.php/blog/add`

# Codeigniter - Controller



```
class Blog extends CI_Controller {  
    public function index() {  
        echo "Halaman Blog";  
    }  
  
    public function add() {  
        echo "Halaman Tambah Blog";  
    }  
}
```

# Codeigniter - View



## Studi Kasus

1. Kita ingin membuat halaman blog yang menampilkan data-data blog
2. Kita ingin membuat form untuk menambah blog

## Problem

1. Kita tidak menuliskan tampilan di dalam Controller karena Controller bertugas untuk mengelola proses atau business logic.

## Solusi

1. Kita membuat view atau tampilan yang terpisah dari Controller. Controller hanya bertujuan merender view yang sudah dibuat



# Codeigniter - View



**View:** View digunakan untuk membuat tampilan dari aplikasi. Controller hanya bertugas untuk merender view yang sudah dibuat.

## Steps:

1. Buat folder blog di dalam views. Folder blog digunakan untuk menyimpan view yang berkaitan dengan blog.
2. Buat file index.php di folder views/blog, lalu buat tampilan halaman blog.
3. Buat file add.php di folder views/blog, lalu buat form untuk menambah blog.

# Codeigniter - View



*# file index.php di folder views/blog*

*# kodingan html*

**<body>**

**<h2>Halaman Blog</h2>**

**<h3>Tutorial CodeIgniter 3</h3>**

**<p>CodeIgniter adalah Framework PHP.</p>**

**</body>**

*# kodingan html*

# Codeigniter - View

---



*# file add.php di folder views/blog*

*# kodingan html*

`<body>`

`<h2>Tambah Blog</h2>`

`<form action="">`

`Nama Blog: <input type="text">`

`</form>`

`</body>`

*# kodingan html*

# Codeigniter - View



## Steps:

1. Kita akan menghapus echo di method index dan add pada controller blog, lalu menggantikannya dengan merender view yang sudah dibuat.
2. Pada method index, kita merender view index.php yang berada di folder views/blog:  
`$this->load->view("blog/index")`
3. Pada method add, kita merender view add.php yang berada di folder views/blog:  
`$this->load->view("add/index")`
4. Akses halaman blog: localhost/blog/index.php/blog
5. Akses halaman tambah blog: localhost/blog/index.php/blog/add

# Codeigniter - View

---



```
class Blog extends CI_Controller {  
    public function index() {  
        $this->load->view("blog/index");  
    }  
  
    public function add() {  
        $this->load->view("blog/add");  
    }  
}
```

# Referensi



- Documentation CodeIgniter 3: [Installation](#)
- Documentation CodeIgniter 3: [Tutorial](#)
- M Fikri: [Tutorial CodeIgniter \(Full Tutorial\)](#)
- Petani Kode: [Install CodeIgniter](#)
- Petani Kode: [Konsep Dasar CodeIgniter](#)