

# **LAPORAN PRAKTIKUM**

"Framework CI"



Ahmad	230210501020
Adven Dwi Putra	230210501030
Adniel Pascal Ancelo	230210501072

**PRODI TEKNIK KOMPUTER**  
**JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER**  
**UNIVERSITAS NEGERI MAKASSAR**  
**TAHUN 2024**

## DAFTAR ISI

MENGENAL FRAMEWORK CODEIGNETER.....	3
INSTALASI SOFTWARE PENDUKUNG .....	5
KONFIGURASI ATAU SETTING FRAMEWORK CODEIGNITER .....	13
MENGENAL CRUD PADA FRAMEWORDK CODEIGNITER 4 .....	19
MENGELOLA DATA BERITA.....	40

## **MODUL 1**

### **MENGENAL FRAMEWORK CODEIGNITER**

#### **TUJUAN**

1. Mahasiswa mengenal framework codeigniter

#### **TUGAS PENDAHULUAN**

1. Mahasiswa dapat mengenal framework codeigniter
2. Mahasiswa dapat mengenal fungsi, kelebihan dan kekurangan CodeIgniter

#### **TEORI**

##### **A. CodeIgniter**

CodeIgniter adalah salah satu framework PHP yang digunakan untuk membangun aplikasi web dinamis dengan cepat. Framework ini menyediakan struktur dasar untuk pengembangan aplikasi, sehingga mempercepat proses penulisan kode. CodeIgniter dirancang ringan dan mudah dipelajari, cocok bagi pengembang yang menginginkan framework dengan kinerja tinggi dan dokumentasi yang lengkap. CodeIgniter menggunakan arsitektur MVC (Model-View-Controller), yang membantu memisahkan logika bisnis, tampilan, dan data dalam aplikasi.

##### **B. Framework**

Framework adalah kerangka kerja atau kumpulan pustaka (library) dan alat bantu yang menyediakan struktur dasar bagi pengembangan aplikasi atau perangkat lunak. Framework membantu pengembang dengan menyediakan modul-modul umum dan pola desain yang sudah teruji, sehingga mereka dapat fokus pada fitur unik aplikasi tanpa harus membangun semuanya dari awal. Dalam pengembangan web, framework juga menyediakan keamanan, kecepatan, dan struktur kode yang teratur.

##### **C. Fungsi dari CodeIgniter**

CodeIgniter berfungsi untuk mempercepat proses pengembangan aplikasi web dengan menyediakan komponen standar dan sistem MVC. Dengan CodeIgniter, pengembang dapat menggunakan berbagai fitur seperti manajemen sesi, manipulasi database, validasi formulir, dan keamanan input, tanpa harus menulis semuanya dari awal. Framework ini membantu memudahkan pengelolaan dan pemeliharaan kode, serta meningkatkan performa aplikasi melalui struktur yang terorganisir.

##### **D. Kelebihan dan Kekurangan CodeIgniter**

###### **Kelebihan :**

- Ringan dan Cepat: CodeIgniter dikenal sebagai salah satu framework PHP yang ringan dan cepat.
- Mudah Dipelajari : CodeIgniter memiliki kurva belajar yang rendah, dengan dokumentasi lengkap, sehingga mudah dipahami pemula.
- Bebas Struktur yang Kaku : Framework ini tidak terlalu ketat dalam aturan penulisan kode, memberikan fleksibilitas pada pengembang.
- Komunitas Besar : Dengan banyaknya pengguna, mudah menemukan bantuan atau sumber daya untuk memecahkan masalah.

**Kekurangan :**

- Kurangnya Fitur Modern : CodeIgniter kurang mendukung fitur-fitur modern yang sudah ada pada framework lain seperti Laravel atau Symfony.
- Keamanan Bawaan yang Rendah : Untuk aplikasi yang membutuhkan keamanan tingkat tinggi, pengaturan tambahan mungkin diperlukan.
- Tidak Fully Compliant dengan MVC : Meski mengikuti arsitektur MVC, implementasinya pada CodeIgniter tidak seketat pada framework lain.

## MODUL 2

### INSTALASI SOFTWARE PENDUKUNG

#### TUJUAN

1. Mahasiswa mampu melakukan instalasi software pendukung dengan baik

#### PENDAHULUAN

1. Software pendukung yang dibutuhkan untuk mengerjakan sebuah aplikasi website
2. Mahasiswa mampu secara mandiri melakukan instalasi Xampp dan Text Visual Studio Code
3. Mahasiswa Mampu menambahkan Packages pada text editor yang digunakan pada PC atau laptop masing-masing

#### ALAT DAN BAHAN

1. Laptop
2. Internet
3. Browser
4. Mouse

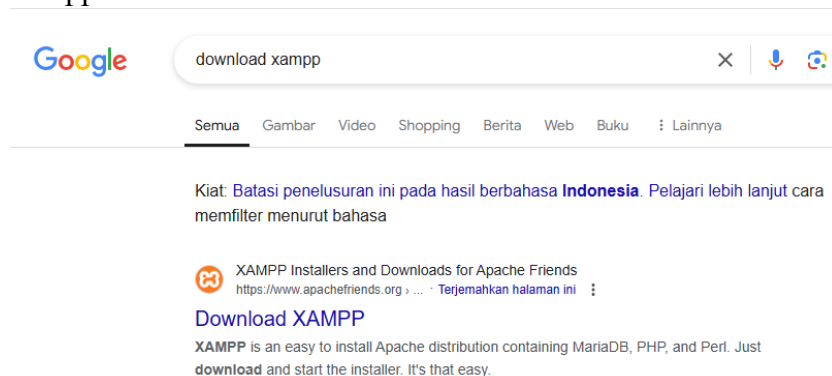
#### TEORI DASAR

Xampp sebagai software pendukung yang kita gunakan (Habibi & Sandi, 2020) Xampp merupakan paket perangkat lunak (software) yang tersedia secara gratis sehingga bebas untuk digunakan tanpa perlu menggunakan licence dari pengembang software. XAMPP berfungsi sebagai server yang berdiri sendiri (localhost) yang terdiri dari Apache HTTP Server, MySQL database dan penerjemah Bahasa yang ditulis dengan Bahasa pemrograman PHP dan Perl, XAMPP saat ini dikembangkan oleh perusahaan apache friends yang biasanya digunakan untuk simulasi pengembangan website.

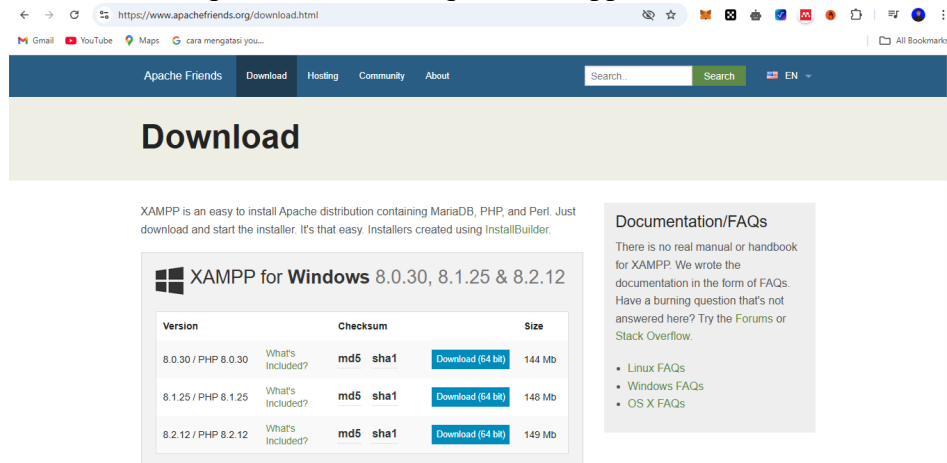
#### LANGKAH KERJA

##### A. Instalasi Xampp.

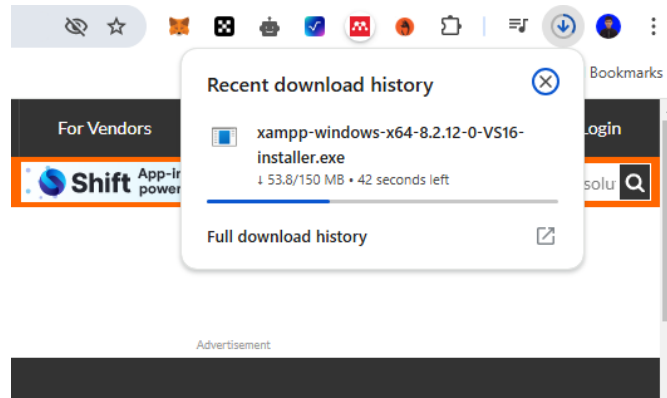
1. Buka Google Chrome dan ketik di pencarian “download xampp” untuk download xampp.



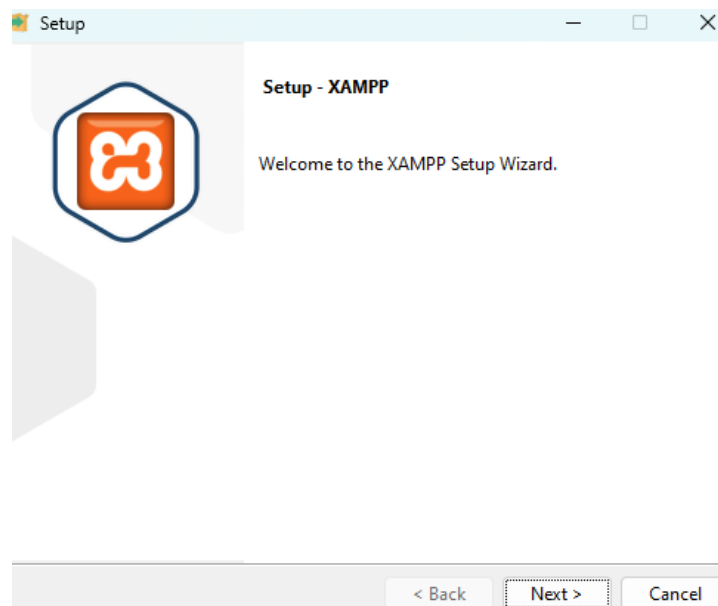
2. Klik download untuk pada versi terbaru pada Exampp.



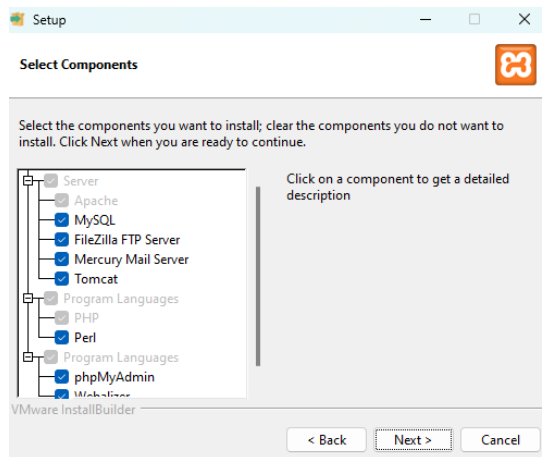
3. Tunggu sampai file selesai di download.



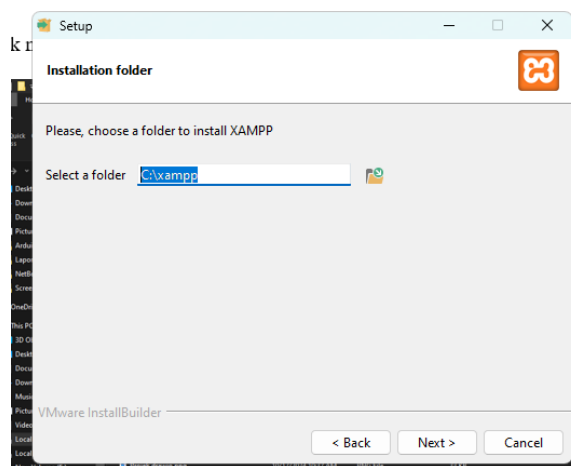
4. Buka file Xampp pada file explorer dan klik next.



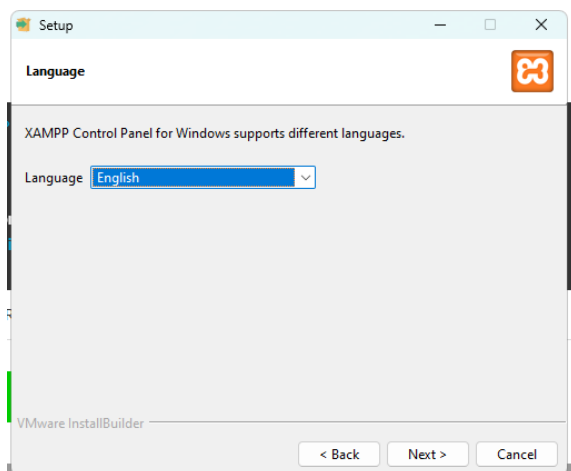
5. Klik next saja.



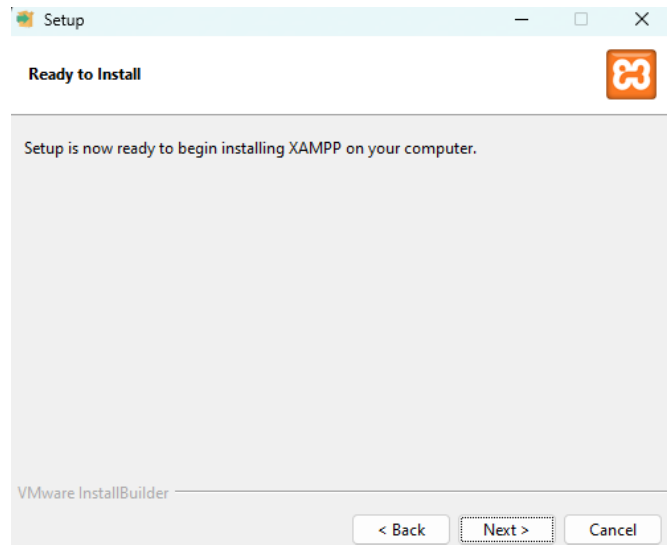
6. Klik next lagi maka file Xampp akan di simpan di direktori C:/Xampp.



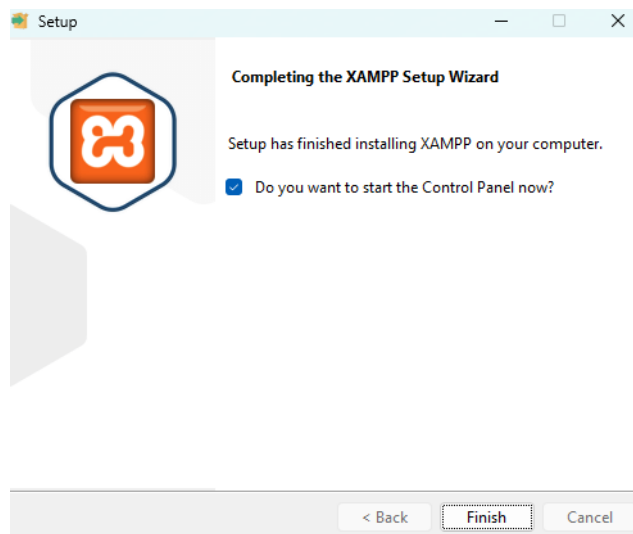
7. Klik next lagi.



8. Klik next lagi untuk install.

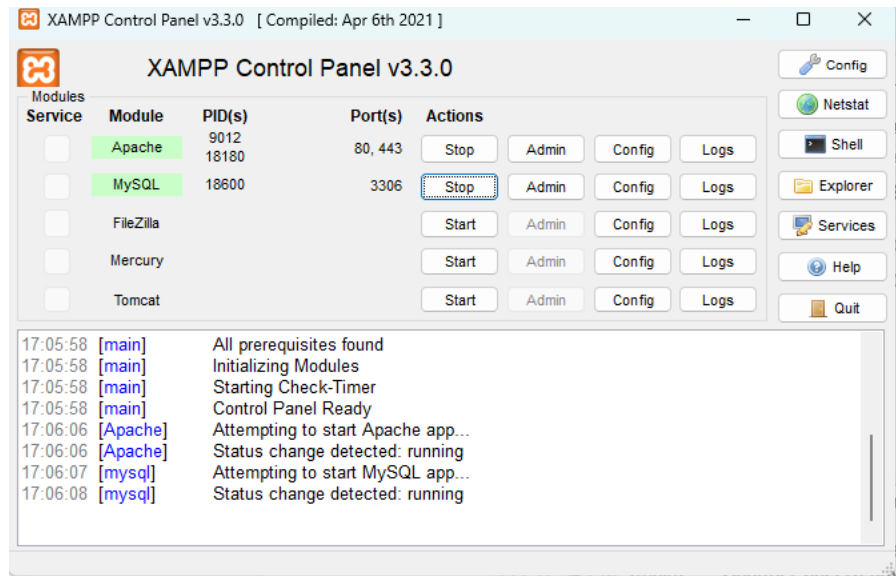


9. Klik Finish untuk menyelesaikan instalasi.

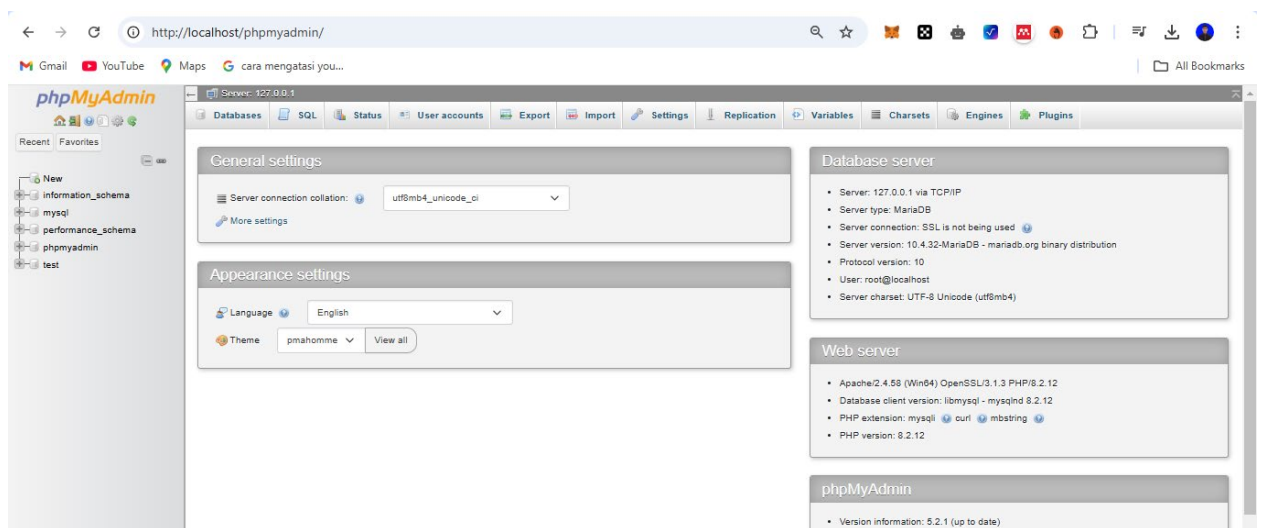


10. Nyalakan Mysql dan Apache untuk mengakses localhost dengan menekan tombol Start.



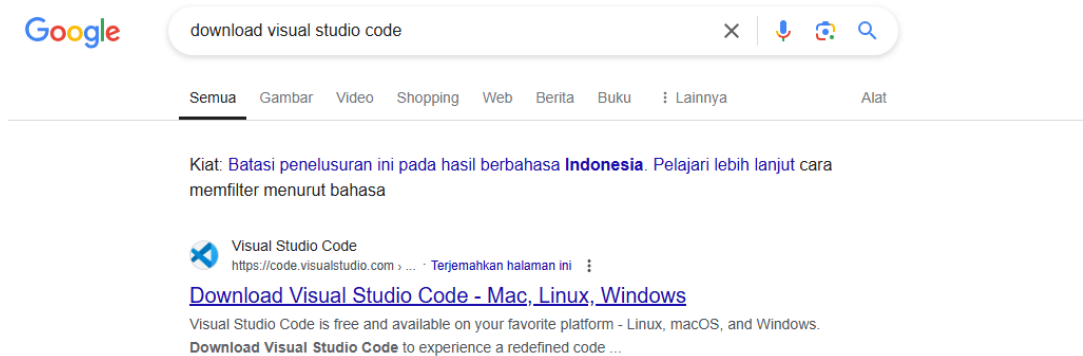


11. Klik admin pada mysql, jika muncul web sebagai berikut maka Xampp berhasil di install.

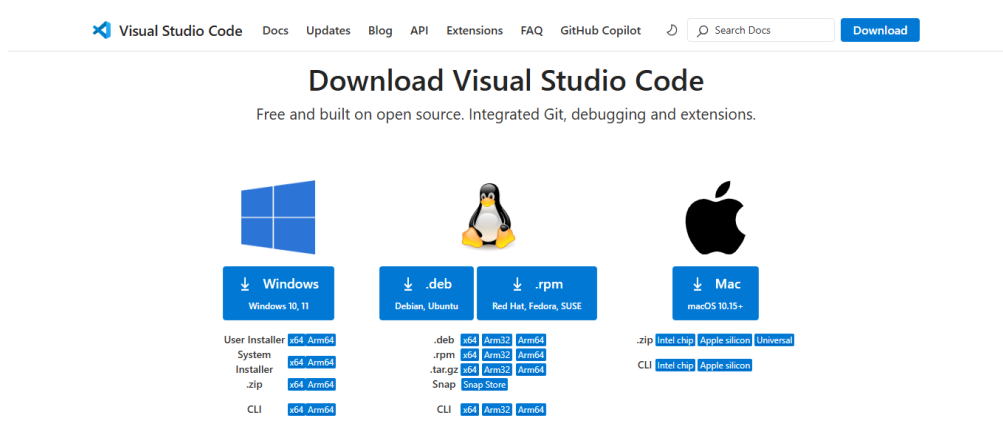


## B. Instalasi Visual Studio Code

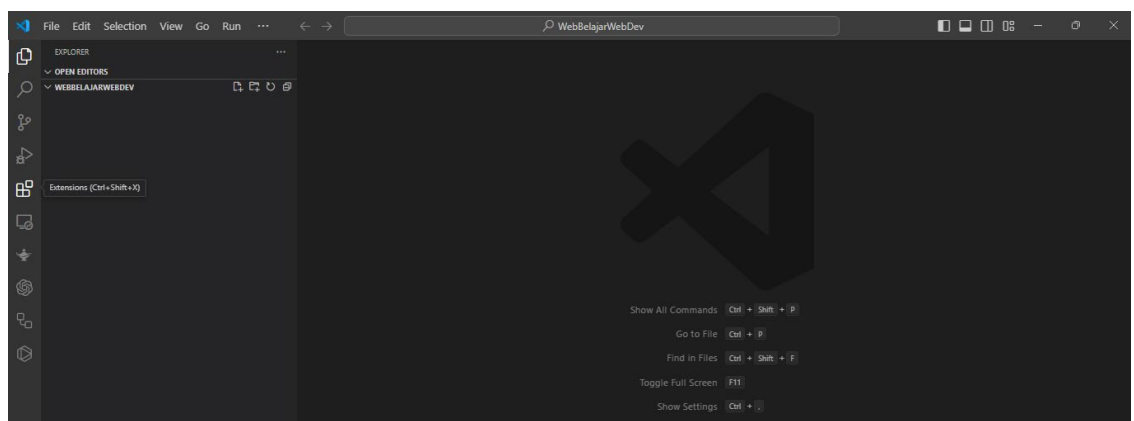
1. Buka Google Chrome dan ketik “download Visual Studio Code” dan masuk pada web resminya.



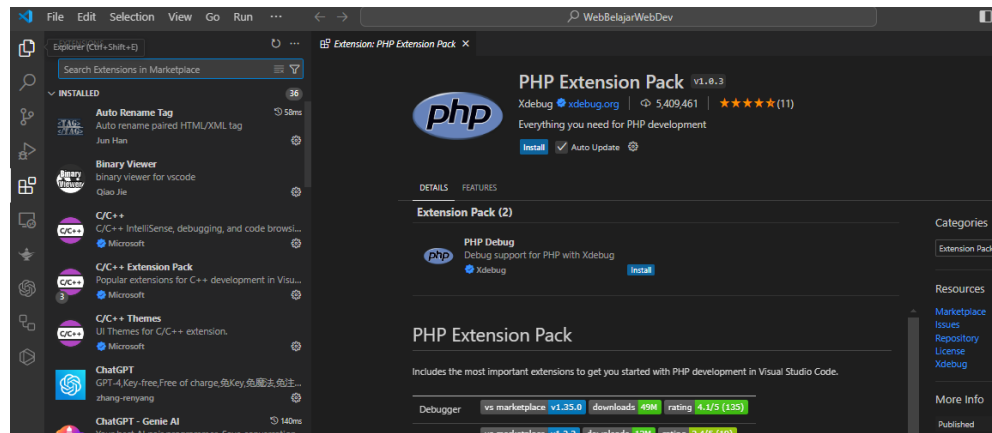
2. Klik download. Sesuaikan dengan system operasi laptop anda.



3. Jika anda sudah menginstal vs code kemudian buka aplikasi vs code dan klik pada Extensions.



4. Download extension yang anda ingin gunakan.



## PACKAGE PENDUKUNG PADA VS CODE

Berikut beberapa package atau extension populer di Visual Studio Code beserta fungsinya secara singkat:

1. Prettier

Mengatur tata letak kode secara otomatis, menjaga format kode tetap konsisten.

2. Live Server

Menyediakan server lokal untuk proyek web dengan pratinjau langsung dan auto-refresh ketika file diperbarui.

3. ESLint

Menemukan kesalahan dalam kode JavaScript dan memastikan kode sesuai aturan gaya, menjaga kualitas kode tetap tinggi.

4. Python

Penting untuk proyek Python; menyediakan dukungan sintaks, linting, debugging, dan pemformatan.

5. Java Extension Pack

Menyediakan fitur kompilasi, debugging, dan intellisense untuk pengembangan Java di VS Code.

6. GitLens

Memungkinkan melihat riwayat perubahan kode, menunjukkan siapa yang mengedit, dan mendukung kolaborasi.

## 7. Jupyter

Memungkinkan menjalankan notebook Jupyter di VS Code, sangat membantu dalam pengembangan data sains, terutama Python.

## 8. Docker

Memfasilitasi pengelolaan kontainer Docker langsung dari VS Code, ideal untuk pengembangan berbasis kontainer.

## 9. Bracket Pair Colorizer

Memberi warna berbeda pada pasangan tanda kurung, memudahkan membaca blok kode bertingkat.

## 10. Code Spell Checker

Menandai kesalahan ejaan dalam kode, membantu menemukan salah ketik di komentar atau nama variabel.

## MODUL 3

### KONFIGURASI ATAU SETTING FRAMEWORK CODEIGNITER

#### TUJUAN

1. Membuat databases dan tabel
2. Mendapatkan framework codeigniter
3. Mahasiswa mampu melakukan konfigurasi atau setting framework codeigniter

#### TUGAS PENDAHULUAN

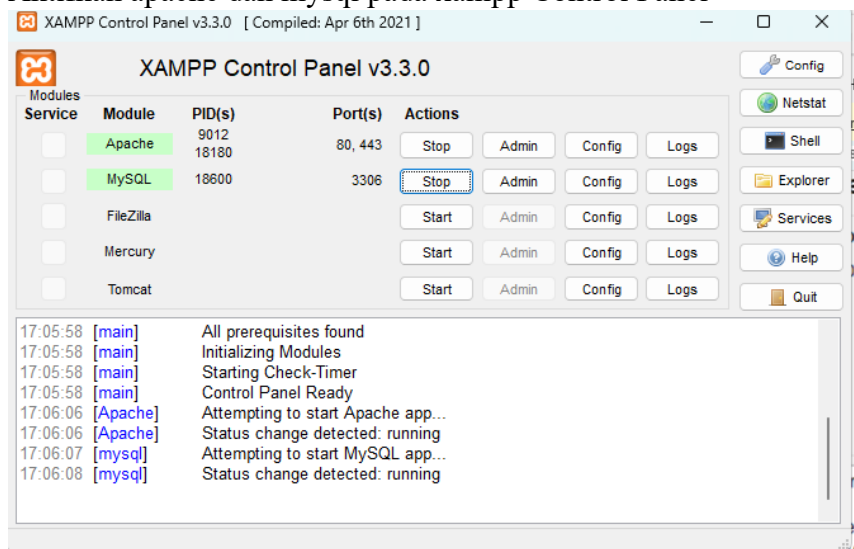
1. Mahasiswa mampu membuat databases dan tabel pada xampp
2. Bagaimana cara mendapatkan Framework CodeIgniter
3. Bagaimana cara melakukan konfigurasi atau setting Framework CodeIgniter

#### KEGIATAN PRAKTIKUM

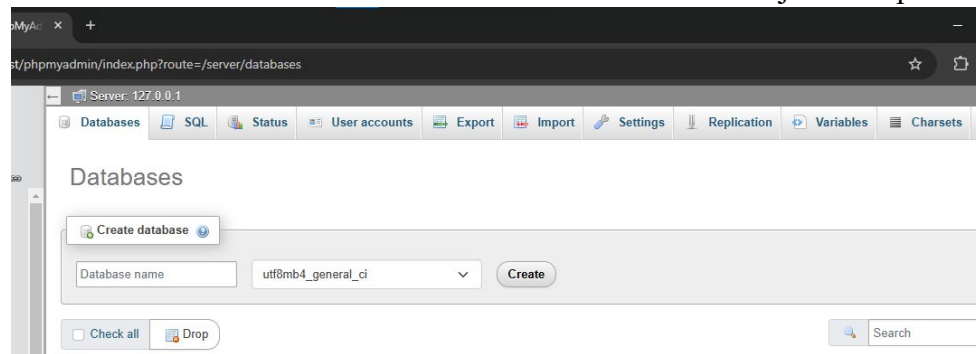
##### A. Membuat database dan Tabel

Sebelum kita merancang sebuah website atau latihan pembelajaran pada pembahasan ini, kita buat databases terlebih dulu, untuk pembuatan databases dan Tabel kali ini melalui menu console. Pada phpmyadmin dengan Langkah-langkah sebagai berikut:

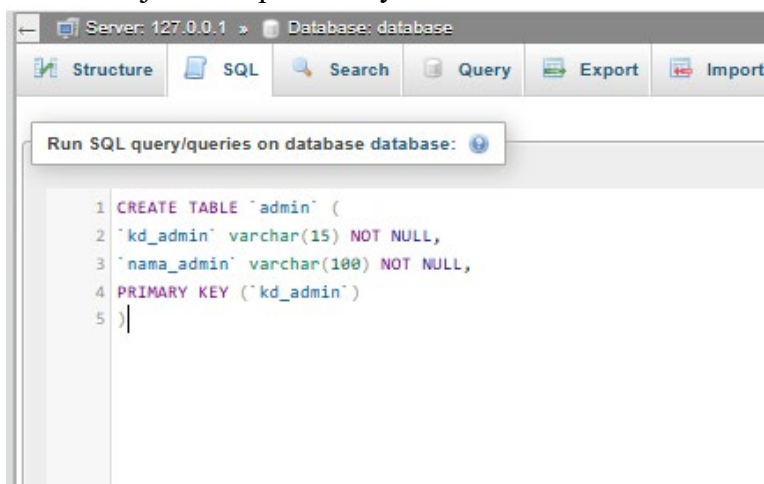
1. Aktifkan apache dan mysql pada xampp Control Panel



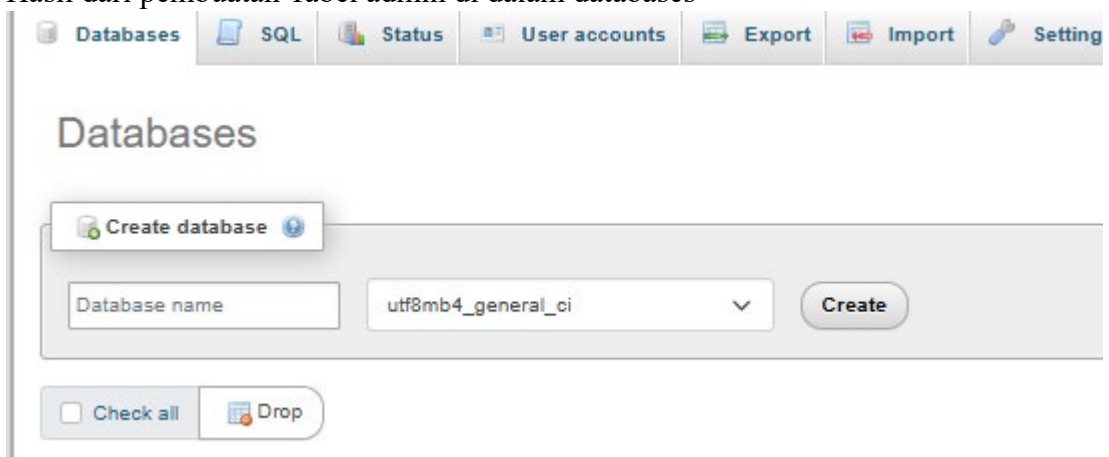
2. Buka browser ketikkan <http://localhost/phpmyadmin> pada menu console ketikkan script dibawah ini untuk membuat databases dan Ctrl+Enter untuk menjalankan perintahnya



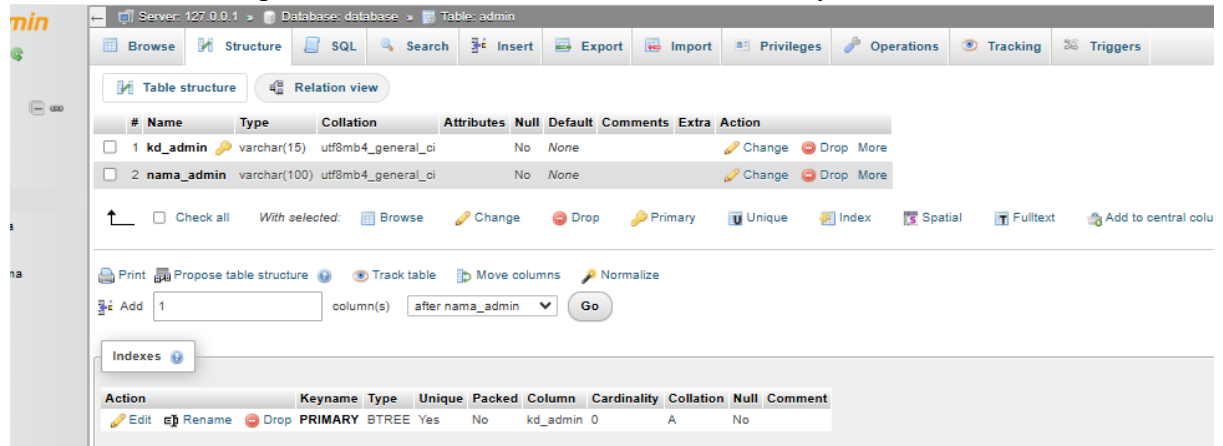
3. Sedangkan untuk tabel ketikkan script dibawah ini untuk membuat tabel dan Ctrl+Enter untuk menjalankan perintahnya.



4. Hasil dari pembuatan Tabel admin di dalam databases



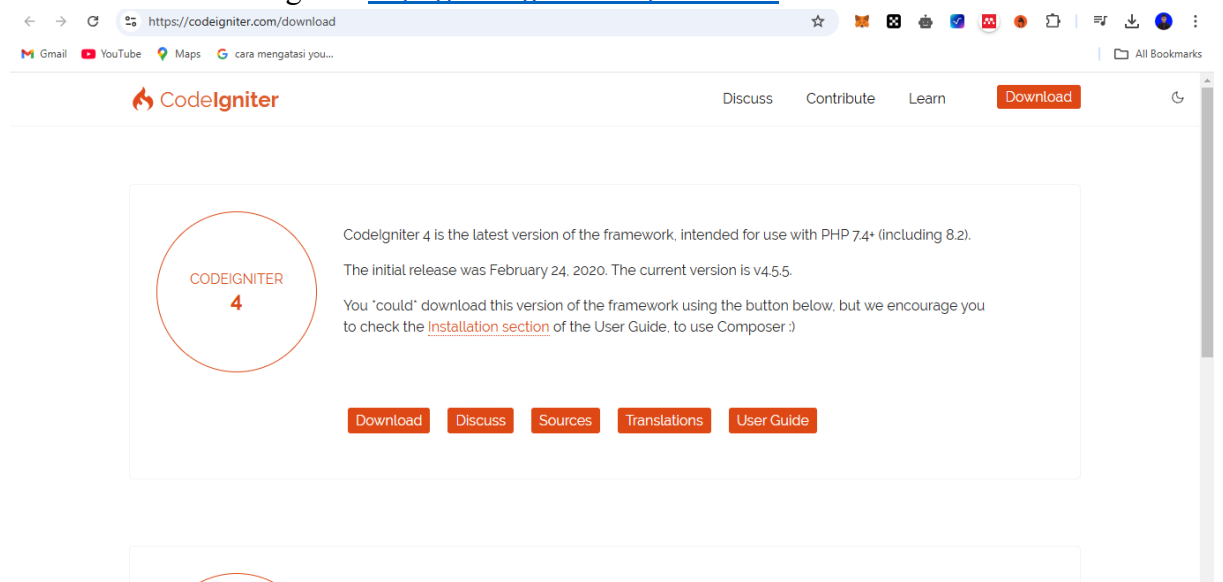
5. Lalu klik Structure pada tabel admin , untuk melihat Struktur nya.



## B. Konfigurasi atau Setting Framework CodeIgniter

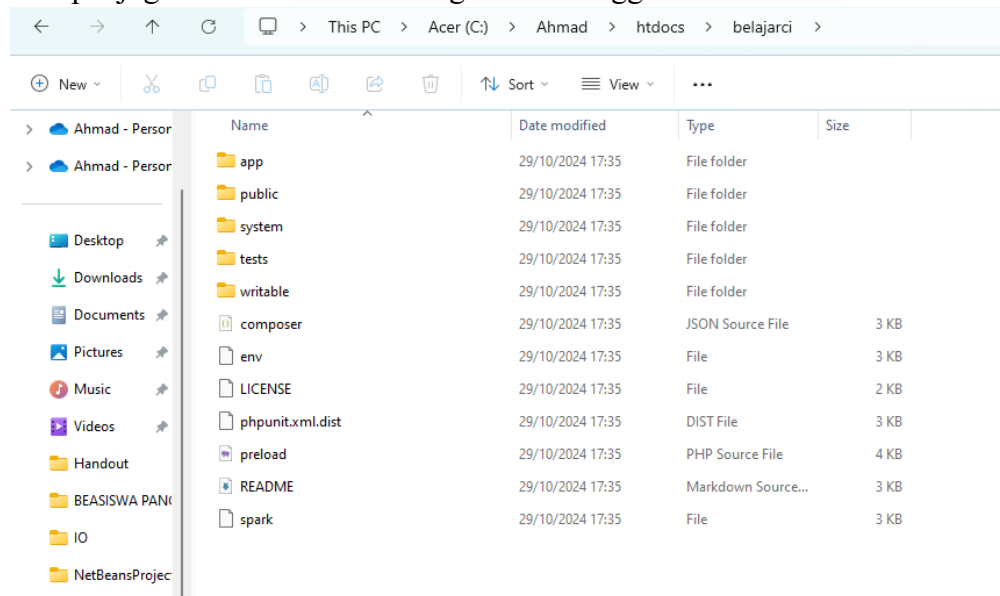
Setelah kita membuat databases dan table, berikutnya barulah kita masuk kepembahasan setting framework codeigniter, dengan Langkah-langkah sebagai berikut ini:

1. Sebelum kita lakukan setting framework codeigniter terlebih dulu kita siapkan framework codeigniter, untuk mendapatkan kita langsung saja kunjungi situs resmi dari framework codeigniter <https://codeigniter.com/download>

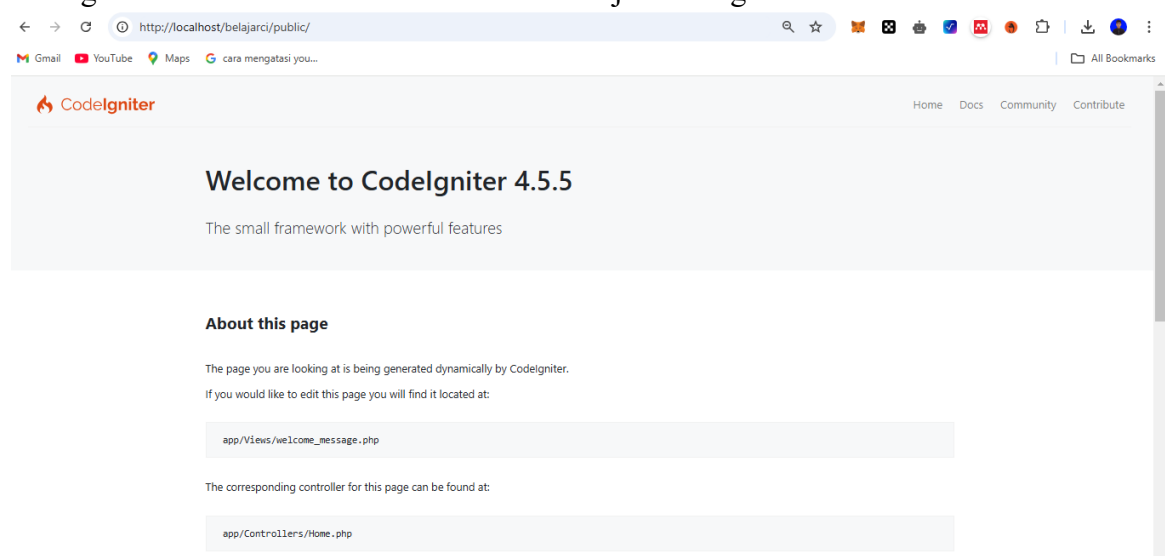


2. Download codeigniter versi Codeigniter 4 pada pembahasan materi ini, setelah kita berhasil mendownload framework codeigniter 4,kita ekstrak pada lokasi menempatkan xampp dengan lokasi C:\xampp\htdocs dan buat folder baru dengan nama belajarci, hasil ekstrak ubah namanya Dengan nama proyek portaldi, pada hal ini penulis xampp berada pada lokasi C:\xampp\htdocs.

Note: Dapat juga mendownload CodeIgniter 4 menggunakan Git Bash.

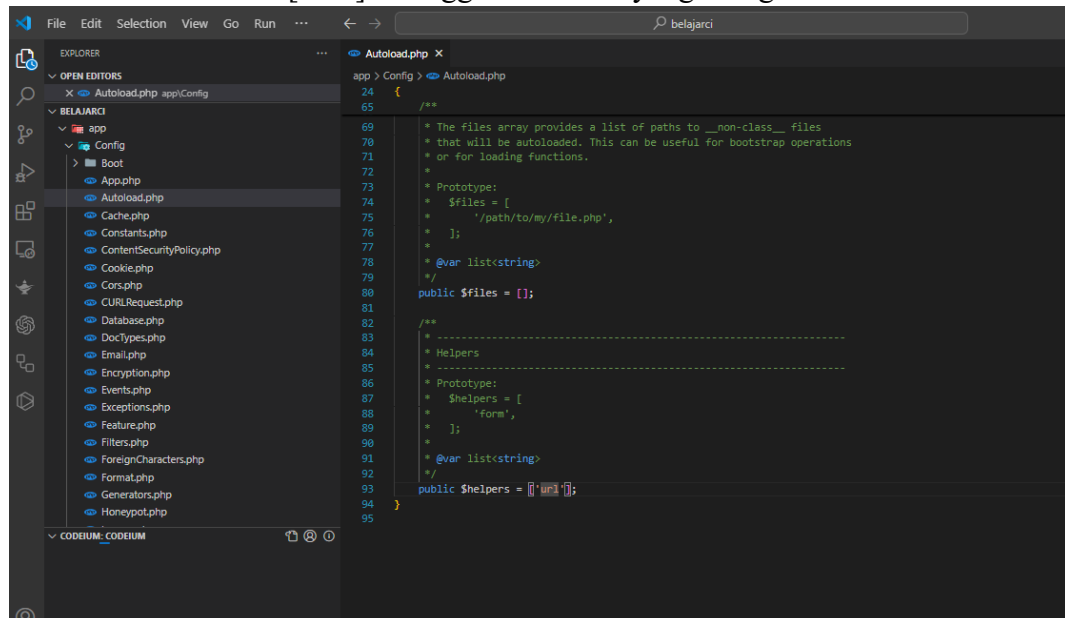


3. Untuk memastikan saja framework CodeIgniter yang kita download dan yang telah kita extract pada lokasi C:\xampp\htdocs\belajarci proyek tersebut tidak ada dokumen yang korup atau rusak maupun error sehingga siap kita gunakan untuk membangun sebuah aplikasi yang kita inginkan. Buka browse ketikkan <http://localhost/belajarci/public> maka secara default akan tampil halaman welcome to codeigniter menandakan bahwa framework berjalan dengan baik.

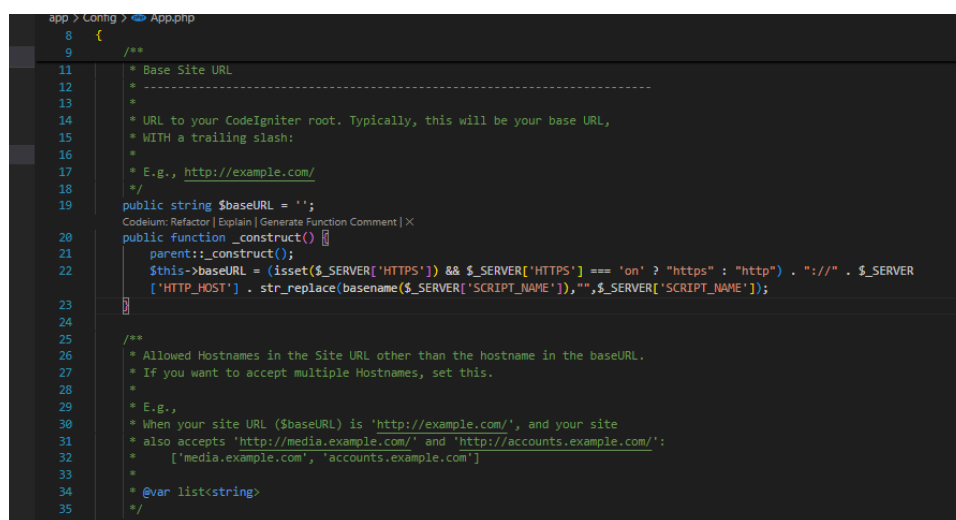




4. pada CodeIgniter 4, struktur dan cara kerja autoloading berbeda dan lebih modern dibandingkan CodeIgniter 3 atau pendahulu nya. Autoloading diatur dalam file `app/Config/Autoload.php` dan `app/Config/Services.php`. Alih-alih menggunakan array, CodeIgniter 4 mengandalkan metode *Service Locator* untuk memanggil library yang dibutuhkan.
5. Masih pada file `application\config\autoload.php` pada file `autoload.php` namun pada line 93 kita tambahkan `['url']` sehingga format url yang kita gunakan akan terbaca.



6. Di CodeIgniter 4, pengaturan baseURL bisa disesuaikan secara dinamis untuk menghindari penyesuaian manual setiap kali aplikasi dipindahkan ke server atau folder berbeda. Anda bisa mengikuti langkah-langkah berikut untuk membuat baseURL secara otomatis sesuai dengan server tempat aplikasi dijalankan.
  - Buka File `app/Config/App.php`  
Di CodeIgniter 4, baseURL dikonfigurasi di file `app/Config/App.php`.
  - Ubah public \$baseURL Menjadi Dinamis  
Pada properti \$baseURL, Anda bisa menggunakan skrip PHP untuk mendeteksi protokol, host, dan path aplikasi. Ubah bagian baseURL di `App.php` seperti ini



Dengan kode di atas, baseURL akan otomatis menyesuaikan protokol (http atau https) dan host, serta path aplikasi sesuai lokasi di server.

7. Untuk setting pada file application\config\database.php pada file Database.php karena local kita focus di line 163 pada \$tests pada username = 'root' dan di line 168 pada database= 'database'.

```
app > Config > Database.php
11  {
153  //      'datetime' => 'Y-m-d H:i:s',
154  //      'time'      => 'H:i:s',
155  //  ],
156  //  ];
157
158  /**
159   * This database connection is used when running PHPUnit database tests.
160   *
161   * @var array<string, mixed>
162   */
163  public array $tests = [
164      'DSN' => '',
165      'hostname' => '127.0.0.1',
166      'username' => 'root',
167      'password' => '',
168      'database' => 'database',
169      'DBDriver' => 'SQLite3',
170      'DBPrefix' => 'db_', // Needed to ensure we're working correctly with prefixes live. DO NOT
171      'pConnect' => false,
172      'DBDebug' => true,
173      'charset' => 'utf8',
174      'DBCollat' => '',
175      'swapPre' => '',
176      'encrypt' => false,
177      'compress' => false,
178      'strictOn' => false,
179      'failover' => [],
180      'port' => 3306,
181      'foreignKeys' => true,
182      'busyTimeout' => 1000,
183      'dateFormat' => [
184          'date' => 'Y-m-d',
185          'datetime' => 'Y-m-d H:i:s',
186          'time' => 'H:i:s',
187      ],
188  ];
189
```

8. Di CodeIgniter 4, untuk mengatur default\_controller, buka file app/Config/Routes.php dan tambahkan \$routes->get('/', 'Home::index');, yang akan memanggil controller Home pada metode index sebagai halaman awal saat aplikasi diakses tanpa URI tambahan.

```
app > Config > Routes.php
1  <?php
2
3  use CodeIgniter\Router\RouteCollection;
4
5  /**
6   * @var RouteCollection $routes
7   */
8  $routes->get('/', 'Home::index');
9  $routes->set404Override('Errors::notFound');
10 $routes->setTranslateURLDashes(false);
```

## MODUL 4

### MENGENAL CRUD PADA FRAMEWORK CODEIGNITER 4

#### TUJUAN

1. Mengenal CRUD dasar framework codeigniter
2. Penerapan MVC dengan Teknik OOP

#### TUGAS PENDAHULUAN

1. Bagaimana Mahasiswa mampu menguasai CRUD dengan baik
2. Mahasiswa mampu menerapkan MVC dengan Teknik OOP

#### KEGIATAN PRAKTIKUM

1. Pada pembuatan script yang pertama kali kita lakukan adalah membuat controller terlebih dulu, cara membuat control adalah ke Terminal dan ketikkan `php spark make:controller Admin` lalu Enter, maka Controller Admin.php akan terbuat pada app/Controllers

```
PS C:\Ahmad\htdocs\belajarci> php spark make:controller Admin

CodeIgniter v4.5.5 Command Line Tool - Server Time: 2024-10-29 11:17:52 UTC+00:00

File created: APPPATH\Controllers\Admin.php
```

```
app > Controllers > Admin.php
1  <?php
2
3  namespace App\Controllers;
4
5  use App\Controllers\BaseController;
6  use CodeIgniter\HTTP\ResponseInterface;
7
8  class Admin extends BaseController
9  {
10     public function index()
11     {
12         //
13     }
14 }
15
```

2. Disini juga kita menambahkan beberapa kolom pada admin yaitu email\_admin (varchar 100), hp\_admin (varchar 13) dan pswd\_admin (varchar 255)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 kd_admin	varchar(15)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	2 nama_admin	varchar(100)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	3 email_admin	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	4 hp_admin	varchar(13)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	5 password_admin	varchar(255)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

3. Kemudian kita lengkapi script controller Admin.php.

```
<?php

namespace App\Controllers;

use App\Controllers\BaseController;
use App\Models\MAdmin;
use CodeIgniter\HTTP\ResponseInterface;

class Admin extends BaseController
{
    private $MAdmin;
    // Property to store the view folder path
    private $view = "Views/";

    // Property to store the redirect path
    private $redirect = "Admin";

    public function __construct()
    {
        // Load the M_Admin model
        $this->MAdmin = new MAdmin();
    }

    public function index()
    {
        // Call GetAll method from MAdmin model
        $read = $this->MAdmin->getAll();
        $data = [
            // Pass 'read' variable to the view
            'read' => $read
        ];

        // Load the read.php view file from $view path
        return view($this->view . 'read', $data);
    }

    public function create()
    {
        // Pass an empty array to the create view
        $data = [
            'create' => ''
        ];

        // Load the create.php view file from $view path
        return view($this->view . 'create', $data);
    }

    public function save()
    {
        $data = [
            'kd_admin' => $this->request->getPost('kd_admin'),
            'nama_admin' => $this->request->getPost('nama_admin'),
            'email_admin' => $this->request->getPost('email_admin'),
```

```

        'hp_admin' => $this->request->getPost('hp_admin'),
        'pswd_admin' => password_hash($this->request-
>getPost('pswd_admin'), PASSWORD_DEFAULT),
    ];
    $this->MAdmin->insert($data);

    echo '<script>
        alert("Sukses Tambah Data Admin");
        window.location="'.base_url('admin').'"
    </script>';
}

public function edit($kd)
{
    // Retrieve the data to be edited based on primary key (kd)
    $data = [
        'edit' => $this->MAdmin->editData($kd)
    ];

    // Load the edit.php view file from $view path
    return view($this->view . 'edit', $data);
}

public function update($kd)
{
    $data = [
        'nama_admin' => $this->request->getPost('nama_admin'),
        'email_admin' => $this->request->getPost('email_admin'),
        'hp_admin' => $this->request->getPost('hp_admin'),
    ];

    // Update the data using the MAdmin model
    $this->MAdmin->update($kd, $data);

    // Redirect to the Admin index page
    echo '<script>
        alert("Sukses Update Data Admin");
        window.location="'.base_url('admin').'"
    </script>';
}

public function delete($kd)
{
    // Delete data based on primary key (kd)
    if ($this->MAdmin->delete($kd)) {
        // Redirect to the index page with a success message
        return redirect()->to(base_url('admin'))->with('success', 'Data berhasil dihapus');
    } else {
        // Redirect back with an error message if deletion fails
        return redirect()->back()->with('error', 'Gagal menghapus data');
    }
}
}
}

```

- Setelah kita membuat controller, selanjutnya kita akan membuat model dimana pada control Admin membutuhkan model dengan nama MAdmin. Untuk membuat model buka Terminal lalu Ketikkan `php spark make:model Madmin`

```
PS C:\Ahmad\htdocs\belajarci> php spark make:model Madmin

CodeIgniter v4.5.5 Command Line Tool - Server Time: 2024-10-29 11:24:22 UTC+00:00

File created: APPPATH\Models\Madmin.php
```

5. Kemudian kita lengkapi script model Madmin.php

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class MAdmin extends Model
{
    // Define the table name and primary key
    protected $table = 'admin';
    protected $primaryKey = 'kd_admin';
    protected $allowedFields = ['kd_admin', 'nama_admin', 'email_admin', 'hp_admin', 'pswd_admin'];

    public function getAll()
    {
        return $this->orderBy($this->primaryKey, 'DESC')->findAll();
    }

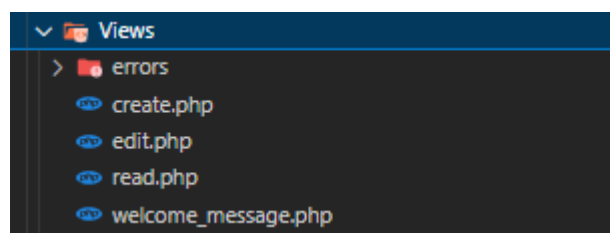
    public function saveData($data)
    {
        return $this->insert($data);
    }

    public function editData($kd)
    {
        return $this->where($this->primaryKey, $kd)->first();
    }

    public function updateData($kd, $data)
    {
        return $this->update($kd, $data);
    }

    public function deleteData($kd)
    {
        return $this->delete($kd);
    }
}
```

6. Berikutnya masuk ke folder View dan buat file read.php, edit.php, dan create.php



7. Pada view pertama yang kita buat tentu saja read.php karena sebagai view index dari control admin dengan kita telah membuat read.php kita bisa memanggil halaman lainnya seperti view dari create.php, edit.php dan aksi dari delete. Berikut ini adalah script dari view read.php yang berfungsi untuk menampilkan seluruh record pada tabel admin.

```
<!--untuk memanggil halaman tambah atau file create.php-->
<a href="admin/create"><button type="button"
name="button" title="untuk menambah data">Tambah</button></a>

<table border="1" width="70%">
  <tr>
    <td>No</td>
    <td>Kode</td>
    <td>Nama</td>
    <td>Email</td>
    <td>HP</td>
    <td>Aksi</td>

  </tr>
  <?php
    $no=1;
    // $read yang diambil dari control function index
    foreach ($read as $row) {
  ?>
  <tr>
    <td><?php echo $no ?></td>
    <td><?php echo $row['kd_admin'] ?></td>
    <td><?php echo $row['nama_admin'] ?></td>
    <td><?php echo $row['email_admin'] ?></td>
    <td><?php echo $row['hp_admin'] ?></td>
    <td>
      <!--memanggil halaman edit atau edit.php-->
      <a href="<? = base_url('admin/edit/' . $row['kd_admin']) ?>" title="tombol utk
merubah data">Ubah</a> |
      <!--memanggil aksi delete-->
      <a href="<? = base_url('admin/delete/' . $row['kd_admin']) ?>" onclick="return
confirm('Yakin ingin menghapus <? = esc($row['nama_admin']); ?>?')">Hapus</a>
    </td>
  </tr>
  <?php
    $no++;
  }
  ?>
</table>
```

8. Sampai pada view read.php ini kita dapat melihat hasil dari control admin, yakni kita kembali pada setting framework codeigniter untuk memanggil control admin app\config\routes.php, lalu masukkan Script ini

```
<?php

use CodeIgniter\Router\RouteCollection;

/**
 * @var RouteCollection $routes
 */
$routes->get('/', 'Admin::index');
$routes->get('admin', 'Admin::index'); // Route for Admin index page
$routes->get('admin/create', 'Admin::create');
$routes->get('admin/edit/(:any)', 'Admin::edit/$1');
$routes->get('admin/delete/(:any)', 'Admin::delete/$1');
$routes->post('admin/update/(:any)', 'Admin::update/$1');
$routes->post('admin/save', 'Admin::save');
$routes->set404Override('Errors::notFound');
$routes->setTranslateURIDashes(false);
```

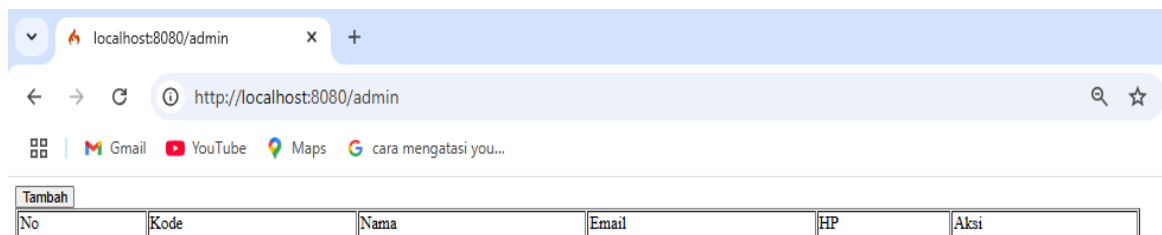
9. Lalu buka terminal dan ketikkan **php spark serve** untuk menjalankan localhost

```
PS C:\Ahmad\htdocs\belajarci> php spark serve

CodeIgniter v4.5.5 Command Line Tool - Server Time: 2024-10-29 12:33:49 UTC+00:00

CodeIgniter development server started on http://localhost:8080
Press Control-C to stop.
```

10. Setelah itu buka aplikasi web yang ada pada PC/Laptop kalian, dan Ketikkan localhost:8080 pada Pencarian





11. Disini kita akan menambahkan data melalui file create.php dimana view dari control admin pada function create yang berfungsi untuk menambah record pada tabel admin.

```
<form action="php= base_url('admin/save') ?" method="post">
  <label>Kode Admin</label><br>
  <input type="text" name="kd_admin" value=""><p></p>

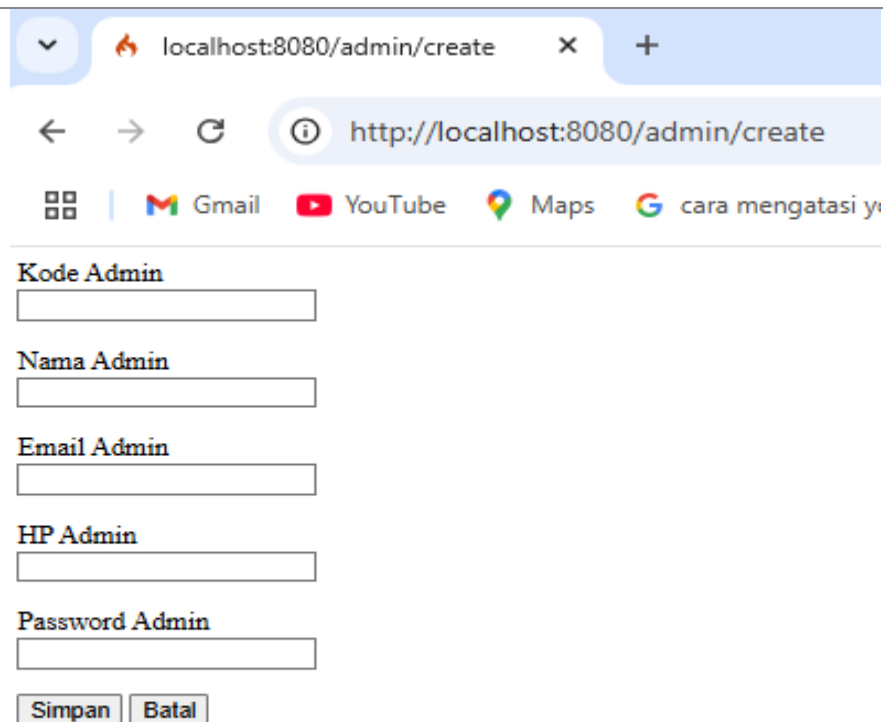
  <label>Nama Admin</label><br>
  <input type="text" name="nama_admin" value=""><p></p>

  <label>Email Admin</label><br>
  <input type="email" name="email_admin" value=""><p></p>

  <label>HP Admin</label><br>
  <input type="number" name="hp_admin" value=""><p></p>

  <label>Password Admin</label><br>
  <input type="password" name="pswd_admin" value=""><p></p>

  <button type="submit" name="button">Simpan</button>
  <a href="/admin"><button type="button" name="button">Batal</button></a>
</form>
```



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/admin/create'. The page content includes a form with the following fields and buttons:

- Kode Admin:
- Nama Admin:
- Email Admin:
- HP Admin:
- Password Admin:
- Buttons: Simpan, Batal

12. View terakhir yaitu view edit.php yakni view data control Admin dengan function edit yang berfungsi untuk mengubah data pada tabel admin. Jika kita perhatikan pada text dengan nama kd\_admin terdapat 2(dua) text, yang membedakan adalah disable dan hidden. Disable yang artinya text tidak dapat di isi atau diubah datanya, tp dengan disable tentu saja text ini tidak

dapat dikirim melalui form. Sebagai pengganti data yang bisa dikirim adalah yang text hidden sehingga text dapat dikirim melalui form.

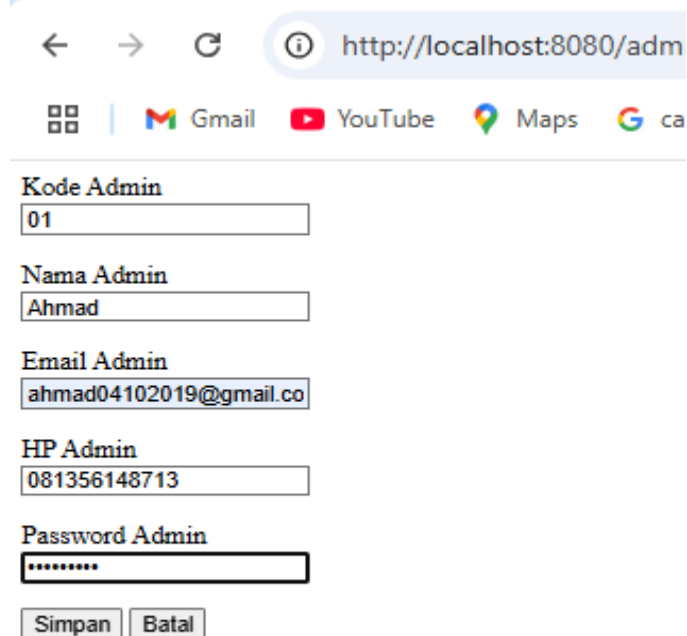
```
<form class="" action="<? = base_url('admin/update/'. $edit['kd_admin']) ?>"
method="post">
  <label>Kode Admin</label><br>
  <input type="text" value="<? = esc($edit['kd_admin']) ?>" disabled><p></p>
  <input type="hidden" name="kd_admin" value="<? = esc($edit['kd_admin']) ?>">

  <label>Nama Admin</label><br>
  <input type="text" name="nama_admin" value="<? = esc($edit['nama_admin'])
?>"><p></p>

  <label>Email Admin</label><br>
  <input type="email" name="email_admin" value="<? = esc($edit['email_admin'])
?>"><p></p>

  <label>HP Admin</label><br>
  <input type="number" name="hp_admin" value="<? = esc($edit['hp_admin']) ?>"><p></p>

  <button type="submit" name="button">Perbaharui</button>
  <a href="/admin"><button type="button" name="button">Batal</button></a>
</form>
```



← → ↻ ⓘ http://localhost:8080/adm

⌵ | Gmail YouTube Maps G ca

**Kode Admin**

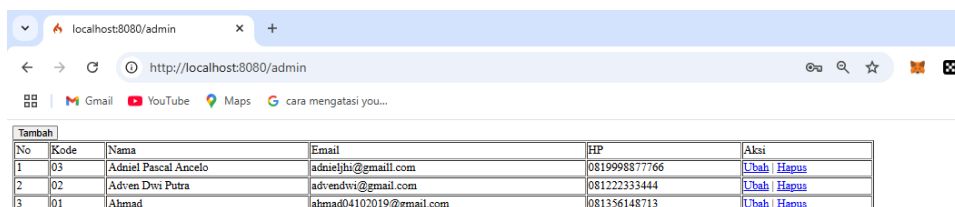
**Nama Admin**

**Email Admin**

**HP Admin**

**Password Admin**

Hasil:



⌵ | Gmail YouTube Maps G cara mengatasi you...

No	Kode	Nama	Email	HP	Aksi
1	03	Adniel Pascal Ancelo	adniejh@gmail.com	0819998877766	<a href="#">Ubah</a> <a href="#">Hapus</a>
2	02	Adven Dwi Putra	advendwi@gmail.com	081222333444	<a href="#">Ubah</a> <a href="#">Hapus</a>
3	01	Ahmad	ahmad04102019@gmail.com	081356148713	<a href="#">Ubah</a> <a href="#">Hapus</a>

## MODUL 5

### MEMBUAT SESSION PADA FRAMEWORK CODEIGNITER

#### TUJUAN

1. Mengenal CRUD dasar framework codeigniter
2. Penerapan MVC dengan Teknik OOP

#### TUGAS PENDAHULUAN

1. Bagaimana Mahasiswa mampu menguasai CRUD dengan baik
2. Mahasiswa mampu menerapkan MVC dengan Teknik OOP

#### LANGKAH PRAKTIKUM

Session adalah sebuah variable yang disimpan pada server bersifat sementara pada akhirnya menjadi data sementara dengan tujuan dari segi keamanan sebuah aplikasi dan data tersebut akan hancur dengan mengakhiri data tersebut. Dengan kata lainnya session adalah validasi login atau hak akses untuk sebuah keamanan yakni ketika kita berhasil login pada aplikasi dan ketika kita logout dari aplikasi tersebut. Pada framework codeigniter untuk session telah disediakan yakni pada library session, sehingga memudahkan kita tinggal memanggilnya saja dengan demikian memudahkan kita dalam penulisan script session. Berikut ini kita terapkan session pada projek dengan Langkah-langkah sebagai berikut:

1. Tambahkan file **Template.php** pada lokasi **application\libraries** dengan dengan *script* sebagai berikut:

```

Template.php ...Libraries X Template.php ...Views Autoload.php islogin_helper.php Au
app > Libraries > Template.php
1  <?php
2
3  namespace App\Libraries;
4
5  class Template
6  {
7      protected $template_data = [];
8
9      // Set a value for use in the template
10
11      public function set($name, $value)
12      {
13          $this->template_data[$name] = $value;
14      }
15
16      // Load a template along with a view
17
18      public function load($template = 'default_template', $view = '', $view_data = [])
19      {
20          // Merge view data with template data
21          $data = array_merge($this->template_data, $view_data);
22
23          // Load the template and embed the view within it
24          echo view($template, ['contents' => view($view, $data)]);
25      }
26  }

```

2. Dengan views **class Template** dengan nama file **template.php** pada lokasi **application\views\backend\template.php**

```

1  <html lang="en" dir="ltr">
2      <head>
3          <meta charset="utf-8">
4          <title>portalci</title>
5      </head>
6      <body>
7          <a href="<?php echo site_url('Home') ?>">Beranda</a> |
8          <a href="<?php echo site_url('Admin') ?>">Data Admin</a> |
9          <a href="<?php echo site_url('Kategori') ?>">Data Kategori</a> |
10         <a href="<?php echo site_url('Berita') ?>">Data Berita</a> |
11         <a href="<?php echo site_url('Auth/logout') ?>">Keluar</a>
12         <hr>
13         <?php echo $judul; echo ' <small>'; echo ' >> ' ; echo $sub; echo ' </small>'; ?>
14         <p></p>
15         <!--isi content-->
16         <?php echo $contents ?>
17         <!--end isi content-->
18     </body>
19 </html>

```

3. Masih pada **autoload.php** yakni pada **application\config\autoload.php** line 92 dengan menambahkan **islogin\_helper.php** berfungsi untuk memanggil function dengan pemberian nilai atau algoritma lainnya pada **islogin\_helper.php**

```
app > Config > Autoload.php
24  {
82      /**
83       * -----
84       * Helpers
85       * -----
86       * Prototype:
87       *   $helpers = [
88       *       'form',
89       *   ];
90       *
91       * @var list<string>
92       */
93       public $helpers = ['url', 'islogin_helper'];
94  }
```

4. Kita tambahkan **islogin\_helper.php** pada lokasi folder **application\helpers** dengan dengan *script* sebagai berikut:

```
app > Config > Autoload.php
24  {
82      /**
83       * -----
84       * Helpers
85       * -----
86       * Prototype:
87       *   $helpers = [
88       *       'form',
89       *   ];
90       *
91       * @var list<string>
92       */
93       public $helpers = ['url', 'islogin_helper'];
94  }
```

5. Berikutnya kita buat control **Auth** dengan tujuan memvalidasi data ketika akan memasuki halaman utama administrator secara default akan menampilkan view login, control **Auth** lokasi file pada **application\controllers\Auth.php**

```

app > Controllers > Auth.php
1  <?php
2
3  namespace App\Controllers;
4
5  use App\Models\MAuth;
6  use App\Controllers\BaseController;
7
8  Codeium: Refactor | Explain
9  class Auth extends BaseController
10 {
11     private $view = "Views/";
12     private $redirect = 'admin';
13     protected $MAuth;
14
15     Codeium: Refactor | Explain | Generate Function Comment | X
16     public function __construct()
17     {
18         // Load model
19         $this->MAuth = new MAuth();
20     }
21
22     // Destroy the session and load the login view
23     Codeium: Refactor | Explain | X
24     public function index()
25     {
26         $data = [
27             'login' => ''
28         ];
29         return view('login', $data);
30     }
31
32     // Login function
33     Codeium: Refactor | Explain | X
34     public function login()
35     {
36         $kd = $this->request->getPost('kd_admin');
37         $pswd = $this->request->getPost('password_admin'); // Consider using password_hash() instead of md5 for better security
38
39         // Check login using the MAuth model
40         $data = $this->MAuth->where('kd_admin', $kd)->first();

```

```

36         $data = $this->MAuth->where('kd_admin', $kd)->first();
37         $pass = $data['password_admin'];
38         $verify_pass = password_verify($pswd, $pass);
39
40         // If login is valid
41         if ($verify_pass) {
42             $array = [
43                 'kd_admin' => $data['kd_admin'],
44                 'nama_admin' => $data['nama_admin'],
45                 'IsAdmin' => 1
46             ];
47             session()->set($array);
48             return redirect()->to(base_url('home'))->with('success', 'Login successful');
49         } else {
50             echo "<script>alert('Username atau Password salah!');</script>";
51             return redirect()->to(site_url('WWW'))->with('error', 'Invalid username or password');
52         }
53     }
54
55     // Logout function
56     Codeium: Refactor | Explain | X
57     public function logout()
58     {
59         // Destroy session data
60         session()->destroy();
61         return redirect()->to(site_url('Auth'))->with('info', 'Logged out successfully');
62     }

```

6. Membuat model dari control **Auth** dengan nama **M\_auth.php** pada lokasi file

**application\models\M\_auth.php**

```
1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  class MAuth extends Model
8  {
9      protected $table = 'admin'; // Set the database table
10     protected $primaryKey = 'kd_admin'; // Set the primary key of the table
11     protected $allowedFields = ['kd_admin', 'passwordd_admin', 'nama_admin']; // Define allowed fields for mass assignment
12
13     // Method to check login details
14     public function CekLogin($table, $pk, $kd)
15     {
16         return $this->db->table($table)
17             ->where($pk, $kd)
18             ->get()
19             ->getRowArray();
20     }
21 }
```

7. Dengan view control **Auth** dengan nama file **login.php** pada lokasi file

**application\views\backend\login.php**

```
1  <form method="post" action="/login">
2      <label for="kd_admin">Kode Admin</label><br>
3      <input type="text" id="kd_admin" name="kd_admin" value=""><p></p>
4
5      <label for="pswd_admin">Password</label><br>
6      <input type="password" id="password_admin" name="pswd_admin" value=""><p></p>
7
8      <input type="submit" value="Login">
9  </form>
```

8. Kali ini kita bisa melihat hasil dari control **Auth**, jangan lupa pada setting framework codeigniter pada bagian **default\_controller** ganti pada control **Auth**

```
app > Config > Routes.php
1  <?php
2
3  use App\Controllers\Auth;
4  use CodeIgniter\Router\RouteCollection;
5
6  /**
7   * @var RouteCollection $routes
8   */
9  $routes->get('admin', 'Admin::index');
10 $routes->get('admin/create', 'Admin::create');
11 $routes->get('admin/edit/(:any)', 'Admin::edit/$1');
12 $routes->get('admin/delete/(:any)', 'Admin::delete/$1');
13 $routes->post('admin/update/(:any)', 'Admin::update/$1');
14 $routes->post('admin/save', 'Admin::save');
15 $routes->get('/', 'Auth::index');
16 $routes->post('/login', 'Auth::login');
17 $routes->get('home', 'Home::index');
18 $routes->set404Override('Errors::notFound');
19 $routes->setTranslateURIDashes(false);
20
```

9. Buka browse ketikkan **http://localhost/belajarci/portalci** maka secara default akan memanggil control Auth dengan view login.php sehingga untuk masuk pada halaman utama atau halaman berikutnya ditahan untuk melakukan validasi **login** terlebih dulu

Kode Admin

Password



## 10. memanggil home

```
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3 <head>
4   <meta charset="utf-8">
5   <title>portalci</title>
6 </head>
7 <body>
8   <a href="?= site_url('Home') ?>">Beranda</a> |
9   <a href="?= site_url('admin') ?>">Data Admin</a> |
10  <a href="?= site_url('Kategori') ?>">Data Kategori</a> |
11  <a href="?= site_url('Berita') ?>">Data Berita</a> |
12  <a href="?= site_url('Auth/logout') ?>">Keluar</a>
13  <hr>
14  <h2>Beranda</h2>
15  <p></p>
16
17  <!-- isi content -->
18  <!-- end isi content -->
19
20 </body>
21 </html>
```

[Beranda](#) | [Data Admin](#) | [Data Kategori](#) | [Data Berita](#) | [Keluar](#)

---

## Beranda

## 11. Buat Database Dengan Struktur seperti ini

Table structure		Relation view							
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id_kategori	int(4)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	2 nama_kategori	varchar(30)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

## 12. Tambahkan file Kategori.php di App>Controllers dengan kode seperti di bawah

```
<?php

namespace App\Controllers;

use App\Models\MKategori;

class Kategori extends BaseController
{
    private $view = "Views/backend/v_kategori/";

    private $redirect = "kategori";
```

```

protected $MKategori;

public function __construct()
{
    $this->MKategori = new MKategori();
    helper(['form', 'url']);
    IsAdmin(); // pastikan fungsi ini disesuaikan dengan CI 4
}

public function index()
{
    $search = $this->request->getGet('search');
    if ($search) {
        $q = $this->MKategori->search($search); // Asumsikan ada metode search di model
    } else {
        $q = $this->MKategori->getAll(); // Mengambil semua kategori
    }

    $data = [
        'judul' => "DATA KATEGORI",
        'sub' => "Lihat Kategori",
        'read' => $q
    ];

    return view($this->view . 'read', $data);
}

public function create()
{
    $data = [
        'judul' => "DATA KATEGORI",
        'sub' => "Tambah Kategori",
        'create' => "
    ];

    return view($this->view . 'create', $data);
}

public function save()
{
    $data = [
        'nama_kategori' => $this->request->getPost('nama_kategori')]

```

```

$this->MKategori->update($id, $data); // Memperbarui kategori berdasarkan ID

echo '<script>

    alert("Sukses Update Data Kagtegori");

    window.location="'.base_url('kategori')."'

</script>';
}

public function delete($id)
{
    if ($this->MKategori->where('id_kategori', $id)->delete()) {
        return redirect()->to(base_url('kategori'))->with('success', 'Data berhasil dihapus');
    } else {
        return redirect()->back()->with('error', 'Data gagal dihapus');
    }
}
}
}

```

13. Setelah itu tambahkan file MKategori.php di folder Models dengan kode berikut

```

<?php

namespace App\Models;

use CodeIgniter\Model; // Memastikan Anda menggunakan Model dari CodeIgniter

class MKategori extends Model {

    protected $table = 'kategori'; // Ganti dengan nama tabel yang sesuai

    protected $primaryKey = 'id_kategori'; // Ganti dengan nama kolom kunci utama jika diperlukan

    protected $allowedFields = ['id_kategori', 'nama_kategori']; // Pastikan ini berisi kolom yang dapat diisi

    // Menampilkan semua kategori

    public function getAll() {

        return $this->orderBy($this->primaryKey, 'DESC')->findAll();

    }
}

```

```

// Mengambil data kategori untuk diedit

public function editData($id) {

    return $this->where($this->primaryKey, $id)->first(); // Mengambil data
    berdasarkan ID

}

// Menyimpan data kategori baru

public function saveData($data) {

    return $this->insert($data); // Menyimpan data kategori

}

// Memperbarui data kategori

public function updateData($id, $data) {

    return $this->update($id, $data); // Memperbarui data kategori berdasarkan
    ID

}

// Menghapus data kategori

public function deleteData($id)

{

    return $this->delete($id);

}

}

?>

```

14. Dan tambahkan kode di Routes.php untuk kategori sehingga seperti ini

```

7  * @var RouteCollection $routes
8  */
9  $routes->get('admin', 'Admin::index');
10 $routes->get('admin/create', 'Admin::create');
11 $routes->get('admin/edit/{:any}', 'Admin::edit/$1');
12 $routes->get('admin/delete/{:any}', 'Admin::delete/$1');
13 $routes->post('admin/update/{:any}', 'Admin::update/$1');
14 $routes->post('admin/save', 'Admin::save');
15 $routes->get('/', 'Auth::index');
16 $routes->post('/login', 'Auth::login');
17 $routes->get('home', 'Home::index');
18
19 $routes->get('kategori', 'Kategori::index');
20 $routes->get('kategori/create', 'Kategori::create');
21 $routes->get('kategori/edit/{:any}', 'Kategori::edit/$1');
22 $routes->get('kategori/update/{:any}', 'Kategori::update/$1');
23 $routes->get('kategori/delete/{:any}', 'Kategori::delete/$1');
24 $routes->post('kategori/save', 'kategori::save');
25
26
27

```

15. Setelah itu, salin link berikut : <http://localhost:8080/index.php/kategori>, kemudian paste di bagian url, maka akan muncul halaman dari Kategori
16. Setelah itu buat folder di dalam folder Views>backend dengan nama "v\_kategori", kemudian didalam folder tersebut, buat file dengan nama create.php, read.php, dan edit.php, kemudian masukkan kode berikut di file masing masing

Read.php

```
<a href="<?= base_url('kategori/create') ?>" title="Tambah Data">
<button type="button">Tambah</button>
</a>

<!-- Table -->
<table border="1" width="70%">
<tr>
<th>ID Kategori</th>
<th>Nama Kategori</th>
<th>Aksi</th>
</tr>
<?php
$no = 1; // Inisialisasi nomor urut
foreach ($read as $row) : ?>
<tr>
<td><?= $row['id_kategori'] ?></td>
<td><?= $row['nama_kategori'] ?></td>
<td>
<a href="<?= base_url('kategori/edit/' . $row['id_kategori']) ?>" title="Ubah">
<button>Ubah</button>
</a>
<a href="<?= base_url('kategori/delete/' . $row['id_kategori']) ?>" title="Delete"
onclick="return confirm('Yakin mau dihapus kategori <?= $row['nama_kategori'] ?>?')">
<button>Hapus</button>
```

```

        <button>Hapus</button>

    </a>

</td>

</tr>

<?php
$no++; // Increment nomor urut
endforeach; ?>
</table>

```

Edit.php

```

<!-- View untuk Update Kategori (edit.php) -->

<h1>Ubah Kategori</h1>

<form action="<?= base_url('kategori/update/' . $edit['id_kategori']) ?>" method="POST">

    <label for="nama_kategori">Nama Kategori:</label>

    <input type="text" id="nama_kategori" name="nama_kategori" value="<?=
    $edit['nama_kategori'] ?>" required>

    <button type="submit">Update</button>

</form>

```

Create.php

```

<!-- View untuk Create Kategori (create.php) -->

<h1>Tambah Kategori</h1>

<form action="<?= base_url('kategori/save') ?>" method="POST">

    <label for="nama_kategori">Nama Kategori:</label>

    <input type="text" id="nama_kategori" name="nama_kategori" placeholder="Nama
    Kategori" required>

    <button type="submit">Tambah</button>

</form>

```

Tambah	ID Kategori	Nama Kategori	Aksi
--------	-------------	---------------	------

# Tambah Kategori

Nama Kategori:

<input type="button" value="Tambah"/>		
ID Kategori	Nama Kategori	Aksi
9	Olahraga	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>

## MODUL 6

### MENGELOLA DATA BERITA

#### TUJUAN

1. Teknik memanggil model lebih dari satu
2. Mengenal function gambar

#### TUGAS PENDAHULUAN

1. Mahasiswa mampu melakukan join dengan tabel satu dengan tabel lainnya
2. Mahasiswa mampu memanfaatkan function gambar
3. Mahasiswa mampu mengelola gambar

#### KEGIATAN PRAKTIKUM

1. Buat table dengan struktur seperti di bawah

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id_berita	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 kd_admin	varchar(15)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	3 id_kategori	int(4)			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	4 st_berita	varchar(6)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	5 tgl_berita	date			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	6 jam_berita	time			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	7 judul_berita	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	8 isi_berita	text	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	9 img_berita	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More

2. Buat file pada lokasi application\controllers dengan nama file Berita.php dengan kode sebagai berikut

```
<?php

namespace App\Controllers;

use App\Models\M_berita;
use App\Models\MKategori;

class Berita extends BaseController
{
    private $view = "Views/backend/v_berita/";
    private $redirect = "berita";

protected $MKategori;

    public function __construct()
    {
```



```

$data = [
    'judul' => "DATA BERITA",
    'sub' => "Tambah Berita",
    'kategori' => $this->MKategori->findAll(),
    'create' => ""
];

return view($this->view . 'create', $data);
}

public function save()
{
    $img = $this->request->getFile('img_berita');

    if ($img && !$img->hasMoved()) {
        $allowedTypes = ['image/jpeg', 'image/jpg', 'image/png'];
        if (!in_array($img->getMimeType(), $allowedTypes)) {
            echo "<script>alert('Format yang digunakan harus jpeg|jpg|png');</script>";
            return redirect()->to($this->redirect);
        }

        $imgName = $img->getRandomName();
        $img->move('./assets/img_berita/', $imgName);

        $data = [
            'judul_berita' => $this->request->getPost('judul_berita'),
            'id_kategori' => $this->request->getPost('id_kategori'),
            'st_berita' => 'Blokir',
            'isi_berita' => $this->request->getPost('isi_berita'),
            'tgl_berita' => date('Y-m-d'),

```

```

        'kd_admin' => session()->get('kd_admin'),
        'img_berita' => $imgName
    ];

    $this->M_berita->save($data);
    return redirect()->to($this->redirect);
}
}

public function edit($kd)
{
    $data = [
        'judul' => "DATA BERITA",
        'sub' => "Ubah Berita",
        'kategori' => $this->MKategori->findAll(),
        'edit' => $this->M_berita->find($kd)
    ];

    return view($this->view . 'edit', $data);
}

public function update($kd)
{
    $img = $this->request->getFile('img_berita');
    $data = [
        'judul_berita' => $this->request->getPost('judul_berita'),
        'id_kategori' => $this->request->getPost('id_kategori'),
        'isi_berita' => $this->request->getPost('isi_berita'),
        'tgl_berita' => $this->request->getPost('tgl_berita'),
        'jam_berita' => $this->request->getPost('jam_berita'),
        'kd_admin' => session()->get('kd_admin')
    ];

```

```

if ($img && !$img->hasMoved()) {
    $allowedTypes = ['image/jpeg', 'image/jpg', 'image/png'];
    if (!in_array($img->getMimeType(), $allowedTypes)) {
        echo "<script>alert('Format yang digunakan harus jpeg|jpg|png');</script>";
    } else {
        $imgName = $img->getRandomName();
        $img->move('./assets/img_berita/', $imgName);
        $data['img_berita'] = $imgName;
    }
}

$this->M_berita->update($kd, $data);
return redirect()->to($this->redirect);
}

public function status($kd, $status)
{
    $data = [
        'st_berita' => $status
    ];

    $this->M_berita->update($kd, $data);
    return redirect()->to($this->redirect);
}

public function delete($kd)
{
    $this->M_berita->delete($kd);
    return redirect()->to($this->redirect);
}
}

```

3. Selanjutnya pada control yaitu buat file di application\models dengan nama file M\_berita.php script sebagai berikut

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class M_berita extends Model
{
    protected $table = 'berita';
    protected $primaryKey = 'id_berita';
    protected $allowedFields = [
        'judul_berita', 'id_kategori', 'st_berita', 'isi_berita',
        'tgl_berita', 'jam_berita', 'kd_admin', 'img_berita'
    ];

    public function getAll()
    {
        return $this->select('berita.*, admin.*, kategori.*')
            ->join('admin', 'berita.kd_admin = admin.kd_admin')
            ->join('kategori', 'berita.id_kategori = kategori.id_kategori')
            ->orderBy('tgl_berita', 'desc')
            ->orderBy('jam_berita', 'desc')
            ->findAll();
    }

    public function saveBerita($data)
    {
        return $this->insert($data);
    }
}
```

```
public function edit($kd)
{
    return $this->select('berita.*, kategori.*')
        ->join('kategori', 'berita.id_kategori = kategori.id_kategori')
        ->where($this->primaryKey, $kd)
        ->first();
}

public function updateBerita($kd, $data)
{
    return $this->update($kd, $data);
}

public function deleteBerita($kd)
{
    return $this->where($this->primaryKey, $kd)->delete();
}

public function status($kd, $data)
{
    return $this->update($kd, $data);
}

public function search($data)
{
    return $this->select('berita.*, kategori.*')
        ->join('kategori', 'berita.id_kategori = kategori.id_kategori')
        ->like($this->primaryKey, $data)
        ->orLike('judul_berita', $data)
        ->orLike('nama_kategori', $data)
        ->findAll();
}}
```

4. Tambahkan fungsi pada file islogin\_helper.php pada lokasi application\helpers dengan isi fungsi sebagai berikut

```
function uploadImg($file, $dst, $style, $size)
{
    $date = date('YmdHis') . '_' . $style;
    $filename = $file->getName();
    $filetype = $file->getMimeType();
    $filetmp = $file->getTempName();
    $fileupload = $dst . $filename;

    // Upload original size
    $file->move($dst, $filename);

    // Identify image type
    if ($filetype === 'image/jpeg' || $filetype === 'image/jpg') {
        $src = imagecreatefromjpeg($fileupload);
    } elseif ($filetype === 'image/png') {
        $src = imagecreatefrompng($fileupload);
    } else {
        return false; // Unsupported file type
    }

    // Original dimensions
    $wsr = imagesx($src);
    $hsrc = imagesy($src);

    // Set new dimensions
    $wdst = $size;
    $hdst = ($wdst / $wsr) * $hsrc;
```

```

// Resize image
$filecreate = imagecreatetruecolor($wdst, $hdst);
imagecopyresampled($filecreate, $src, 0, 0, 0, 0, $wdst, $hdst, $wsrc, $hsrc);

// Randomize filename
$fileParts = explode(".", $filename);
$name = $fileParts[0];
$extension = $fileParts[1];
$filename = $date . '.' . $extension;

// Reupload resized image
if ($filetype === 'image/jpeg' || $filetype === 'image/jpg') {
    imagejpeg($filecreate, $dst . $filename);
} elseif ($filetype === 'image/png') {
    imagepng($filecreate, $dst . $filename);
}

// Remove the original upload
unlink($fileupload);

return $filename;
}

```

5. Membuat file read.php, edit.php, dan create.php di lokasi application\views\backend\v\_berita kode seperti berikut

Read.php

```

<a href="<?= base_url('berita/create') ?>" title="Tambah Data">
    <button type="button">Tambah</button>
</a>

<table border="1" width="70%">

```

```

<a href="<?= base_url('berita/create') ?>" title="Tambah Data">
    <button type="button">Tambah</button>
</a>

<table border="1" width="70%">
<tr>
    <th>No</th>
    <th>Kategori</th>
    <th>Tanggal</th>
    <th>Status</th>
    <th>Judul</th>
    <th>Author</th>
    <th>Aksi</th>
</tr>
<?php
$no = 1;
foreach ($read as $row) : ?>
    <tr>
        <td><?= $no ?></td>
        <td><?= $row['nama_kategori'] ?></td>
        <td><?= $row['tgl_berita'] . ' ' . $row['jam_berita'] ?></td>
        <td><?= $row['st_berita'] ?></td>
        <td><?= $row['judul_berita'] ?></td>
        <td><?= $row['kd_admin'] ?></td>
        <td>
            <?php if ($row['st_berita'] === 'Publik') : ?>
                <a href="<?= base_url('berita/status/' . $row['id_berita'] . '/Blokir') ?>"
title="Ubah status ke Blok">
                    <button>Blokir</button>
                </a>
            <?php elseif ($row['st_berita'] === 'Blokir') : ?>
                <a href="<?= base_url('berita/status/' . $row['id_berita'] . '/Publik') ?>

```



```

" title="Ubah status ke Publik">

        <button>Publik</button>

    </a>

<?php endif; ?>

    <a href="<?= base_url('berita/edit/' . $row['id_berita']) ?>" title="Ubah">

        <button>Ubah</button>

    </a>

    <a href="<?= base_url('berita/delete/' . $row['id_berita']) ?>" title="Delete"
onclick="return confirm('Yakin Mau dihapus <?= $row['judul_berita'] ?>')">

        <button>Hapus</button>

    </a>

</td>

</tr>

<?php
$no++;
endforeach;
?>
</table>

<!--EndTable-->

```

## Create.php

```

<!--Form-->

<form method="post" action="<?= base_url('berita/save') ?>" enctype="multipart/form-data">

    <label>Judul Berita</label><br>

    <input type="text" name="judul_berita" placeholder="Masukan Judul Berita" required
oninvalid="this.setCustomValidity('Judul Berita Harus Di Isi')" oninput="setCustomValidity('')"><p></p>

    <label>Kategori</label><br>

    <select name="id_kategori" required>

        <option value="0" selected>- Pilih kategori -</option>

        <?php foreach ($kategori as $r) : ?>

            <option value="<?= $r['id_kategori'] ?>"><?= $r['nama_kategori'] ?></option>

```

```

        <?php endforeach; ?>
    </select><p></p>

    <label>Isi Berita</label><br>
    <textarea name="isi_berita" placeholder="Isi Berita" rows="20" cols="80" required></textarea><p></p>

    <label>Foto Berita</label><br>
    <input type="file" name="img_berita" required><p></p>

    <button type="submit">Simpan</button>

    <a href="<?= base_url('berita') ?>"><button type="button">Batal</button></a>
</form>
<!--EndForm-->

```

## Edit.php

```

<!--Form-->

<form method="post" action="<?= base_url('berita/update/' . $edit['id_berita']) ?>" enctype="multipart/form-data">

    <label>Judul Berita</label><br>

    <input type="text" name="judul_berita" placeholder="Masukan Judul Berita" value="<?= $edit['judul_berita'] ?>" required oninvalid="this.setCustomValidity('Judul Berita Harus Di Isi')" oninput="setCustomValidity('')"><p></p>

    <label>Tanggal</label><br>

    <input type="date" name="tgl_berita" value="<?= $edit['tgl_berita'] ?>" required oninvalid="this.setCustomValidity('Tanggal Berita Harus Di Isi')" oninput="setCustomValidity('')"><p></p>

    <label>Jam</label><br>

    <input type="time" name="jam_berita" value="<?= $edit['jam_berita'] ?>" required oninvalid="this.setCustomValidity('Jam Berita Harus Di Isi')" oninput="setCustomValidity('')"><p></p>

    <label>Kategori</label><br>

    <select name="id_kategori" required>

        <option value="<?= $edit['id_kategori'] ?>"><?= $edit['nama_kategori'] ?></option>

        <?php foreach ($kategori as $r) : ?>

            <option value="<?= $r['id_kategori'] ?>"><?= $r['nama_kategori']; ?></option>

        <?php endforeach; ?>

    </select><p></p>

```

```

</select><p></p>

<label>Isi Berita</label><br>

<textarea name="isi_berita" placeholder="Isi Berita" rows="20" cols="80" required><?=$edit['isi_berita']
?></textarea><p></p>

<br>

<label>Ganti Foto</label><br>

<input type="file" name="img_berita"><p></p>

<button type="submit">Perbaharui</button>

<a href="<?=$base_url('admin') ?>"><button type="button">Batal</button></a>

</form>

<!--EndForm-->

```

6. Untuk Routes.php, tambahkan code ini

```

$routes->get('berita', 'berita::index');
$routes->get('berita/create', 'berita::create');
$routes->get('berita/edit/{:any}', 'berita::edit/$1');
$routes->get('berita/update/{:any}', 'berita::update/$1');
$routes->get('berita/delete/{:any}', 'berita::delete/$1');
$routes->get('berita/delete/{:any}', 'berita::publik/$1');
$routes->post('berita/save', 'berita::save');

```

7. Setelah itu buka link ini : <http://localhost:8080/berita> untuk melihat hasilnya

Tambah						
No	Kategori	Tanggal	Status	Judul	Author	Aksi

Judul Berita

Barcelona

Kategori

Olahraga

- Pilih kategori -

Olahraga

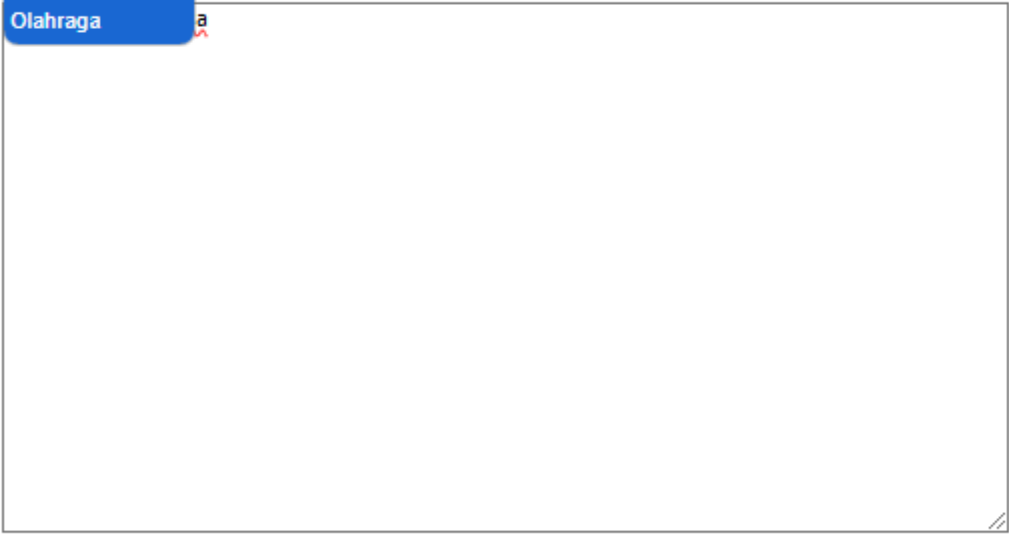


Foto Berita

Choose File css.png

Simpan Batal

Tambah								
No	Kategori	Tanggal	Status	Judul	Author	Aksi		
1	Olahraga	2024-10-30 05:36:27	Blokir	Barcelona	05	Publik	Ubah	Hapus