

**LAPORAN PRAKTIKUM PEMROGRAMAN WEB
WEBSITE DENGAN FRAMEWORK CODEIGNITER**



NAMA ANGGOTA :

- | | |
|--------------------------|------------|
| 1. ALVIAN PUTRA HARDIADI | (12221819) |
| 2. M. HABIB DWI PUTRA | (12221904) |
| 3. RETNO WIDYARTI | (12221911) |

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA dan ILMU KOMPUTER EL
RAHMA YOGYAKARTA**

TAHUN 2023/2024

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT, karena atas rahmat dan hidayah-Nya, kami dapat menyelesaikan laporan praktikum ini dengan baik. Laporan praktikum ini disusun sebagai salah satu tugas mata kuliah Pemrograman Web dengan topik "Website dengan Framework CodeIgniter" pada semester ini.

Framework CodeIgniter (CI) dipilih sebagai alat bantu dalam pengembangan aplikasi web ini karena kemudahannya dalam mengelola struktur proyek serta keandalannya dalam mengoptimalkan proses pengembangan. Melalui praktikum ini, kami dapat memahami lebih dalam konsep dasar pemrograman web dengan memanfaatkan CI sebagai kerangka kerja yang mendukung.

Laporan ini mencakup langkah-langkah praktis dalam membangun aplikasi web mulai dari perencanaan, pengaturan proyek, pembuatan fitur, hingga pengujian aplikasi. Harapan kami, laporan ini dapat memberikan gambaran yang jelas dan komprehensif bagi pembaca mengenai proses pengembangan sebuah website menggunakan CI.

Tak lupa, kami mengucapkan terima kasih kepada semua pihak yang telah membantu dalam proses pembuatan laporan ini, baik itu dosen pengampu, asisten praktikum, teman-teman seperjuangan, serta pihak-pihak lain yang turut serta mendukung. Semoga laporan ini bermanfaat dan dapat memberikan wawasan yang lebih luas dalam dunia pemrograman web.

Akhir kata, segala kesalahan dan kekurangan yang ada dalam laporan ini, kami mohon maaf yang sebesar-besarnya. Semoga Allah SWT senantiasa memberkahi dan melimpahkan rahmat-Nya kepada kita semua.

Wassalamu'alaikum Wr. Wb.

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI.....	3
BAB I.....	4
PENDAHULUAN	4
Latar Belakang	4
Rumusan Masalah	4
Landasan Teori.....	4
BAB II.....	7
PEMBAHASAN	7
Pengertian CodeIgniter.....	7
Fitur-Fitur CodeIgniter	7
Cara Kerja CodeIgniter	9
Keuntungan CodeIgniter	9
Kekurangan CodeIgniter	10
Instalasi CodeIgniter 4	10
Instalasi CodeIgniter 4 melalui Composer	10
LANGKAH LANGKAH PEMBUATAN	13
BAB III.....	28
PENUTUP.....	28
Kesimpulan.....	28

BAB I

PENDAHULUAN

Latar Belakang

Laporan ini membahas pembuatan aplikasi "To-Do" menggunakan CodeIgniter 4 (CI 4), sebuah framework PHP yang populer dalam pengembangan aplikasi web. Aplikasi "To-Do" dipilih karena sifatnya yang sederhana namun esensial dalam membantu pengguna mengelola dan mengatur tugas-tugas sehari-hari secara efisien. Dalam konteks pembelajaran, proyek ini bertujuan untuk menggali dan menerapkan prinsip-prinsip dasar MVC (Model-View-Controller) yang menjadi fondasi CI 4, serta untuk memahami penggunaan basis data dalam menyimpan dan mengelola informasi tugas pengguna. Melalui proyek ini, diharapkan dapat meningkatkan pemahaman dalam mengimplementasikan fitur-fitur dasar seperti menambahkan, menghapus, dan menandai selesai suatu tugas dalam aplikasi web, serta memperluas pengetahuan tentang praktik-praktik terbaik dalam pengembangan aplikasi berbasis web menggunakan teknologi modern. Dukungan komunitas yang luas terhadap CI 4 juga menjadi nilai tambah dalam mengakses sumber daya dan panduan yang mempercepat proses pengembangan serta memperluas kemampuan teknis pengembang dalam menangani tantangan dalam proyek ini.

Rumusan Masalah

1. Apa itu Codeigniter?
2. Kelebihan Codeigniter?
3. Cara kerja Codeigniter?
4. Cara Instalasi Codeigniter 4?
5. Implementasi Codeigniter 4 dalam Todo App ?

Landasan Teori

1. Pengertian Framework CodeIgniter 4

Framework adalah kerangka kerja yang menyediakan struktur dan alat bantu untuk membangun aplikasi web secara efisien. CodeIgniter 4 (CI4) adalah salah satu framework PHP yang ringan dan cepat, dirancang untuk memudahkan pengembangan aplikasi web dengan pendekatan MVC (Model-View-Controller).

2. Arsitektur MVC dalam CodeIgniter 4

MVC (Model-View-Controller) adalah paradigma desain yang terdiri dari tiga komponen utama:

- a. Model: Bertanggung jawab untuk mengelola data dan logika bisnis aplikasi.
- b. View: Bertanggung jawab untuk menampilkan informasi kepada pengguna, biasanya berupa halaman web.
- c. Controller: Bertanggung jawab untuk mengatur interaksi pengguna, memproses input, dan mengoordinasikan Model dan View.

Dalam CI4, struktur MVC ini membantu memisahkan logika aplikasi dari tampilan, sehingga mempermudah pengelolaan dan pengembangan aplikasi.

3. Fitur Utama CodeIgniter 4

- a. Routing: CI4 menyediakan sistem routing yang fleksibel untuk mengarahkan permintaan HTTP ke fungsi-fungsi yang sesuai.
- b. Database Support: Mendukung berbagai jenis database dengan menggunakan Query Builder atau ORM (Object-Relational Mapping).
- c. Security: Memiliki fitur keamanan seperti proteksi terhadap XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery), dan hashing password.
- d. Performance: Framework ini dikenal karena kinerjanya yang cepat, baik dalam proses pengolahan maupun dalam memuat halaman.
- e. Pengelolaan File dan Pengujian: Memiliki utilitas bawaan untuk pengelolaan file dan dukungan yang baik untuk pengujian unit dan fungsional.

4. Instalasi dan Konfigurasi CodeIgniter 4

Untuk memulai pengembangan dengan CI4, langkah-langkah dasar meliputi instalasi framework, konfigurasi file lingkungan (environment), dan pengaturan database. CI4 mendukung pengaturan yang mudah dan jelas melalui file konfigurasi yang terstruktur.

5. Pengembangan Aplikasi dengan CI4

Langkah-langkah dalam pengembangan aplikasi web menggunakan CI4 meliputi:

- a. Desain Database: Merancang struktur database sesuai kebutuhan aplikasi.
- b. Membuat Model: Implementasi fungsi logika bisnis dan interaksi dengan database.
- c. Membuat Controller: Menangani permintaan pengguna dan mengatur interaksi antara Model dan View.
- d. Membuat View: Menampilkan informasi kepada pengguna dalam bentuk halaman web.

6. Kelebihan dan Kekurangan CodeIgniter 4

Kelebihan:

- a. Ringan dan cepat.
- b. Mudah dipelajari dan diimplementasikan.
- c. Dukungan komunitas yang besar.
- d. Memiliki dokumentasi yang lengkap dan jelas.

Kekurangan:

- a. Tidak selengkap framework besar seperti Laravel dalam hal fitur bawaan.
- b. Beberapa fitur mungkin perlu diimplementasikan secara manual.

7. COMPOSER

Composer adalah sebuah tools manajemen dependensi untuk bahasa pemrograman PHP. Dalam konteks pemrograman, *Composer* adalah sebuah aplikasi yang digunakan untuk mengelola dan mengatur dependensi seperti library, framework, atau paket lain yang diperlukan dalam proyek PHP. Dengan menggunakan Composer, Anda dapat dengan mudah mengelola dependensi proyek dengan mendefinisikan library atau paket yang diperlukan dalam berkas konfigurasi `composer.json`. Composer kemudian akan mengelola unduhan, instalasi, dan

pembaruan dari dependensi-dependensi ini. *Composer* pertama kali dikembangkan oleh Nils Adermann dan Jordi Boggiano di tahun 2011, dan terus dikembangkan sampai saat ini. Dalam beberapa tahun terakhir, *Composer* telah menjadi standar untuk manajemen dependensi dalam pengembangan PHP, dan banyak proyek besar yang mengandalkan tools ini untuk mengatur komponen-komponen perangkat lunak yang dikembangkan.

BAB II

PEMBAHASAN

Pengertian CodeIgniter

CodeIgniter merupakan salah satu framework yang populer di dunia pengembangan aplikasi dan web dengan menggunakan konsep Model-View-Controller (MVC). Sebagai platform open-source, CodeIgniter menjadi pilihan utama bagi para pengembang yang bekerja dengan bahasa pemrograman PHP. Penggunaan CodeIgniter bertujuan untuk mempercepat dan menyederhanakan proses pengembangan proyek.

Diperkenalkan pada tahun 2006 oleh Rick Ellis, CodeIgniter membanggakan berbagai pustaka (library) yang lebih luas dibandingkan dengan beberapa framework lainnya. Keistimewaan CodeIgniter terletak pada sifatnya yang ringan, sehingga tidak memberikan beban besar pada sumber daya website. Selain itu, sebagai framework open-source, CodeIgniter dapat diakses dan diunduh secara gratis oleh para pengembang.

CodeIgniter mengadopsi model MVC, suatu pendekatan yang sangat cocok untuk merancang aplikasi atau website yang bersifat dinamis. Konsep MVC digunakan untuk memisahkan komponen utama ke dalam beberapa bagian, membentuk pola yang efektif saat proses pengembangan aplikasi.

Fitur-Fitur CodeIgniter

CodeIgniter dikenal sebagai pilihan utama para pengembang berkat kinerjanya yang unggul dan berbagai fitur lengkap yang mempermudah tugas pengembangan. Berikut adalah fitur-fitur utama yang membedakan CodeIgniter.

1. Model-View-Controller (MVC)

Fitur ini merupakan favorit di antara pengembang dan menjadikan CodeIgniter sebagai framework yang sangat dihargai. Model-View-Controller (MVC) adalah praktik standar industri yang memisahkan data, logika, dan presentasi dalam aplikasi web.

2. Library Bawaan

Fitur ini memungkinkan pengguna menemukan berbagai solusi seputar pemrograman, termasuk buku, tutorial, dan panduan penggunaan terkait CodeIgniter. Pengembang CodeIgniter memiliki akses ke dokumentasi lengkap yang menjelaskan cara kerja sistem aplikasi serta informasi lainnya seputar pengembangan PHP.

Dalam library bawaan ini, tersedia tutorial atau petunjuk yang mencakup masalah umum yang pernah dihadapi oleh pengembang lain dan solusi yang ditemukan, sehingga bisa menjadi referensi bagi pengguna lain.

3. Sangat Ringan

Dalam era di mana kecepatan pemuatan aplikasi menjadi prioritas, CodeIgniter memberikan waktu pemuatan kurang dari satu detik setelah instalasi. Dengan respons cepat kurang dari 50 milisekon, CodeIgniter menjadi lebih cepat dibandingkan dengan beberapa framework lainnya.

4. Source Code yang Kompak

Ukuran source code yang kecil menjadi keunggulan, menyederhanakan implementasi dan pembaruan sistem aplikasi. CodeIgniter memiliki source code yang kurang lebih hanya 2 MB, memudahkan pengguna untuk memahami CodeIgniter dan cara kerjanya.

Manfaat CodeIgniter

Saat ingin membangun situs web yang profesional, modern, dan efisien, CodeIgniter menjadi salah satu pilihan terbaik. Tidak hanya memiliki performa dan fitur yang sangat baik, pengguna CodeIgniter juga dapat meraih sejumlah manfaat tambahan yang signifikan.

Berikut adalah beberapa keuntungan yang diberikan CodeIgniter, sebagaimana dijelaskan oleh Seasia Info Tech:

1. Otomatis Melacak Bug

CodeIgniter memiliki kemampuan untuk secara otomatis melacak dan mengidentifikasi bug, memudahkan pengguna dalam proses debugging dan meningkatkan kualitas keseluruhan dari situs web yang dikembangkan.

2. Penyesuaian Desain dan Sistem

Pengguna memiliki keleluasaan untuk mengatur desain dan sistem CodeIgniter sesuai dengan kebutuhan spesifik mereka. Fleksibilitas ini memungkinkan pembangunan situs web yang sesuai dengan visi dan tujuan pengguna.

3. Keamanan yang Terjamin

CodeIgniter menawarkan tingkat keamanan yang tinggi, memberikan jaminan keamanan bagi situs web yang dikembangkan dengan framework ini. Ini membantu melindungi situs web dari potensi serangan dan ancaman keamanan.

4. Migrasi Data yang Mudah

Framework ini memungkinkan pengguna untuk dengan mudah melakukan migrasi data dari satu server ke server lainnya. Kemampuan ini sangat berguna ketika pengguna perlu mentransfer atau mengelola data di lingkungan server yang berbeda.

5. Peningkatan Peringkat SEO

CodeIgniter menyediakan fitur yang memudahkan pengguna untuk meningkatkan peringkat SEO situs web mereka. Dengan dukungan yang kuat dari framework ini, pengguna dapat mengoptimalkan kontennya agar lebih terindeks oleh mesin pencari dan mendapatkan visibilitas yang lebih baik online.

Cara Kerja CodeIgniter

CodeIgniter adalah sebuah framework yang mengadopsi pola Model-View-Controller (MVC). MVC, yang merupakan singkatan dari Model-view-controller, dapat dijelaskan sebagai berikut: ketika Anda mencari informasi tertentu, controller akan menanggapi permintaan Anda, kemudian menyajikan informasi yang dicari jika tersedia.

Sebagai contoh sederhana, bayangkan jika Anda ingin mencari informasi dengan id=3. Pada saat tersebut, controller akan menanggapi permintaan Anda, dan kemudian meminta Model untuk mengambil data yang terkait dengan id=3. Model akan mengembalikan data tersebut ke controller. Controller selanjutnya akan menyajikan data tersebut kepada View dalam format yang mudah dibaca oleh manusia (Human-readable format). Hasilnya akan ditampilkan pada browser Anda.

CodeIgniter mengikuti alur kerja sebagai berikut:

1. Setiap ada request pada website, maka akan diarahkan ke halaman index.php.
2. Routing akan menentukan alur request dari pengguna. Jika halaman yang diminta ada pada caching, maka routing akan melakukan langkah 3.
3. Jika routing mengarahkan ke caching, maka halaman yang ditampilkan adalah halaman yang telah disimpan ke dalam cache sebelumnya.
4. Jika routing mengarahkan ke security, maka seluruh data dari pengguna akan difilter untuk meningkatkan keamanan sebelum diarahkan ke controller.
6. Controller akan memanggil model, library, helper, dan tools lainnya yang diperlukan bagi halaman website yang diminta.
7. View akan ditampilkan sesuai dengan permintaan pengguna. Dengan begitu, pengguna dapat melihat halaman website yang sesuai dengan yang diinginkan.
8. Halaman website akan disimpan ke dalam cache. Tujuannya untuk meningkatkan performa website karena request pengguna tidak perlu diarahkan ke controller lagi.

Keuntungan CodeIgniter

CodeIgniter menyajikan sejumlah keunggulan yang mungkin tidak dapat ditemukan pada framework lain. Berikut beberapa manfaat dan kelebihan yang ditawarkan CodeIgniter:

- Mudah Dipelajari: CodeIgniter merupakan framework yang sangat mudah dipelajari, terutama jika Anda sudah menguasai PHP. Anda dapat menguasai CodeIgniter dalam waktu singkat untuk membangun dan mengembangkan aplikasi.
- Ukuran Kecil: Framework CodeIgniter memiliki ukuran yang sangat kecil, sekitar 2MB. Ukuran yang kecil ini memudahkan dalam menjalankan dan mengupdate

CodeIgniter. Dibandingkan dengan jenis framework lainnya, CodeIgniter jelas lebih ramah terhadap ruang penyimpanan perangkat.

- **Dapat Dimodifikasi:** CodeIgniter sudah dilengkapi dengan library yang berisi banyak bantuan dan petunjuk. Jika Anda tidak menemukan yang Anda cari, Anda dapat dengan mudah menambahkan bantuan, petunjuk, atau library sendiri secara mandiri.
- **Sangat Cepat:** Pengguna umumnya lebih menyukai perangkat lunak yang mampu memproses proses dengan cepat. CodeIgniter hadir sebagai framework super cepat, di mana proses loading dan eksekusi perintah dapat diproses dalam waktu kurang dari 50 milisekon.
- **Fleksibel:** CodeIgniter dilengkapi dengan sejumlah fitur dan komponen canggih yang berdiri sendiri tanpa saling bergantung. Hal ini memungkinkan pengguna untuk meng-upgrade dan memperbaiki masalah tertentu dengan mudah.

Kekurangan CodeIgniter

Diantara beberapa keunggulan yang dimiliki codeigniter, terdapat beberapa kekurangan yang harus diantisipasi programmer jika ingin menggunakan framework ini. Berikut beberapa kekurangan yang dimiliki codeigniter :

- Walau terdapat berbagai perpustakaan (library) yang tersedia, CodeIgniter tidak dirancang untuk pengembangan situs web dengan skala besar atau kebutuhan perusahaan.
- CodeIgniter masih memiliki keleluasaan yang signifikan dalam penulisan kode, seperti kemampuan untuk secara bebas menambahkan file.
- Framework ini tidak sepenuhnya mencerminkan konsep Model-View-Controller (MVC) dengan benar, seperti contoh penggunaan perintah echo yang masih dapat dilakukan langsung pada file Controller.

Instalasi CodeIgniter 4

Sebelum melakukan instalasi CodeIgniter 4, ada beberapa hal yang perlu Anda persiapkan, yaitu :

- **XAMPP** : aplikasi server yang mendukung proses pengembangan aplikasi website pada server lokal (localhost) sebelum di hosting.
- **Composer** : tool pengelola package library PHP yang memudahkan Anda untuk menginstall dan melakukan update library yang digunakan.
- **Code Editor** : aplikasi yang digunakan untuk menulis kode pemrograman. Di tutorial ini, kami menggunakan Visual Studio Code (VS Code).

Selanjutnya, menginstall CodeIgniter 4 dengan cara instalasi melalui Composer.

Instalasi CodeIgniter 4 melalui Composer

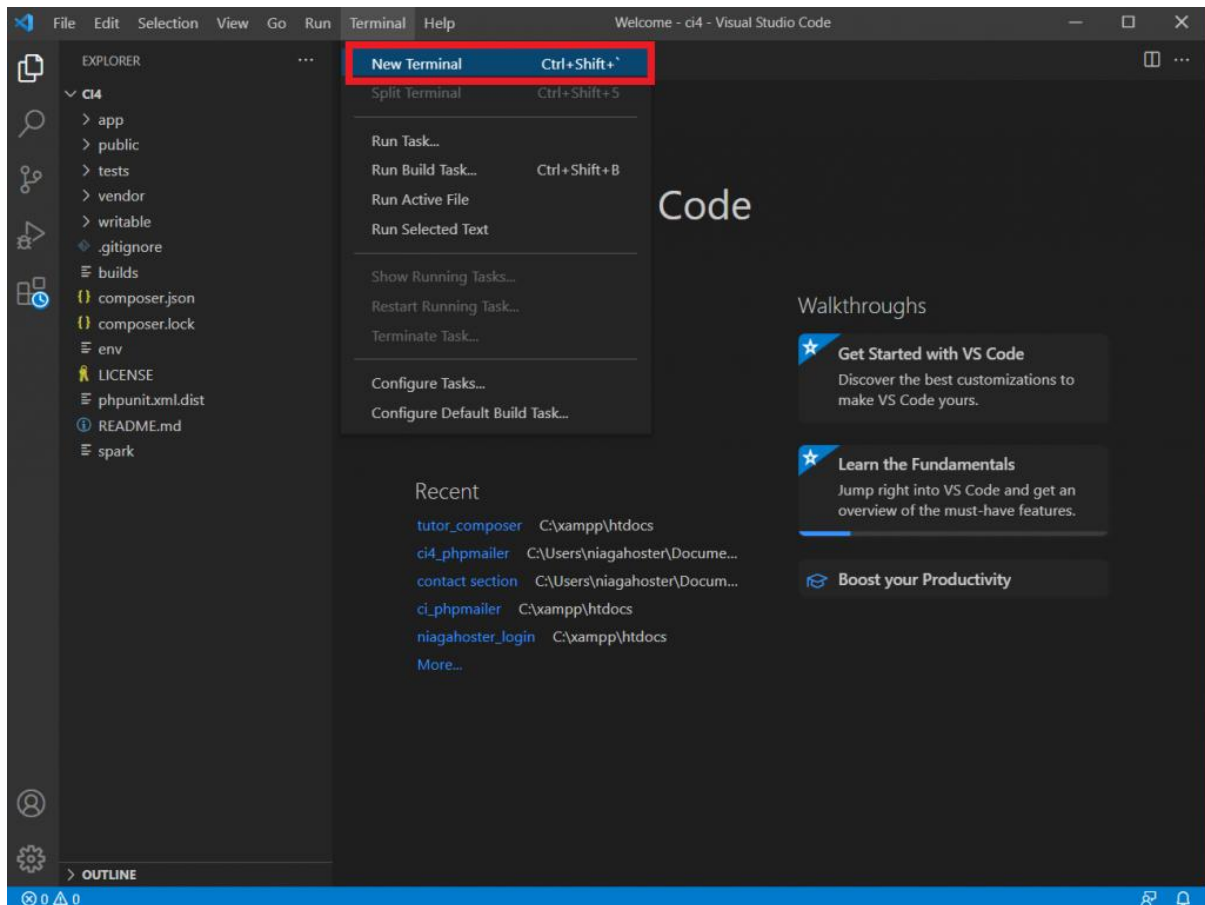
Cara menginstall CodeIgniter 4 adalah menggunakan Composer. Kelebihannya, path folder project Anda bisa dibuat dimana saja, tidak perlu di folder **htdocs**. Dan yang paling penting, proses install mudah hanya dengan satu baris perintah saja, yaitu :

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

```
C:\Users\niagahoster\Desktop>composer create-project codeigniter4/appstarter ci4
Creating a "codeigniter4/appstarter" project at "./ci4"
Installing codeigniter4/appstarter (v4.1.4)
- Downloading codeigniter4/appstarter (v4.1.4)
- Installing codeigniter4/appstarter (v4.1.4): Extracting archive
Created project in C:\Users\niagahoster\Desktop\ci4
Loading composer repositories with package information
Updating dependencies
Lock file operations: 42 installs, 0 updates, 0 removals
- Locking codeigniter4/framework (v4.1.4)
- Locking doctrine/instantiator (1.4.0)
- Locking fakerphp/faker (v1.16.0)
- Locking kint-php/kint (3.3)
```

Pada tutorial ini, kami membuat folder project di **Desktop** dengan nama **ci4**.

Setelah folder berhasil terbuat, buka folder tersebut dengan Visual Studio Code. Jika project telah terbuka, bukalah terminal di VS Code :



Lalu, jalankan perintah di bawah ini untuk menginstall semua library yang dibutuhkan CodeIgniter 4 :

```
composer install -vvv
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\niagahoster\Desktop\ci4> composer install -vvv
Reading ./composer.json (C:\Users\niagahoster\Desktop\ci4\composer.json)
Loading config file ./composer.json (C:\Users\niagahoster\Desktop\ci4\composer.json)
Checked CA file C:\xampp\apache\bin\curl-ca-bundle.crt: valid
Executing command (C:\Users\niagahoster\Desktop\ci4): git branch -a --no-color --no-abbrev -v
Executing command (C:\Users\niagahoster\Desktop\ci4): git describe --exact-match --tags
Executing command (CWD): git --version
Executing command (C:\Users\niagahoster\Desktop\ci4): git log --pretty="%H" -n1 HEAD --no-show-signature
```

Jika proses instalasi sudah selesai dilakukan, Anda dapat mengakses CodeIgniter 4 dengan menjalankan perintah :

php spark serve

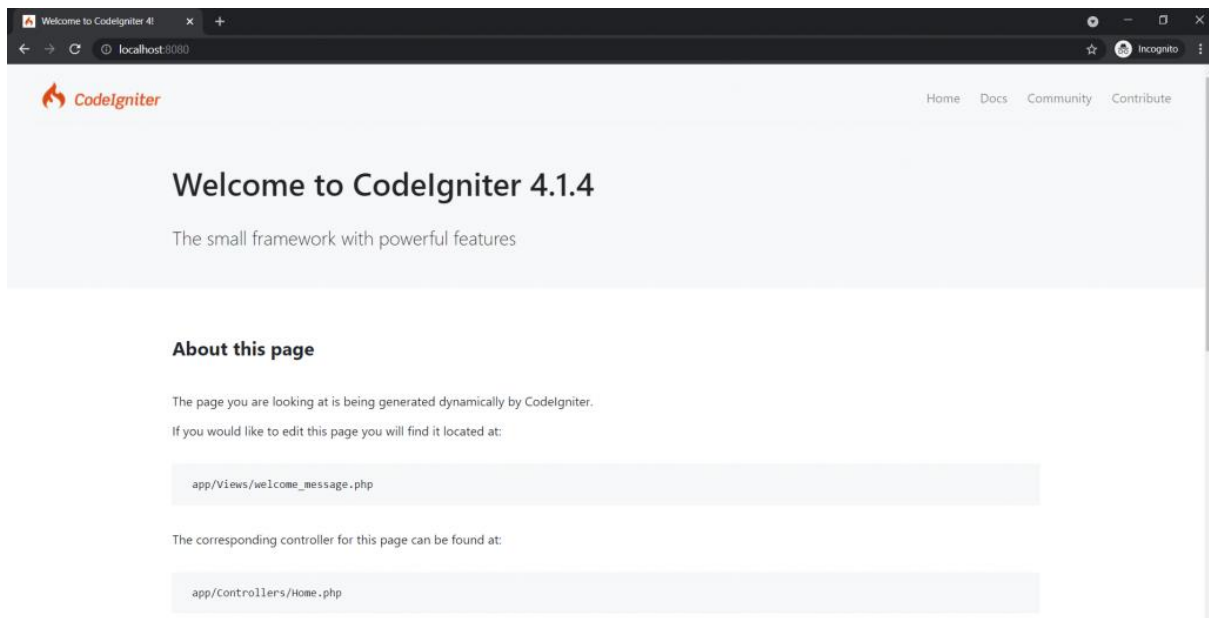
```
PS C:\Users\niagahoster\Desktop\ci4> php spark serve

CodeIgniter v4.1.4 Command Line Tool - Server Time: 2021-09-07 20:05:39 UTC-05:00

CodeIgniter development server started on http://localhost:8080
Press Control-C to stop.
[Wed Sep 8 08:05:39 2021] PHP 8.0.9 Development Server (http://localhost:8080) started
```

Gambar di atas menunjukkan bahwa local development server sudah berjalan.

Selanjutnya, bukalah web browser Anda dan akses **localhost:8080**. Halaman awal CodeIgniter akan tampil seperti pada gambar di bawah ini :



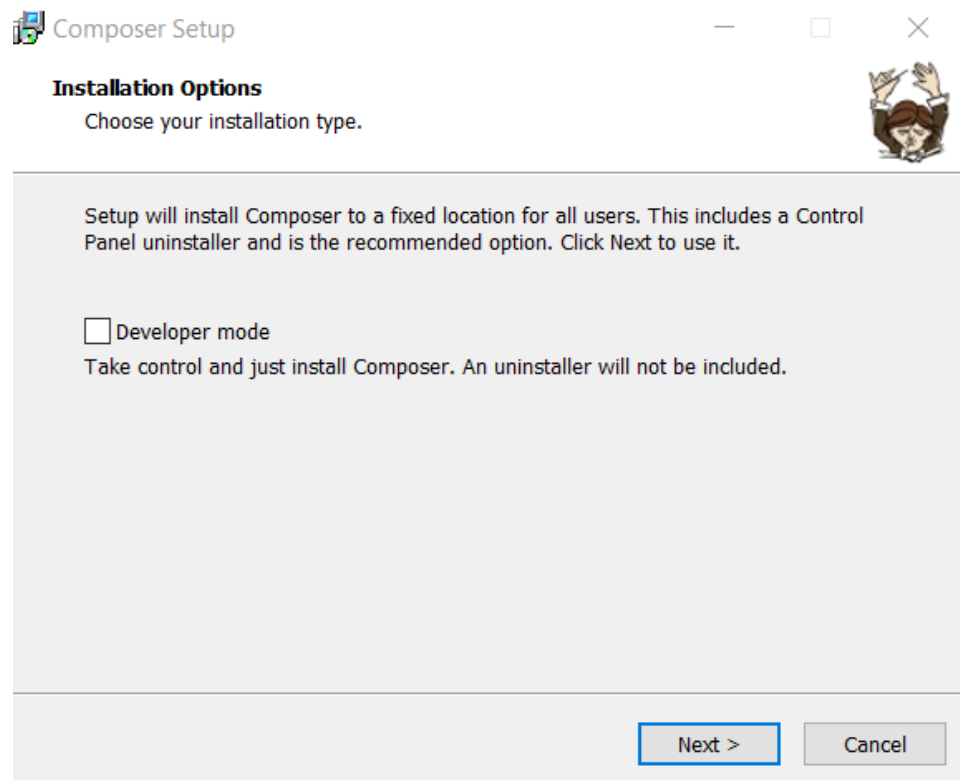
LANGKAH LANGKAH PEMBUATAN

1. Persiapan sebelum membuat :

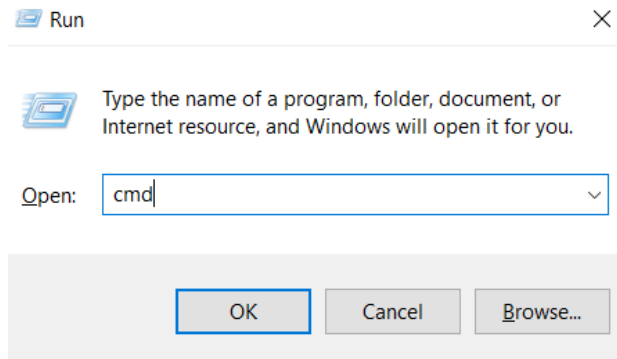
a. Install Composer

Langkah – langkah instal :

1. Gunakan XAMPP.
2. Unduh versi terbaru Composer.
3. Jalankan (run) Composer installation wizard. Abaikan jika Anda diminta untuk mengaktifkan mode developer. Lanjutkan proses install.



4. Muncul jendela yang akan meminta Anda untuk mencari baris perintah PHP. Secara default, baris perintah dapat ditemukan di C:/xampp/php/pjp.exe. Setelah itu, klik Next.
5. Permintaan Pengaturan Proxy (Proxy Settings) akan muncul. Kosongkan boks (tidak perlu ditandai) dan lewati bagian ini dengan meng-klik opsi Next. Klik Install di jendela berikutnya.
6. Buka command prompt. Tekan tombol CTRL+R, ketik “cmd” di dalam kolom prompt, dan klik OK.



7. Jalankan perintah berikut ini: `composer`

Selesai! Kini Composer berhasil terinstall di komputer Windows . Installer akan menambahkan Composer secara otomatis ke variabel PATH. Buka command prompt dan jalankan software dari mana saja.

b. Install VS code

Langkah – langkah :

1. Buka web <https://code.visualstudio.com/> dan download sesuai dengan sistem operasi komputer kamu.
2. Setelah berhasil didownload, silakan klik 2x hasil download tadi untuk menginstall VSCode nya.
3. Pada bagian License Agreement klik I accept the agreement lalu klik Next.
4. Pada bagian Select Destination Location biarkan apa adanya lalu klik Next.
5. Pada bagian Select Start Menu Folder klik Next.
6. Pada bagian Select Additional Tasks klik Next.
7. Pada bagian Ready to Install, klik Install.
8. Tunggu proses Installing selesai.
9. Setelah itu klik Finish.

c. Install XAMPP

Berikut adalah tiga langkah mudah untuk menginstal XAMPP di komputer Anda:

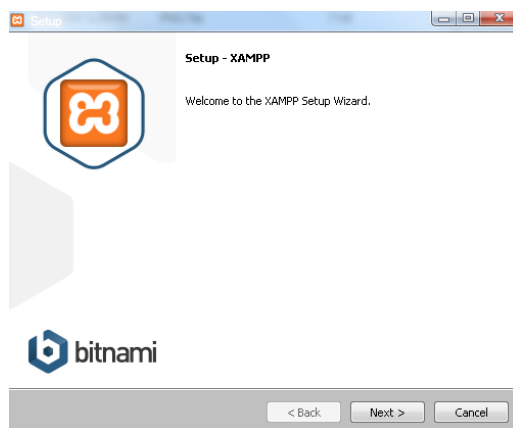
Download XAMPP

- a. Download XAMPP terbaru melalui website Apache Friends berikut ini.

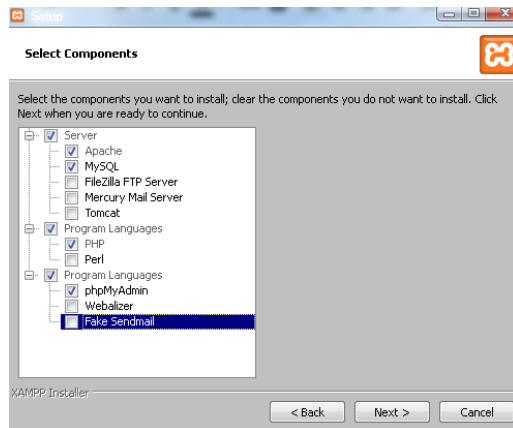


Instal XAMPP

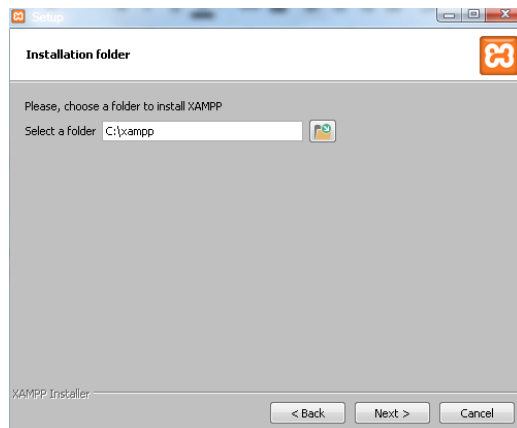
Lakukan instalasi setelah Anda selesai mengunduh. Selama proses instalasi mungkin Anda akan melihat pesan yang menanyakan apakah Anda yakin akan menginstalnya. Silakan tekan Yes untuk melanjutkan instalasi. Klik tombol Next.



- b. Pada tampilan selanjutnya akan muncul pilihan mengenai komponen mana dari XAMPP yang ingin dan tidak ingin Anda instal. Beberapa pilihan seperti Apache dan PHP adalah bagian penting untuk menjalankan website dan akan otomatis diinstal. Silakan centang MySQL dan phpMyAdmin, untuk pilihan lainnya biarkan saja.



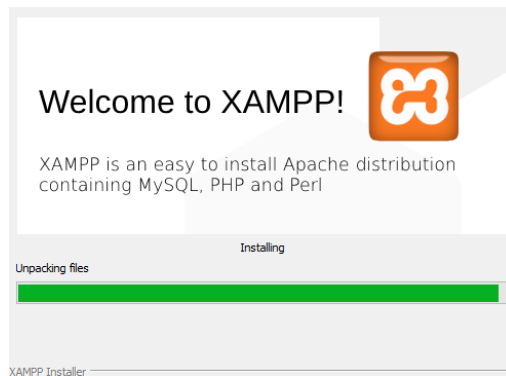
Berikutnya silakan pilih folder tujuan dimana XAMPP ingin Anda instal. Contohnya di direktori *C:\xampp*.



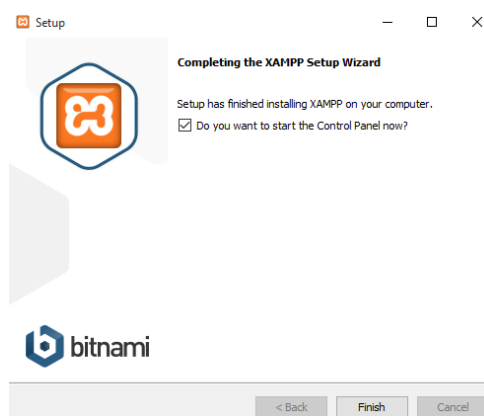
Pada halaman selanjutnya, akan ada pilihan apakah Anda ingin menginstal Bitnami untuk XAMPP, dimana nantinya dapat Anda gunakan untuk install WordPress, Drupal, dan Joomla secara otomatis.



Pada langkah ini proses instalasi XAMPP akan dimulai. Silakan klik tombol Next.

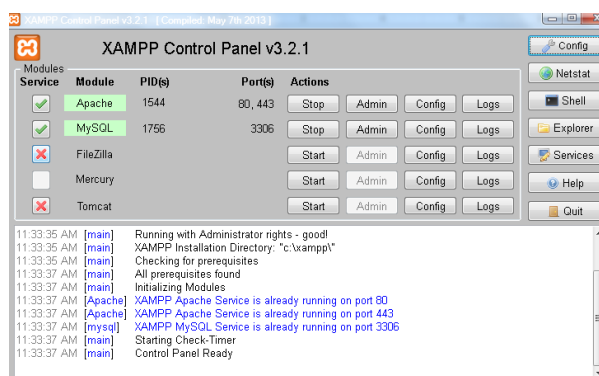


Setelah berhasil diinstal, akan muncul notifikasi untuk langsung menjalankan control panel. Silakan klik Finish.



Jalankan XAMPP

Silakan buka aplikasi XAMPP kemudian klik tombol Start pada Apache dan MySQL. Jika berhasil dijalankan, Apache dan MySQL akan berwarna hijau seperti gambar di bawah ini.



Untuk melakukan pengecekan, silakan akses link berikut melalui browser Anda *http://localhost*.

c. Install PHP Versi 7.2+

1. Unduh file PHP

Dapatkan paket ZIP Aman Thread PHP x64 terbaru dari <https://windows.php.net/download/> .

2. Ekstrak file

Buat folder baru di root C:\drive Anda dan ekstrak konten ZIP ke dalamnya. Dapat menginstal PHP di mana saja pada sistem Anda, namun Anda harus mengubah jalur yang dirujuk di bawah ini jika Anda menggunakan selain C:\php.

3. Konfigurasi php.ini

File konfigurasi PHP adalah `php.ini`. Awalnya ini tidak ada, jadi salin `C:\php\php.ini-development` ke `C:\php\php.ini`. Konfigurasi default ini menyediakan pengaturan pengembangan yang melaporkan semua kesalahan dan peringatan PHP. Anda dapat mengedit `php.ini` dalam editor teks, dan Anda mungkin perlu mengubah baris seperti yang disarankan di bawah ini (gunakan pencarian untuk menemukan pengaturannya). Dalam kebanyakan kasus, Anda harus menghapus tanda titik koma (;) di depan untuk menghapus komentar pada suatu nilai.

Pertama, aktifkan ekstensi apa pun yang diperlukan sesuai dengan perpustakaan yang ingin Anda gunakan. Ekstensi berikut ini cocok untuk sebagian besar aplikasi termasuk WordPress:

```
extension=curl
extension=gd
extension=mbstring
extension=pdo_mysql
```

Jika Anda ingin mengirim email menggunakan fungsi PHP `mail()`, masukkan detail server SMTP di [mail function] bagian tersebut (pengaturan ISP Anda harus sesuai):

```
[mail function]
; For Win32 only.
; http://php.net/smtp
SMTP = mail.myisp.com
; http://php.net/smtp-port
smtp_port = 25

; For Win32 only.
; http://php.net/sendmail-from
sendmail_from = my@emailaddress.com
```

Langkah 4: Tambahkan C:\php ke PATH variabel lingkungan

Untuk memastikan Windows dapat menemukan PHP yang dapat dieksekusi, Anda harus menambahkannya ke PATH variabel lingkungan. Klik tombol Start Windows dan ketik “lingkungan”, lalu klik Edit variabel lingkungan sistem. Pilih tab Tingkat Lanjut, dan klik tombol Variabel Lingkungan.

Gulir ke bawah daftar Variabel sistem dan klik Jalur , diikuti dengan tombol Edit . Klik Baru dan tambahkan C:\php.
Perhatikan bahwa Windows edisi lama menyediakan satu kotak teks dengan jalur yang dipisahkan oleh titik koma (;).

Sekarang oke jalan keluarmu. Anda tidak perlu melakukan boot ulang, tetapi Anda mungkin perlu menutup dan memulai ulang cmdterminal apa pun yang Anda buka.

4. Konfigurasi PHP sebagai modul Apache

Pastikan Apache tidak berjalan dan buka C:\Apache24\conf\httpd.conf file konfigurasinya di editor teks. Tambahkan baris berikut ke bagian bawah file untuk mengatur PHP sebagai modul Apache (ubah lokasi file jika perlu tetapi gunakan garis miring daripada garis miring terbalik Windows):

```
# PHP8 module
PHPIniDir "C:/php"
LoadModule php_module "C:/php/php8apache2_4.dll"
AddType application/x-httpd-php .php
```

Secara opsional, ubah DirectoryIndex pengaturan untuk digunakan index.php sebagai default dalam preferensi ke index.html. Pengaturan awalnya adalah:

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

Ubah menjadi:

```
<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
```

Simpan httpd.conf dan uji pembaruan dari cmdbaris perintah:

```
cd C:\Apache24\bin
httpd -t
```

Syntax OK akan muncul... kecuali jika Anda memiliki kesalahan dalam konfigurasi Anda .

Jika semuanya berjalan dengan baik, mulai Apache dengan httpd.

Langkah 6: Uji file PHP

Buat file baru dengan nama index.php di folder root halaman web Apache di C:\Apache24\htdocs. Tambahkan kode PHP berikut:

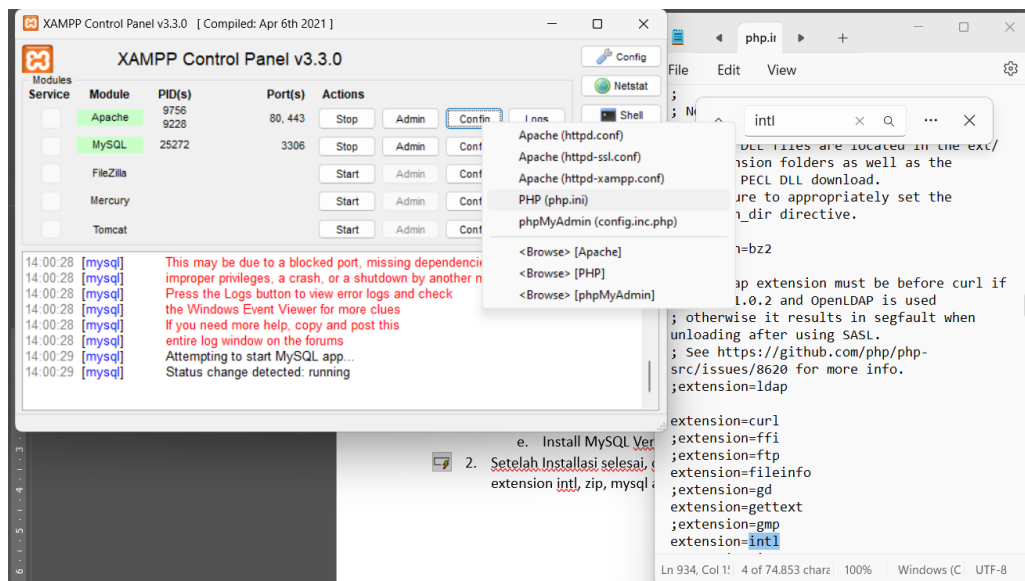
```
<?php
phpinfo();
```

?>

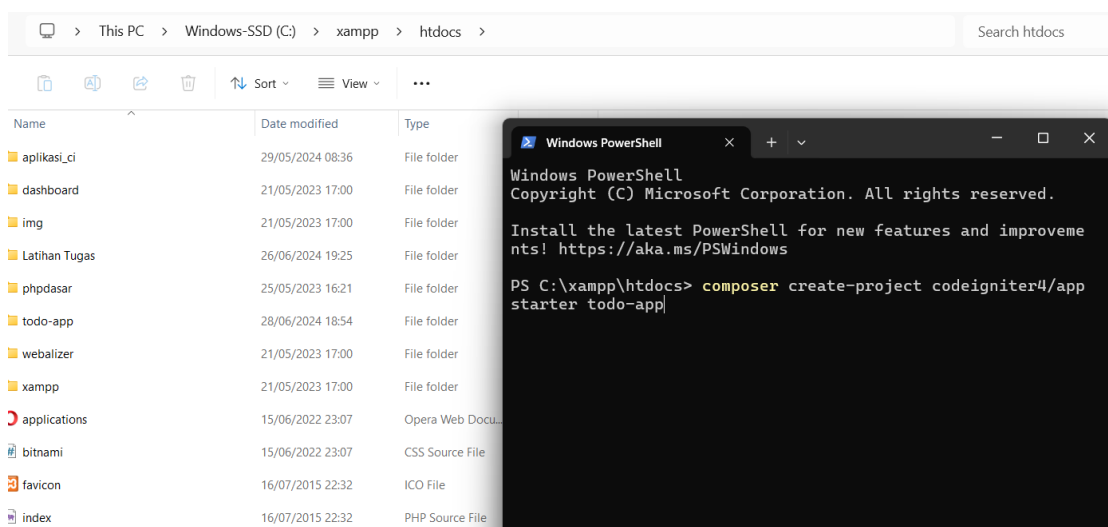
Buka browser web dan masukkan alamat server Anda: `http://localhost/`. Halaman versi PHP akan muncul, menampilkan semua pengaturan konfigurasi PHP dan Apache.

d. Install MySQL Versi 5.1+

- Setelah Installasi selesai, cek pada XAMPP klik pada bagian `php.ini` hilangkan ; pada bagian extension `intl`, `zip`, `mysql` agar dapat menginstall Code Igniter 4 melalui Composer.




- Setelah itu, buka pada bagian folder `htdocs` yang ada di `xampp`, lalu klik kanan dan cari `git bash` atau terminal ketikkan perintah berikut: `composer create-project codeigniter4/appstarter 'project-root'` terserah kalian mau namain apa.



- Bila sudah terinstal semua ketikkan perintah code "nama folder kalian"

```
Windows PowerShell
- Installing phpunit/php-timer (6.0.0): Extracting archive
- Installing phpunit/php-text-template (3.0.1): Extracting archive
- Installing phpunit/php-invoker (4.0.0): Extracting archive
- Installing phpunit/php-file-iterator (4.1.0): Extracting archive
- Installing theseer/tokenizer (1.2.3): Extracting archive
- Installing nikic/php-parser (v5.0.2): Extracting archive
- Installing sebastian/lines-of-code (2.0.2): Extracting archive
- Installing sebastian/complexity (3.2.0): Extracting archive
- Installing sebastian/code-unit-reverse-lookup (3.0.0): Extracting archive
- Installing phpunit/php-code-coverage (10.1.15): Extracting archive
- Installing phar-io/version (3.2.1): Extracting archive
- Installing phar-io/manifest (2.0.4): Extracting archive
- Installing myclabs/deep-copy (1.12.0): Extracting archive
- Installing phpunit/phpunit (10.5.24): Extracting archive
17 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating optimized autoload files
26 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found.
PS C:\xampp\htdocs> code todo-app
```

5. Nanti kalian akan masuk ke vscode dengan workspace folder yang sudah dibuat.
6. Tapi sebelum ke vscode, aktifkan dulu XAMPP dan masuk ke dalam phpmyadmin untuk membuat database. Buat database dengan nama "todos" buat juga table nya dengan nama todos pula. Buat isian table nya seperti ini :

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra
<input type="checkbox"/>	1	id 	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
<input type="checkbox"/>	2	todoname	varchar(45)	utf8mb4_general_ci		Tidak	Tidak ada		
<input type="checkbox"/>	3	description	varchar(45)	utf8mb4_general_ci		Tidak	Tidak ada		

7. Lalu masuk lagi kedalam vscode lagi, cari bagian env dan copy paste di dalam folder workspace nya. Ganti namanya menjadi ".env" dan isian nya akan seperti ini:

```
todo-app > .env
12
13 #-----
14 # ENVIRONMENT
15 #-----
16
17 CI_ENVIRONMENT = development
18
19 #-----
20 # APP
21 #-----
22
23 app.baseURL = 'http://localhost:8080'
24 # If you have trouble with `.` , you could also use `\_`.
25 # app_baseURL = ''
26 # app.forceGlobalSecureRequests = false
27 # app.CSPEnabled = false
28
29 #-----
30 # DATABASE
31 #-----
32
33 database.default.hostname = localhost
34 database.default.database = todos
35 database.default.username = root
36 database.default.password =
37 database.default.DBDriver = MySQLi
38 database.default.DBPrefix =
39 # database.default.port = 3306
40
41 # If you use MySQLi as tests, first update the values of Config\Database::$tests.
42 # database.tests.hostname = localhost
```

8. Setelah itu kita coba test dan jalankan dengan buka terminal vscode dan ketikkan php spark server. Pastikan muncul tampilan CodeIgniter 4. Bila muncul aman, bila tidak ada yang bermasalah.
9. Cari file database yang ada di config sesuaikan dengan ini :

```
Home.php  todos.php  Database.php X  .env  TodoModel.php
todo-app > app > Config > Database.php > PHP > Database > default
10 class Database extends Config
11     1 reference
26     public $defaultGroup = 'default';
27
28 /**
29  * The default database connection.
30  *
31  * @var array
32  */
33     0 references
34     public $default = [
35         'DSN' => '',
36         'hostname' => 'localhost',
37         'username' => '',
38         'password' => '',
39         'database' => 'todos',
40         'DBDriver' => 'MySQLi',
41         'DBPrefix' => '',
42         'pConnect' => false,
43         'DBDebug' => (ENVIRONMENT !== 'production'),
44         'charset' => 'utf8',
45         'DBCollat' => 'utf8_general_ci',
46         'swapPre' => '',
47         'encrypt' => false,
48         'compress' => false,
49         'strictOn' => false,
50         'failover' => [],
51         'port' => 3306,
52     ];
```

10. Lalu cek bagian routes :

```
Home.php  todos.php  Routes.php X  .env  TodoModel.php

todo-app > app > Config > Routes.php > ...
1  <?php
2
3  namespace Config;
4
5  // Create a new instance of our RouteCollection class.
6  $routes = Services::routes();
7
8  // Load the system's routing file first, so that the app and ENVIRONMENT
9  // can override as needed.
10 if (file_exists(SYSTEMPATH . 'Config/Routes.php')) {
11     require SYSTEMPATH . 'Config/Routes.php';
12 }
13
14 /*
15  * -----
16  * Router Setup
17  * -----
18  */
19 $routes->setDefaultNamespace('App\Controllers');
20 $routes->setDefaultController('Home');
21 $routes->setDefaultMethod('index');
22 $routes->setTranslateURIDashes(false);
23 $routes->set404Override();
24 $routes->setAutoRoute(true);
25
26 /*
27  * -----
28  * Route Definitions
29  * -----
30  */
31
```

11. Lalu cari folder views didalam nya, dan buat file baru dengan nama todos.php dengan isian berikut:

```
Home.php  todos.php X  .env  TodoModel.php

todo-app > app > Views > todos.php > html > body > main.container > div.container > form.row
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <!-- Required meta tags -->
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7
8      <!-- Bootstrap CSS -->
9      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-18mE4k8Q78iYhF1dvKuhfTA
10
11      <!-- Custom CSS -->
12      <link href="{<?>= base_url('css/styles.css'); ?>" rel="stylesheet">
13
14      <title>Todo App</title>
15      <style>
16          .logo {
17              width: 80px;
18              height: 90px;
19              margin-bottom: -10px; /* Mengatur jarak antara Logo dan garis bawah */
20          }
21          header {
22              padding-bottom: 10px; /* Mengurangi padding bottom pada header */
23          }
24      </style>
25  </head>
26  <body>
27      <main class="container">
28          <header class="py-2 mb-4 border-bottom">
29              <div class="container d-flex align-items-center">
30                  
31              </div>
32      </main>
33  </body>
34  </html>
```

```

42     </div>
43 </header>
44 <div class="container">
45     <p class="lead">Kelola tugas Anda dengan lebih efektif menggunakan Todo App. Tambahkan, edit, dan hapus daftar tugas Anda dengan mudah.
46     <?php if ($flashMessage = session()->getFlashdata('successMessage')): ?>
47     <div class="alert alert-success alert-dismissible fade show" role="alert">
48         <span><?=$flashMessage; ?></span>
49         <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
50     </div>
51     <?php endif; ?>
52
53     <?php
54         $errors = session()->get('errorsMessages');
55         session()->remove('errorsMessages');
56     ?>
57
58     <form class="row" method="POST" action="<?=$base_url('home/store/' . ($dataEdit['id'] ?? '')); ?>"
59         <?=$csrf_field(); ?>
60     <div class="col-10">
61         <input name="todoname" class="form-control mb-2 <?=$errors['todoname'] ? 'is-invalid' : '' ?>" value="<?=$dataEdit['todoname'] ? $dataEdit['todoname'] : '' ?>"
62         <div class="invalid-feedback"><?=$errors['todoname'] ?? '' ?></div>
63         <textarea name="description" class="form-control mb-2 <?=$errors['description'] ? 'is-invalid' : '' ?>" placeholder="Deskripsi"
64         <div class="invalid-feedback"><?=$errors['description'] ?? '' ?></div>
65     </div>
66
67     <div class="col-2">
68         <button class="btn btn-outline-primary" type="submit">Simpan
69     </div>
70 </form>
71 </div>
72 </main>
73
74 <!-- Optional JavaScript; choose one of the two! -->
75 <!-- Option 1: Bootstrap Bundle with Popper -->

```

```

59     <?=$empty($dataEdit) ? "Update Rencana" : "Tambahkan Rencana"; ?>
60 </button>
61 </div>
62 </form>
63 @var unset $data
64
65 <?php foreach ($data as $chunk): ?>
66 <div class="row mt-5">
67     <?php foreach ($chunk as $todo): ?>
68     <div class="col-3">
69         <div class="card" style="width: 18rem;">
70             <div class="card-body">
71                 <h5 class="card-title"><?=$todo['todoname']; ?></h5>
72                 <p class="card-text"><?=$todo['description']; ?></p>
73                 <a href="<?=$base_url('home/index/' . $todo['id']); ?>" class="btn btn-outline-info card-link" type="submit">Edit</a>
74                 <a href="<?=$base_url('home/delete/' . $todo['id']); ?>" class="btn btn-outline-danger card-link" type="submit">Hapus</a>
75             </div>
76         </div>
77     </div>
78     <?php endforeach; ?>
79 </div>
80 <?php endforeach; ?>
81 </div>
82 </main>
83
84 <!-- Optional JavaScript; choose one of the two! -->
85 <!-- Option 1: Bootstrap Bundle with Popper -->

```

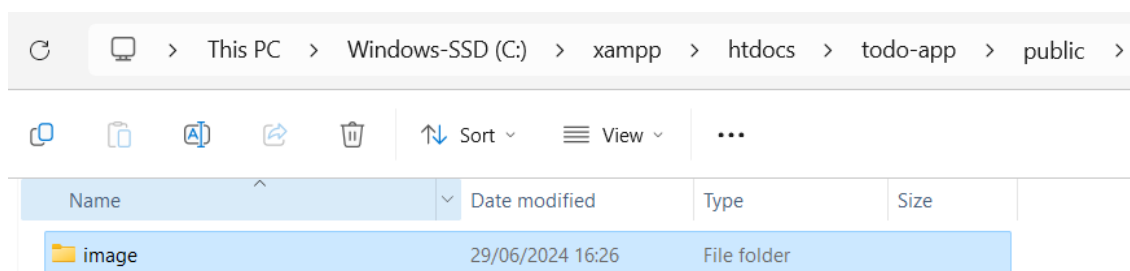

12. Lalu cari folder Controller dan ganti pada bagian file Home.php dengan code berikut:

```
Home.php todos.php .env TodoModel.php
todo-app > app > Controllers > Home.php > PHP Intelephense > Home > index
1  <?php
2  namespace App\Controllers;
3  use App\Models\TodoModel;
   0 references | 0 implementations
4  class Home extends BaseController
5  {
   0 references | 0 overrides
6      public function index($id='')
7      {
8          $dataEdit = array();
9          $todosModel = new TodoModel();
10         $data = $todosModel->orderBy('id', 'desc')->findAll();
11         $data = array_chunk($data, 4);
12         if (is_numeric($id)) {
13             $dataEdit = $todosModel->find($id);
14         }
15         return view('todos', array("data" => $data, 'dataEdit' => $dataEdit));
16     }
17
   0 references | 0 overrides
18     public function store($id='')
19     {
20         $data = $this->request->getPost();
21         $todosModel = new TodoModel();
22         if (!empty($id) && is_numeric($id)) {
23
24             $isDone = $todosModel->update($id, $data);
25         } else {
26
27             $isDone = $todosModel->insert($data);
28         }
29
30         if (!$isDone) {
31             $errors = $todosModel->errors();
32             session()->set('errorsMessages', $errors);
33         }
34
35         return redirect()->to('home');
36     }
37
   0 references | 0 overrides
38     public function delete($id)
39     {
40         $todosModel = new TodoModel();
41         $todosModel->delete($id);
42         session()->setFlashdata('successMessage', 'Deleted Successfully');
43
44         return redirect()->to('home');
45     }
46 }
```

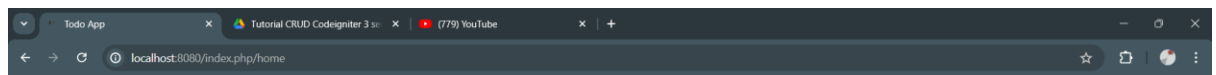
13. Lalu sekarang kita buat file baru di folder Models dengan nama TodoModel.php dengan isian berikut:

```
Home.php | todos.php | .env | TodoModel.php X
todo-app > app > Models > TodoModel.php > PHP Intelephense > TodoModel
1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  4 references | 0 implementations
8  class TodoModel extends Model
9  {
10     0 references
11     protected $table = 'todos';
12     0 references
13     protected $primaryKey = 'id';
14
15     0 references
16     protected $useAutoIncrement = true;
17
18     0 references
19     protected $returnType = 'array';
20
21     0 references
22     protected $allowedFields = ['todoname', 'description'];
23
24     0 references
25     protected $validationRules = [
26         'todoname' => 'required|min_length[3]',
27         'description' => 'permit_empty|min_length[3]',
28     ];
29
30     0 references
31     protected $validationMessages = [
32         'todoname' => [
33             'min_length' => 'Todo name must have minimum 3 characters.',
34         ],
35         'description' => [
36             'min_length' => 'Description must have minimum 3 characters.',
37         ],
38     ];
39 }
```

14. Nah setelahnya jangan lupa membuat folder image didalam folder public untuk logo yang ada di todos.php, sesuaikan nama gambarnya.



15. Selesai dan tampilan TodoApp nya akan seperti ini:



Kelola tugas Anda dengan lebih efektif menggunakan Todo App. Tambahkan, edit, dan hapus daftar tugas Anda dengan mudah. Jadikan hari Anda lebih produktif dan terorganisir!

[Tambahkan Rencana](#)

Kelola tugas Anda dengan lebih efektif menggunakan Todo App. Tambahkan, edit, dan hapus daftar tugas Anda dengan mudah. Jadikan hari Anda lebih produktif dan terorganisir!

[Tambahkan Rencana](#)

Hari Senin

Ada UAS Cloud Computing

[Edit](#)[Delete](#)

Jam 00.00 WIB

dsadsadasdasdsd

[Edit](#)[Delete](#)

BAB III PENUTUP

Kesimpulan

Pada praktikum ini, kami berhasil mengimplementasikan aplikasi Todo menggunakan framework CodeIgniter 4 dengan sukses. CodeIgniter 4 memberikan fondasi yang solid untuk membangun aplikasi web dengan pola desain MVC yang terstruktur. Langkah pertama kami adalah merancang basis data menggunakan fitur Migrations, yang memungkinkan kami untuk membuat tabel-tabel dengan mudah. Selanjutnya, kami mengatur logika aplikasi dalam Controller untuk mengelola interaksi antara Model yang bertanggung jawab atas manipulasi data dan View yang menampilkan informasi kepada pengguna. Kami menambahkan fitur-fitur penting seperti penambahan tugas baru, penampilan daftar tugas, kemampuan untuk menandai tugas sebagai selesai, dan opsi untuk menghapus tugas. Selain fungsionalitas dasar, kami juga mengimplementasikan perlindungan keamanan menggunakan fitur bawaan seperti CSRF protection untuk melindungi form dari serangan yang berpotensi. Pengalaman ini memperluas pemahaman kami tentang penggunaan framework untuk pengembangan aplikasi web dan memberikan landasan yang kokoh dalam praktik pemrograman web dengan menggunakan teknologi modern seperti CodeIgniter 4.