

**PANDUAN LENGKAP MENGENAI SISTEM PROSPEK  
SALES MENGGUNAKAN REGRESI LINIER  
BERGANDA**

**PANDUAN LENGKAP MENGENAI SISTEM PROSPEK  
SALES MENGGUNAKAN REGRESI LINIER  
BERGANDA**

**Imron Sumadireja**

Student

Penulis

ISBN

Editor

Penyunting

Dan lain-lain

## *Quotes*

## ***CONTRIBUTORS***

## ***CONTENTS IN BRIEF***

## **DAFTAR ISI**

## **DAFTAR GAMBAR**



## **DAFTAR TABEL**

## ***LISTINGS***

## **KATA PENGANTAR**

Buku ini merupakan panduan lengkap mengenai sistem prospek sales yang sedang dibangun menggunakan regresi linier berganda

## ***ACKNOWLEDGMENTS***

## ***ACRONYMS***

## ***SYMBOLS***

## ***INTRODUCTION***

## BAB I

### *HYPERTEXT PREPROCESSOR (PHP)*

#### 1.1 Sejarah PHP

PHP pertama kali dibuat oleh Rasmus Lerdorf, seorang programmer C. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data *form* dari *web*. Jadi, awal mula PHP digunakannya untuk menghitung jumlah pengunjung didalam *web*-nya.

Dengan alasan untuk meningkatkan performa, Rasmus Lerdorf membuat ulang kode program tersebut dalam bahasa C. Lerdorf menyebut kode program ini sebagai *Personal Home Page*. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan skrip PERL yang dibuatnya untuk membuat halaman *web*-nya menjadi dinamis. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Prerocessing / Form Interpreter*.

Dengan perilsan kode sumber ini menjadi *open source*, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah dapat mengakses *database* dan dapat terintegrasi dengan *Hypertext Markup Language* (HTML). Pada rilis ini *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP / FI secara signifikan. PHP versi 2.0 ini telah menarik banyak perhatian *programmer*, namun bahasa ini memiliki masalah dengan kestabilan yang kurang bisa



diandalkan. Hal ini lebih dikarenakan Rasmus Lerdorf hanya bekerja sendiri untuk mengembangkan PHP.

Pada saat itulah Zeev Suraski dan Andi Gutmans, ikut mengambil bagian dan membuat ulang *parsing engine* yang menjadi dasar dari PHP agar lebih stabil. Dengan dukungan dari banyak *programmer* lainnya, proyek PHP secara perlahan beralih dari proyek satu orang menjadi proyek masal yang lebih akrab kita kenal sebagai *open-source project*. PHP selanjutnya dikembangkan oleh The PHP Group yang merupakan kumpulan banyak *programmer* dari seluruh dunia.

Pada tahun 1998 tepatnya pada tanggal 6 Juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang *software*-nya.

PHP versi 4.0 keluar pada tanggal 22 Mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya *Zend Engine* yang dibuat oleh Zeev Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP *scripting engine*, yang lainnya adalah *build in HTTP session*, tidak lagi menggunakan *library* tambahan seperti pada PHP. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi *web*. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan *web server*.

PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, *Zend* merilis PHP 5.0. Dalam versi ini, inti dari *interpreter* PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Beberapa penambahan fitur

meliputi *PHP Data Objects* (PDO) untuk mengakses *database*, *closures*, *trait*, dan *namespaces*.

Versi lanjutan dari PHP, yakni PHP 6.x sebenarnya telah lama dikembangkan, bahkan sejak tahun 2005. Fokus pengembangan PHP 6 terutama dalam mendukung *unicode*, agar PHP bisa mendukung berbagai jenis karakter bahasa *non-latin*.

Namun dikarenakan beberapa alasan seperti kurangnya *programmer*, dan performa yang tidak memuaskan, pengembangan PHP 6 dihentikan dan fitur yang ada dimasukkan kedalam PHP 5.

Pada tahun 2014, sebuah proyek lanjutan PHP mulai mengemuka, yakni PHP 7 yang berkembang dari banyak eksperimen yang dinamakan *PHP Next Generation* (PHPNG), yang dikembangkan Dmitry Stogov, Xinchun Hui, dan Nikita Popov. Proyek ini menggunakan pendekatan modern agar PHP diproses lebih cepat seperti memakai teknik *just-in-time* (JIT) *compiler*. Proyek PHPNG bertujuan untuk menyusun ulang kode PHP untuk meningkatkan performa. Selain performa yang meningkat, terdapat beberapa fitur baru pada PHP 7, seperti *combined comparison operator* atau dikenal dengan *spaceship operator* “ $\Leftarrow$ ”, *anonymous classes*, dan dukungan yang lebih stabil untuk server 64-bit.

Beberapa fitur yang sudah usang (*deprecated*) juga dihapus, seperti penulisan PHP dengan *ASP style* `<% %>` dan tag `<script language=php></script>`. Kedua cara ini sudah tidak bisa digunakan lagi. Modul *mysql extension* juga dihapus karena sudah digantikan dengan *mysqli extension*. Penghapusan modul *mysql extension* ini sering menjadi masalah karena banyak kode program atau buku PHP lama yang masih menggunakannya.



## 1.2 Pengertian PHP

PHP adalah bahasa pemrograman *script* sisi server yang didesain untuk pengembangan *web*. Dimana PHP ini merupakan singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa pemrograman umum. Selain itu PHP juga digunakan bersamaan dengan bahasa pemrograman lainnya seperti bahasa pemrograman HTML, dan Javascript.

PHP disebut sebagai bahasa pemrograman *server-side* karena, hal ini berbeda dengan bahasa pemrograman *client-side* seperti Javascript yang diproses pada *web browser (client)*. PHP juga menjadi dasar dari aplikasi *Content Management System* (CMS) yang populer seperti Joomla, Drupal, dan Wordpress.

## 1.3 Fungsi PHP

Salah satu fungsi dari PHP ini dapat disisipkan pada dokumen HTML. Karena kemampuan inilah PHP juga sering disebut sebagai bahasa pemrograman *script* atau *scripting language*. Berikut ini merupakan sintaksis dasar pada PHP.

a. Pembatas

PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP. Apapun di luar pembatas tidak diproses oleh PHP. Pembatas paling umum adalah “<?php” untuk membuka dan “?>” untuk menutup kode PHP. Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML, dan Javascript.

b. Variabel

Variabel dalam PHP diawali dengan simbol dolar “\$”. Pada versi PHP 5 diperkenalkan jenis isyarat yang memungkinkan fungsi untuk memaksa mereka menjadi parameter objek dari *class* tertentu, *array*, atau fungsi. Namun, jenis petunjuk tidak dapat digunakan dengan jenis skalar seperti angka atau *string*. Contoh variabel dapat ditulis sebagai *\$nama\_variabel*.

c. Komentar

PHP memiliki 3 jenis sintaks sebagai komentar pada kode yaitu blok “/\* \*/”, komentar 2 baris “//”, serta tanda “#” digunakan untuk komentar 1 baris. Komentar bertujuan untuk meninggalkan catatan pada kode PHP dan tidak akan diterjemahkan ke program.

#### 1.4 Keunggulan & Kekurangan

a. Keunggulan bahasa pemrograman PHP, antara lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web server* yang mendukung PHP dapat ditemukan di mana-mana dari mulai *apache*, *IIS*, *Lightpad*, hingga *xitami* dengan konfigurasi yang relatif mudah.

3. Dalam sisi pengembangan lebih mudah, karena banyaknya *developer* yang siap membantu dalam pengembangan.
  4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
  5. PHP adalah bahasa *open source* yang digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta dapat menjalankan perintah-perintah sistem.
- b. Kekurangan bahasa pemrograman PHP, antara lain :
1. Tidak ideal jika untuk pengembangan skala besar.
  2. Tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya.
  3. Tidak dapat memisahkan antara tampilan dengan logik dengan baik.
  4. PHP mempunyai kelemahan *security* tertentu yang mana jika *programmer* tidak jeli dalam melakukan pemrograman dan kurang memperhatikan isu dan konfigurasi PHP.

Berikut adalah perbandingan antara PHP dengan bahasa *scripting server-side* lainnya seperti ASP – *Active Server Pages*, JSP – *Java Server Pages*, CFML – *Cold Markup Language*.

FITUR	PHP	ASP	JSP	CFML
Kurva belajar	pendek	Lebih lama dari PHP	Lebih lama dari PHP	Lebih lama dari PHP
Hosting web	Didukung oleh hampir semua server hosting	Membutuhkan server khusus	Cukup didukung	Membutuhkan server khusus
Sumber terbuka	Iya	Tidak	Iya	Baik komersial maupun open source
Dukungan layanan web	Dibangun	Menggunakan kerangka .NET	Menggunakan add on libraries	Dibangun
Integrasi dengan HTML	Mudah	Cukup rumit	Cukup rumit	Mudah
Dukungan MySQL	Asli	Membutuhkan driver pihak ketiga	Membutuhkan driver pihak ketiga	Versi saat ini memiliki dukungan asli. Versi yang lebih lama menggunakan ODBC
Mudah diperluas dengan bahasa lain	Iya	Tidak	Diperpanjang menggunakan kelas dan perpustakaan Java.	Iya

## 1.5 Sejarah OOP

Konsep *Object Oriented Programming* (OOP) pertama kali muncul di MIT (*Massachusetts Institute of Technology*) pada era 1960-an. Sekitar beberapa tahun kemudian antara 1962-1965, sebuah bahasa pemrograman yang mendasari konsep OOP diperkenalkan dengan nama bahasa pemrograman SIMULA 1, dikembangkan oleh Kristen Nygaard dan Ole-Johan yang merupakan warga negara Norwegia. Setelah itu pada tahun 1967 keluarlah SIMULA 67.

Bahasa SIMULA dengan OOP menginspirasi banyak pengembang, sehingga pada tahun 70-an tercipta bahasa SMALTALK, bahasa yang pertama kali disebut oleh dunia sebagai *object oriented programming*, dengan membawa fitur barunya yaitu *inheritance*.

Semenjak itu konsep OOP semakin dikenal, pada tahun 1980-an banyak bahasa pemrograman menggunakan konsep OOP, diperkenalkan kepada dunia, tetapi hanya empat yang cukup populer, diantaranya ADA (US Department of Defense), PROLOG (The Japanese “Fifth Generation Computer Project), Eiffle dan C++. Bahasa pemrograman ADA dan PROLOG dipercaya akan bersaing ketat sebagai bahasa pemrograman yang paling dominan.

Namun pada tahun 1980-an bahasa pemrograman C++ mematahkan kepercayaan tersebut. Bahasa pemrograman C++ menjadi bahasa pemrograman yang populer dan mendominasi hingga sekarang. Bahasa pemrograman C++ yang merupakan gabungan dari 2 konsep bahasa pemrograman, yakni C dan SIMULA.

Semenjak C++ terkenal, banyak sekali pengembang yang terinspirasi oleh C++ dan pada tahun 1990-an, bahasa pemrograman Java diperkenalkan yang mengaku terinspirasi oleh C++, dan tahun 2002 perusahaan Microsoft juga mengeluarkan bahasa turunan dari C++ yaitu C# (C-Sharp), disusul dengan VB.Net dengan fitur OOP yang merupakan penyempurnaan dari bahasa VB 0.6 yang tidak mendukung fitur OOP.

## 1.6 Pengertian OOP

*Object Oriented Programming* atau disingkat menjadi OOP adalah paradigma pemrograman dalam melakukan pemrograman yang berorientasi kepada objek, semua fungsi, data dan pengolahan data akan dibungkus dalam kelas-kelas dan objek-objek.

Masing-masing objek dapat memiliki sifat dan tugasnya. Pada paradigma ini, objek-objek tersebut dapat bekerja sendiri dan juga dapat saling bekerja sama dengan kemungkinan untuk saling berhubungan, seperti menerima, mengirim data kepada objek lainnya dan memproses data.

Paradigma OOP dapat dilihat sebagai interaksi dari objek yang saling berhubungan satu sama lain untuk melakukan tugasnya. Pengembangan program OOP dilakukan dengan cara mengikuti model yang telah ada di dalam dunia nyata.

OOP bertujuan untuk memberikan pola pikir dalam mengembangkan program, pola pikir tersebut dipercaya dapat memberikan kemudahan, fleksibilitas, kemudahan pembuatan, pengembangan program, dan perawatan program.

### 1.7 Jenis-jenis OOP pada bahasa pemrograman

Karena kemudahan yang diberikan oleh konsep OOP, banyak bahasa yang membawa dukungan fitur OOP, karena hal itu OOP menjadi memiliki dan dibagi menjadi 3 jenis, yaitu :

- a. Bahasa OOP murni, merupakan sebuah bahasa yang mengharuskan program ditulis hanya berupa objek saja. Contoh – Eifel, Smaltalk, Ruby, Jade, dan lain-lain.
- b. Bahasa OOP *hybrid*, merupakan bahasa yang dirancang untuk pemrograman objek dengan beberapa elemen prosedural.
- c. Bahasa OOP *hybrid* dalam *web*, salah seperti bahasa OOP *hybrid*, yang membedakan hanya konsep yang digunakan dalam pemrograman *web*.

### 1.8 Konsep *Object Oriented Programming*

- a. *Class*, adalah sebuah rancangan untuk mendefinisikan karakter dan perilaku dari objek, yang merupakan kumpulan atas definisi dan fungsi-fungsi dalam suatu unit, untuk suatu tujuan tertentu.
- b. *Object*, adalah dasar dari modularitas dan struktur pada OOP, dan merupakan representasi dari *class*, objek akan memiliki sifat dan perilaku dari *class* yang digunakan.
- c. *Encapsulation*, adalah konsep dalam implementasi untuk membungkus data dan fungsi menjadi satu entitas, dan membatasi akses dari luar *class*.
- d. *Inheritance*, adalah konsep pewarisan *class*. *Class* juga dapat menurun dan memiliki apa yang dimiliki oleh *class* lainnya.
- e. *Abstraction*, adalah konsep untuk mendesain sebuah objek, teknik dalam menyembunyikan detail suatu proses dalam objek tersebut, dengan tujuan untuk memfokuskan pengguna pada fungsi inti objek.



- f. *Polymorphism*, adalah kemampuan dalam menyampaikan pesan tertentu keluar dari hierarki objeknya, dimana objek yang berbeda memberikan tanggapan atau respon terhadap pesan yang sama sesuai dengan sifat masing-masing objek.

#### 1.9 Keunggulan & Kekurangan OOP

- a. Keunggulan OOP, diantaranya :
  - 1. OOP menyediakan struktur modular yang jelas untuk program sehingga OOP sangat bagus digunakan untuk mendefinisikan tipe data abstrak dimana detail implementasinya tersembunyi.
  - 2. OOP akan mempermudah dalam maintaining dan memodifikasi kode yang sudah ada. Objek yang baru dapat dibuat tanpa mengubah kode yang sudah ada.
  - 3. OOP menyediakan *framework* untuk *library* kode dimana komponen *software* yang tersedia dapat dengan mudah diadaptasi dan dimodifikasi oleh programmer. Hal ini sangat berguna untuk mengembangkan *Graphical User Interfaces* (GUI).
- b. Kekurangan OOP, diantaranya :
  - 1. Tidak memperbolehkan implementasi yang kuat pada *reuse*.
  - 2. *Property software* tidak terikat dalam satu unit fungsional sehingga harus *crosscut* di antara komponennya.
  - 3. *Crosscut* tersebut mengakibatkan sulitnya pengembangan dan pemeliharaan.

#### 1.10 Sejarah CodeIgniter

Kelahiran codeignier adalah bermula dari kegalauan Rick Ellis atas banyaknya kode PHP yang harus ditulis ketika

membangun salah satu CMS kesayangannya, *expression engine*. Rick Ellis ingin mempermudah penulisan kode-kode program PHP dan membuatnya lebih singkat dan cepat. Akhirnya Rick Ellis mengambil inisiatif dengan membuat sendiri kode singkat / *shortcode* dari fungsi-fungsi yang ada di PHP.

Pembuatan *shortcode* tersebut membuatnya mampu membangun *expression engine* dengan sangat bagus, efisien dan cepat. Selain itu, performanya juga sangat bagus.

Setelah tidak berapa lama, Rick Ellis melalui situsnya ellislab(dot)com membagikan *shortcode* yang dibuat sendiri itu untuk digunakan oleh *developer* lainnya. Tujuannya yaitu membantu *developer* lain dalam menangani masalah dalam *framework* yang terkenal. Banyak *developer* turun tangan membantu perkembangan CodeIgniter dan puncaknya, CI menjadi *framework* terpopuler tahun 2006.

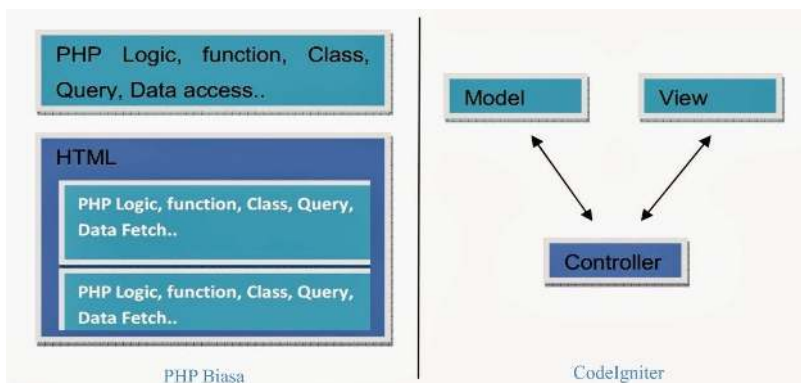


### 1.11 Pengertian CodeIgniter

CodeIgniter (CI) merupakan sebuah *web application framework* yang bersifat *open source* dimana digunakan untuk membangun aplikasi php dinamis. *Framework* itu sendiri merupakan abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh

kode yang dibuat *user*, sehingga dapat menyediakan perangkat lunak untuk aplikasi tertentu.

Codeigniter menjadi sebuah *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP yang dapat mempercepat pengembangan untuk membuat sebuah aplikasi *web*. Selain ringan dan cepat, codeigniter juga memiliki dokumentasi yang lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih codeigniter sebagai *framework* pilihannya. Berikut merupakan gambar perbandingan PHP biasa dengan Codeigniter.



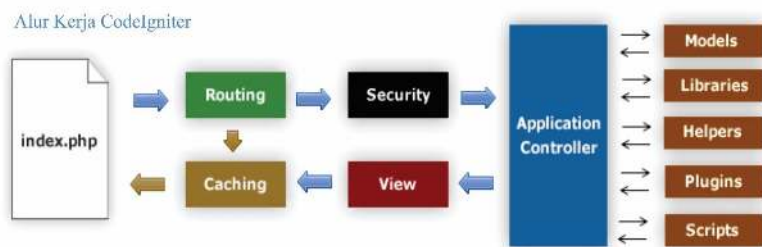
MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi, diantaranya :

- a. Model, dapat berhubungan dengan *database* seperti *insert, update, delete*. Menangani validasi dari bagian

controller, namun model tidak berhubungan langsung dengan *view*.

- b. Controller, merupakan bagian yang mengatur hubungan antara model dan *view*, controller berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.
- c. *View*, merupakan bagian yang menangani *presentation*. Pada suatu aplikasi *web* bagian ini biasanya berupa file template HTML, yang diatur oleh controller. *View* berfungsi untuk menerima dan mempresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.

Alur kerja *framework* Codeigniter dapat digambarkan sebagai berikut :



- a. Index.php: berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- b. Router: router akan memeriksa HTTP *request* untuk menentukan hal apa yang harus dilakukan oleh program.
- c. *Cache* File: apabila dalam program sudah terdapat “*cache* file” maka file tersebut akan langsung dikirim ke browser. File *cache* inilah yang dapat membuat sebuah *website* dapat dibuka dengan lebih cepat. *Cache* file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.

- d. *Security*: sebelum *file controller* di *load* keseluruhan, HTTP *request* dan data yang disubmit oleh *user* akan disaring terlebih dahulu melalui fasilitas *securtiy* yang dimiliki oleh codeigniter.
- e. *Controller*: controller akan membuka file model, *core libraries*, *helper* dan semua *resources* yang dibutuhkan dalam program tersebut.
- f. *View*: hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam *view* file dan mengirimkannya ke browser supaya dapat dilihat. Apabila file *view* sudah ada yang di “cache” maka file *view* baru yang belum ter-*cache* akan *update* file *view* yang sudah ada.

#### 1.12 Fungsi CodeIgniter

- a. Mempercepat dan mempermudah kita dalam pembuatan *website*.
- b. Menghasilkan struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file phpnya.
- c. Memberikan standar *coding* sehingga memudahkan kita atau orang lain untuk mempelajari kembali sistem aplikasi yang dibangun.

#### 1.13 Keunggulan & Kekurangan CodeIgniter

- a. Keunggulan Codeigniter, diantaranya :
  - 1. Performa sangat cepat, salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scratch*, tetapi codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan *framework* yang paling cepat dibanding *framework* yang lainnya.
  - 2. Konfigurasi yang sangat minim (*nearly zero configuration*), tentu saja untuk menyesuaikan

dengan *database* dan keleluasaan *routing* tetapi diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan codeigniter dengan setingan standar, anda hanya perlu mengubah sedikit saja file pada folder config.

3. Dokumentasi yang lengkap, setiap paket instalasi codeigniter sudah disertai *user guide* yang bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.
- b. Kekurangan Codeigniter, diantaranya :
1. Codeigniter dikembangkan oleh Ellis lab dan bukan oleh suatu komunitas, yang menyebabkan *update code engine*-nya tidak secepat *framework* lain.
  2. Tidak ditujukan untuk pembuatan *web* dengan skala besar (*enterprise*) walaupun tersedia banyak *library*.
  3. Masih banyak kelonggaran dalam hal *coding*, misalnya bebas dalam menambah file.

## BAB II

### INSTALASI APLIKASI YANG AKAN DIGUNAKAN

#### 1. Visual Studio Code

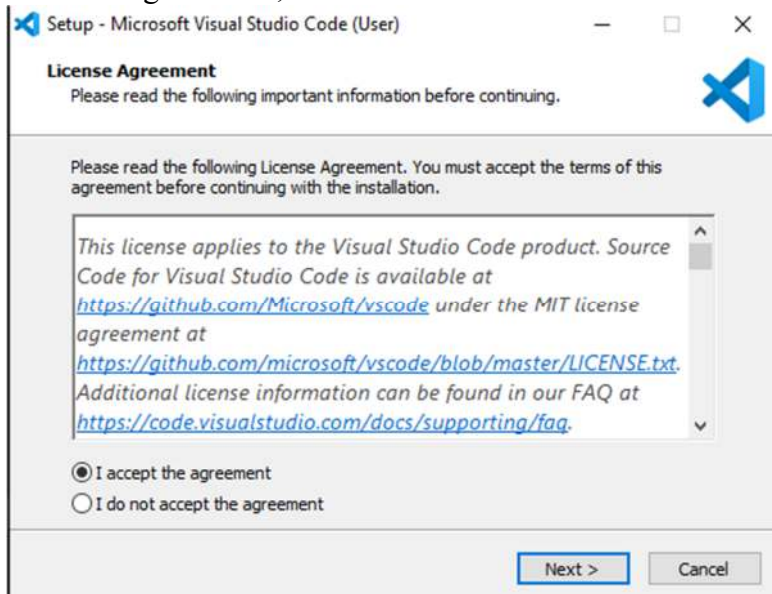
Visual studio code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via *marketplace* Visual Studio Code (seperti C++, C#, PHP, Python, Go, Java, dst).

Berikut ini merupakan langkah-langkah instalasi visual studio code pada sistem operasi windows.

- a. Pertama, *download* terlebih dahulu file installer visual studio code melalui situs resminya pada link berikut ini <https://code.visualstudio.com/download>

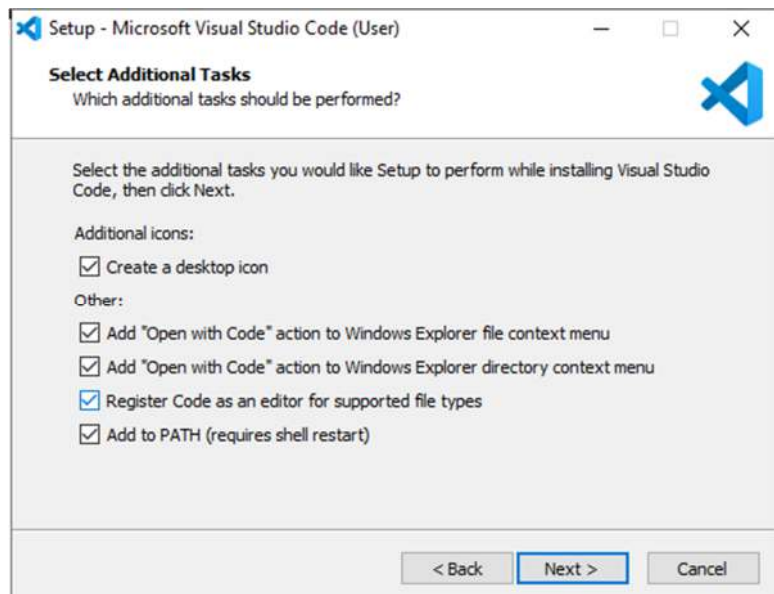


- b. Terdapat 2 pilihan untuk sistem operasi windows, diantaranya ada yang 64 bit dan 32 bit. Pilihlah installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Jika, sudah berhasil di *download*, lanjut pada proses instalasi, pertama klik pada file installer vs code.
- d. Pilih “I accept the agreement” untuk menyetujui “License agreement”, kemudian klik next.

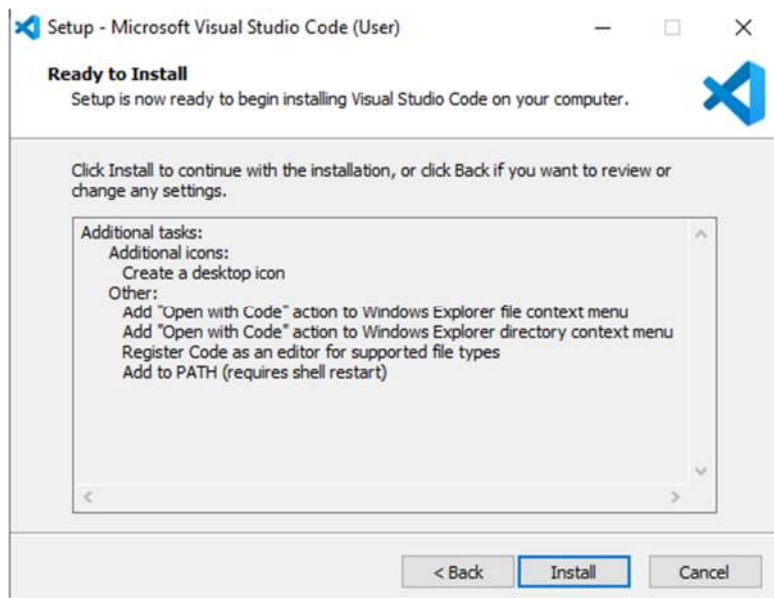


- e. Pada bagian select additional tasks, centang semuanya kemudian klik next.

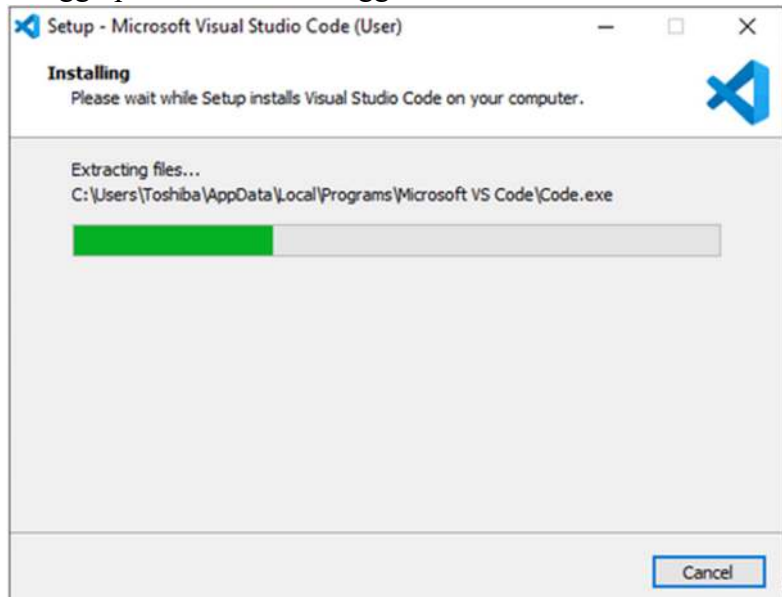




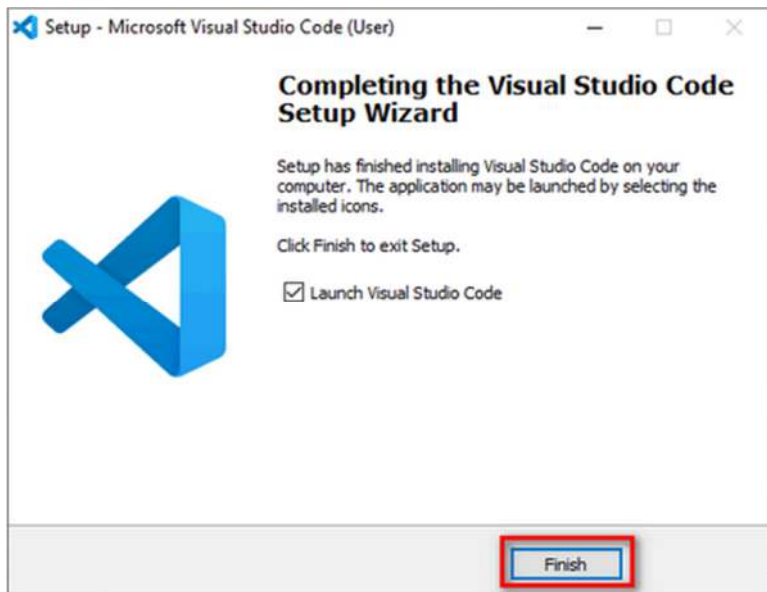
- f. Kemudian klik install untuk melanjutkan proses instalasi.



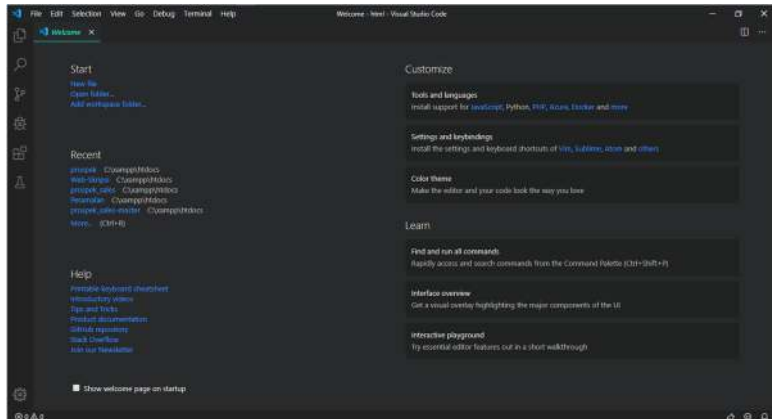
- g. Tunggu proses instalasi hingga selesai.



- h. Setelah selesai klik Finish.



- i. Jika berhasil, maka tampilan utamanya akan seperti berikut.



## 2. Xampp

Xampp adalah sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL / MariaDB, PHP, dan PERL. Sementara huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi yang berbeda.

Jika dijabarkan secara gamblang, masing-masing huruf yang adad di dalam nama XAMPP memiliki arti sebagai berikut.

**X = *Cross Platform***, merupakan kode penanda untuk *software cross platform* atau yang bisa berjalan di banyak sistem operasi.

**A = Apache**, apache adalah aplikasi *web server* yang bersifat gratis dan bisa dikembangkan oleh banyak orang (*open source*).

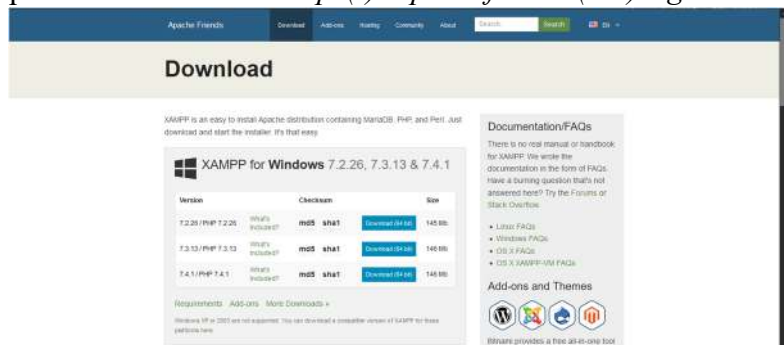
**M = MySQL / MariaDB**, MySQL atau MariaDB merupakan aplikasi *database server* yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui *database*.

**P = PHP**, huruf “P” yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat *website dinamis*.

**P = Perl**, untuk huruf “P” selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi, sehingga sangat fleksibel dan banyak digunakan.

Berikut merupakan langkah-langkah untuk melakukan instalasi xampp pada sistem operasi windows.

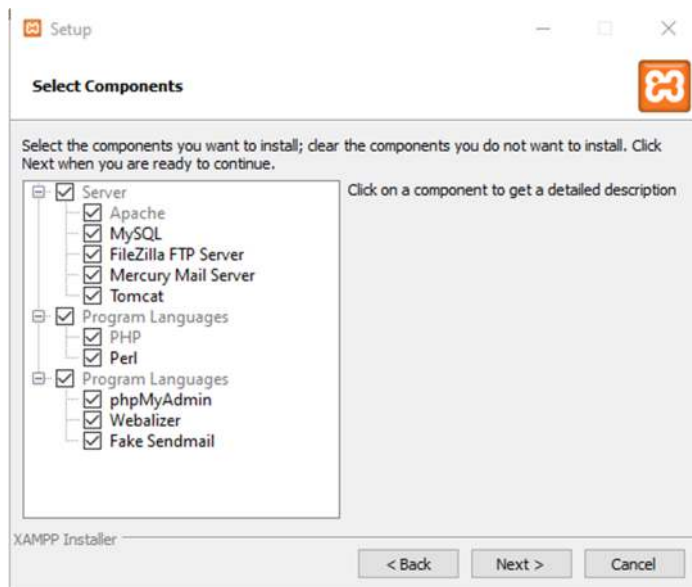
- a. Pertama, *download* terlebih dahulu file installer xampp pada link berikut ini. <https://apachefriends.org>



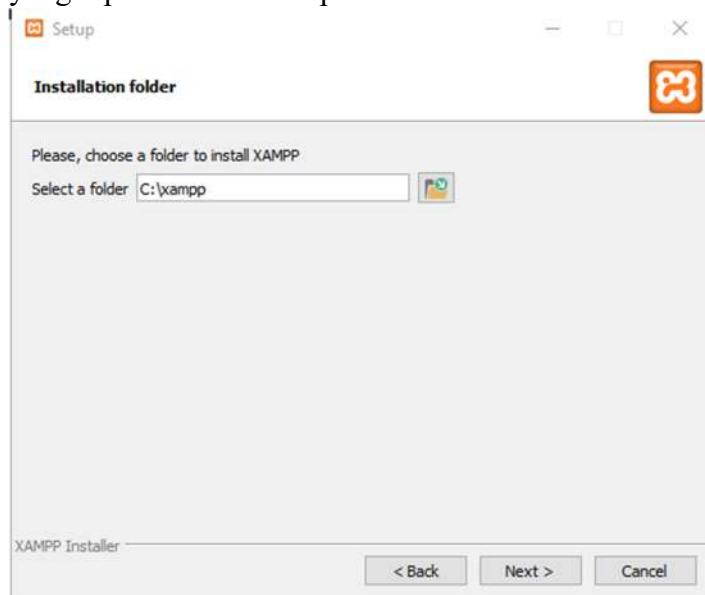
- b. Pilih file installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Klik file xampp yang sudah berhasil di *download*, maka akan muncul tampilan seperti ini.



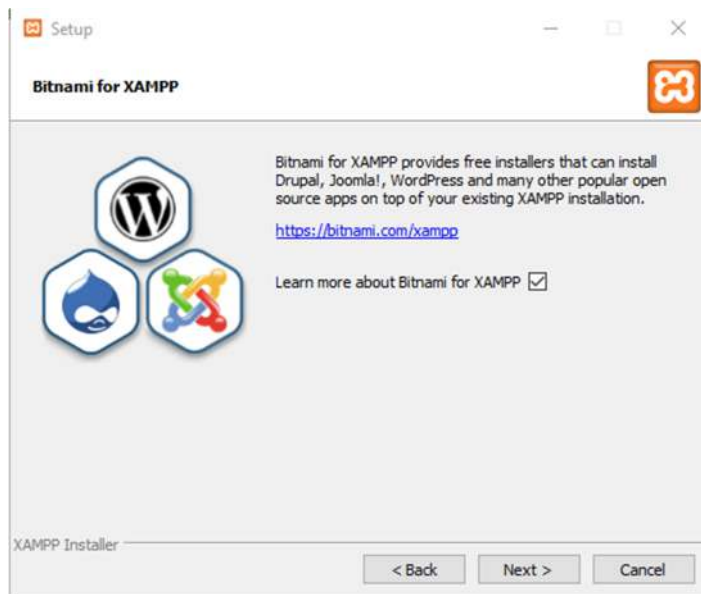
- d. Kemudian, pilih next untuk melanjutkan pada proses instalasi.



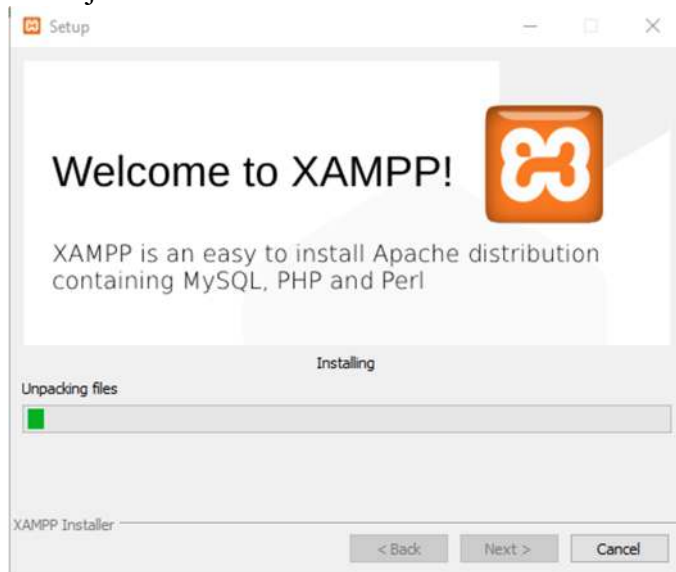
- e. Pada tampilan *select component*, centang semua pilihannya agar dapat menjalankan semua program yang dipilih. Kemudian pilih *next*.



- f. Pada, *installation folder* secara *default* sistem akan menyimpannya pada direktori C. Jika sudah klik *next* untuk melanjutkan proses instalasi.



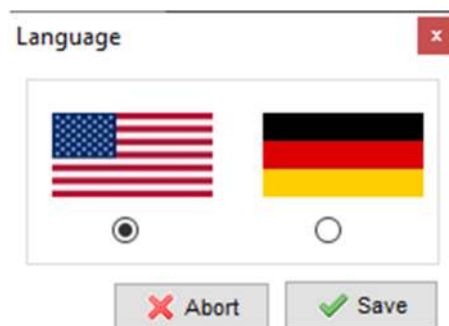
- g. Pada tampilan ini langsung saja klik *next* untuk melanjutkan.



- h. Kemudian tunggu hingga proses instalasi selesai.

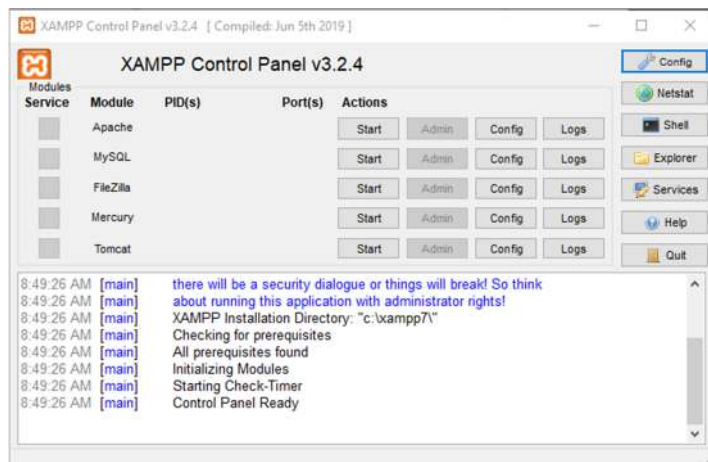


- i. Jika telah selesai, selanjutnya klik *Finish* untuk membuka *control panel*.



- j. Pilih bahasa yang akan digunakan pada xampp. Lalu klik *save*.





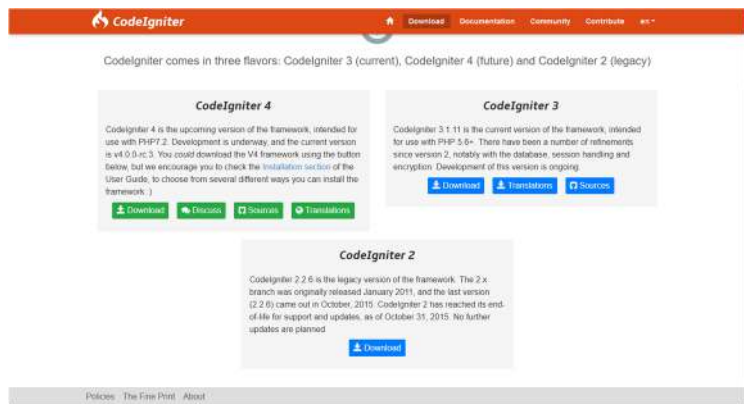
- k. Jika berhasil, maka tampilan utama *control panel* dari xampp seperti berikut.

### 3. Codeigniter

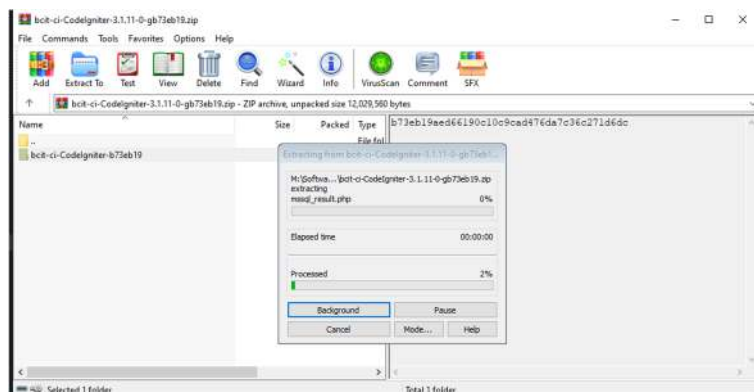
Codeigniter (CI) merupakan sebuah *web application framework* yang bersifat *open source* dimana digunakan untuk membangun aplikasi php dinamis. *Framework* itu sendiri merupakan abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh kode yang dibuat *user*, sehingga dapat menyediakan perangkat lunak untuk aplikasi tertentu.

Berikut ini merupakan tahapan-tahapan untuk instalasi codeigniter pada sistem operasi windows.

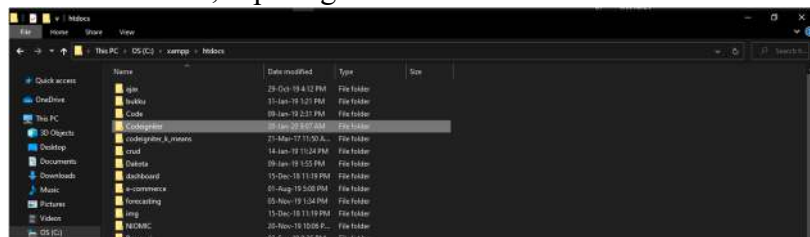
- a. Pertama, *download* terlebih dahulu codeigniter pada *website* resminya pada link berikut.  
*<https://codeigniter.com/download>*.



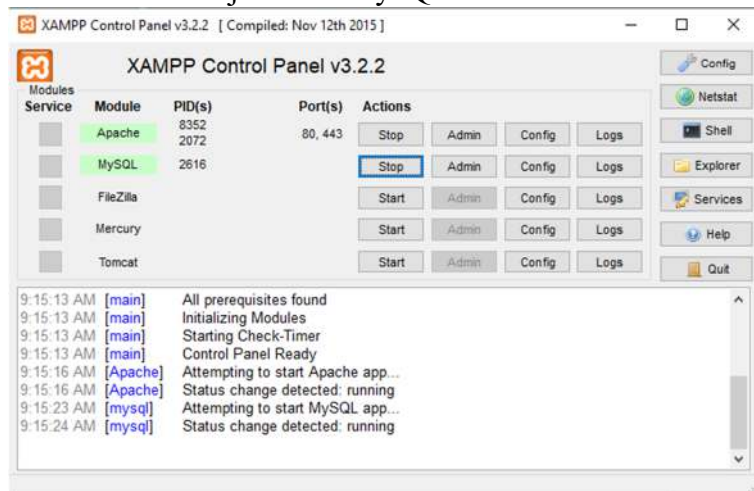
- b. Setelah berhasil di *download*, selanjutnya ekstrak file tersebut.



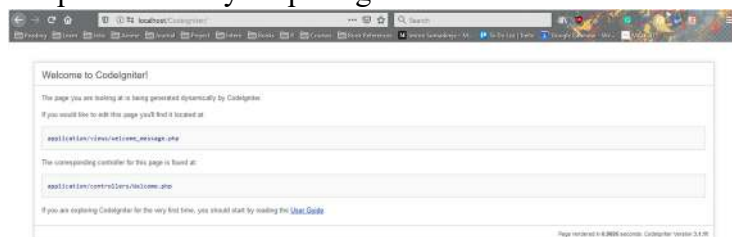
- c. Kemudian, *copy* file yang telah di ekstrak ke dalam direktori httdocs, seperti gambar berikut.



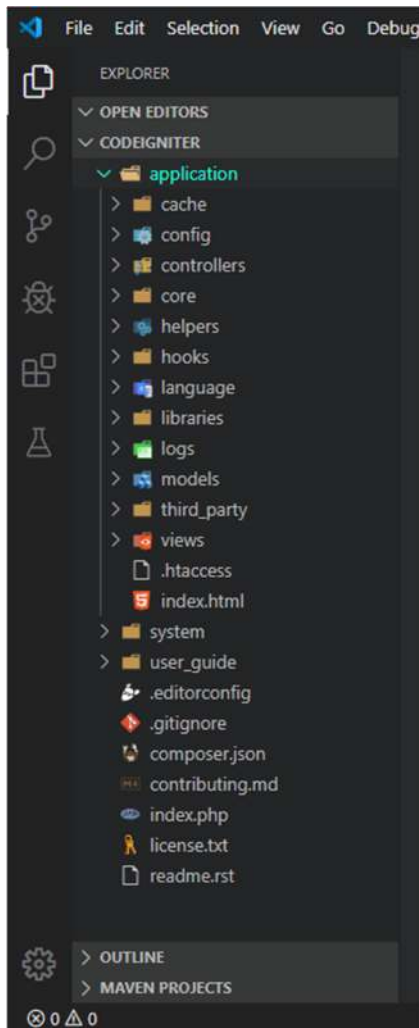
- d. Untuk menjalankannya, kita menggunakan xampp, untuk itu silakan jalankan terlebih dahulu aplikasi xampp-nya seperti berikut.
- e. Buka aplikasi xamppnya, kemudian klik tombol start pada kolom actions baris pertama untuk menjalankan apache, dan klik tombol start pada kolom actions baris kedua untuk menjalankan MySQL.



- f. Setelah xamppnya berhasil dijalankan, proses selanjutnya tinggal memanggil folder yang telah disimpan pada direktori htdocs di browser. Maka tampilan utamanya seperti gambar berikut.



- g. Berikut ini merupakan struktur folder yang terdapat pada codeigniter, diantaranya sebagai berikut.

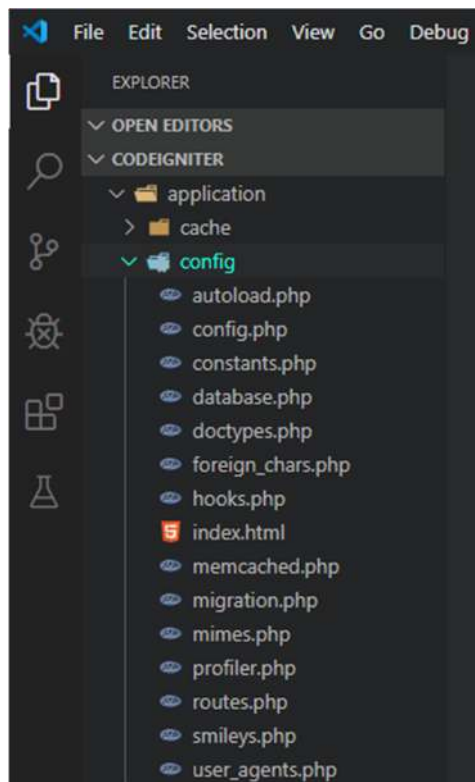


- a) *Application*, merupakan folder yang pada dasarnya menyimpan aplikasi yang sedang kita buat.
- b) *Cache*, merupakan folder yang menyimpan semua cache yang dibuat oleh cache library.

- c) *Config*, merupakan folder yang menyimpan informasi mengenai konfigurasi aplikasi seperti autoload, database, routes, dan lainnya.
- d) *Controller*, merupakan folder yang menyimpan controller-controller aplikasi yang dapat digunakan untuk menyusun aktivitas program.
- e) *Core*, merupakan folder untuk memperluas *class* inti codeigniter.
- f) *Helpers*, merupakan folder untuk menyimpan helpers.
- g) *Hooks*, merupakan folder untuk menyimpan hooks untuk mengubah alur fungsi dari core codeigniter.
- h) *Language*, merupakan folder untuk menyimpan bahasa-bahasa yang akan digunakan.
- i) *Libraries*, merupakan folder untuk menyimpan library.
- j) *Logs*, merupakan folder untuk menyimpan semua error log apabila error log diaktifkan.
- k) *Models*, merupakan folder untuk menyimpan models yang akan didefinisikan tabel dari database yang dapat kita gunakan oleh controller yang kita buat untuk mengakses database.
- l) *Third\_party*, merupakan folder untuk menyimpan fungsi-fungsi tambahan dalam cara kerja codeigniter.
- m) *Views*, merupakan folder untuk menyimpan tampilan dari aplikasi yang kita buat.
- n) *System*, merupakan folder untuk menyimpan sistem inti dari codeigniter.

#### 4. Konfigurasi dasar pada codeigniter

Dalam memulai codeigniter, ada beberapa konfigurasi dasar yang perlu kita ketahui, diantaranya autoload.php, config.php dan database.php. Semua konfigurasi pada codeigniter, terletak pada satu tempat yakni di dalam folder *application/config*.



- a. Autoload.php, file ini digunakan untuk mengatur fungsi-fungsi yang akan dimuat otomatis di awal ketika program dijalankan. Untuk melakukan konfigurasi pada file autoload.php, silakan buka file-nya seperti gambar berikut.

```
autoload.php X
application > config > autoload.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 /*
5  * AUTO-LOADING
6  * This file specifies which systems should be loaded by default.
7  *
8  * In order to keep the framework as light-weight as possible only the
9  * absolute minimum resources are loaded by default. For example,
10  * the database is not connected to automatically since no assumption
11  * is made regarding whether you intend to use it. This file lets
12  * you globally define which systems you would like loaded with every
13  * request.
14  *
15  * Instructions
16  *
17  * There are the things you can load automatically:
18  *
19  * 1. Packages
20  * 2. Libraries
21  * 3. Drivers
22  * 4. Helper files
23  * 5. Custom config files
24  */
```

- b. Kemudian temukan kode berikut.

```
61 $autoload['libraries'] = array();
```

- c. Ubah kode tersebut menjadi seperti berikut.

```
61 $autoload['libraries'] = array('database');
```

Kode tersebut dapat diartikan kita dapat meload library 'database' secara otomatis.

- d. Selanjutnya, temukan kode berikut, dan tambahkan 'url' didalamnya.

```
92 $autoload['helper'] = array('url');
93
```

Kode tersebut dapat diartikan kita dapat meload helper "url" secara otomatis.

- e. Config.php, pada file ini terdapat beberapa konfigurasi yang secara standar sudah terkonfigurasi, namun terdapat beberapa konfigurasi yang perlu diperhatikan, untuk konfigurasi dasar, cukup mengetahui konfigurasi base\_url.

```
autoloader.php  config.php X
application > config > config.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 /*
5 .....
6 / Base Site URL
7 .....
8 /
9 / URL to your CodeIgniter root. Typically this will be your base URL,
10 / WITH a trailing slash:
11 /
12 / http://example.com/
13 /
14 / WARNING: You MUST set this value!
15 /
16 / If it is not set, then CodeIgniter will try guess the protocol and path
17 / your installation, but due to security concerns the hostname will be set
18 / to a generic 'example.com' if available, or localhost otherwise.
19 / The auto-detection mechanism exists only for convenience during
20 / development and MUST NOT be used in production!
21 /
22 / If you need to allow multiple domains, remember that this file is still
23 / a script and you can easily do that on your own.
24 /
25 /
26 $config['base_url'] = '';
27
```

f. Kemudian temukan kode berikut.

```
26 $config['base_url'] = '';
```

g. Kemudian tambahkan kode tersebut, menjadi nama folder yang disimpan pada htdocs.

```
26 $config['base_url'] = 'http://localhost/Codeigniter/';
```

h. Database.php, file ini digunakan untuk melakukan konfigurasi yang berkaitan dengan konfigurasi database dari website yang akan dibuat. Adapun konfigurasi yang perlu diperhatikan, diantaranya: hostname, username, password dan database. Buka file database.php pada teks editor seperti gambar berikut.

```
autoloader.php  config.php  database.php X
application > config > database.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 /*
5 .....
6 / DATABASE CONNECTIVITY SETTINGS
7 .....
8 / This file will contain the settings needed to access your database.
9 /
10 / For complete instructions please consult the 'Database Connection'
11 / page of the User Guide.
12 /
13 / .....
14 / EXPLANATION OF VARIABLES
15 / .....
16 /
17 / ['dsn'] The full DSN string describe a connection to the database.
18 / ['hostname'] The hostname of your database server.
19 / ['username'] The username used to connect to the database
20 / ['password'] The password used to connect to the database
21 / ['database'] The name of the database you want to connect to
22 / ['driver'] The database driver, e.g.: mysqli.
23 /
24 / Currently supported:
25 /   - mysqli
26 /   - mysql
27 /   - mssql
28 /   - oci8
29 /   - pdo
30 /   - sqlite
31 /   - sqlsrv
32 /   - sqlsybase
33 /   - sqlsybase_ats
34 /   - sqlsybase_ats_10
35 /   - sqlsybase_ats_11
36 /   - sqlsybase_ats_12
37 /   - sqlsybase_ats_13
38 /   - sqlsybase_ats_14
39 /   - sqlsybase_ats_15
40 /   - sqlsybase_ats_16
41 /   - sqlsybase_ats_17
42 /   - sqlsybase_ats_18
43 /   - sqlsybase_ats_19
44 /   - sqlsybase_ats_20
45 /   - sqlsybase_ats_21
46 /   - sqlsybase_ats_22
47 /   - sqlsybase_ats_23
48 /   - sqlsybase_ats_24
49 /   - sqlsybase_ats_25
50 /   - sqlsybase_ats_26
51 /   - sqlsybase_ats_27
52 /   - sqlsybase_ats_28
53 /   - sqlsybase_ats_29
54 /   - sqlsybase_ats_30
55 /   - sqlsybase_ats_31
56 /   - sqlsybase_ats_32
57 /   - sqlsybase_ats_33
58 /   - sqlsybase_ats_34
59 /   - sqlsybase_ats_35
60 /   - sqlsybase_ats_36
61 /   - sqlsybase_ats_37
62 /   - sqlsybase_ats_38
63 /   - sqlsybase_ats_39
64 /   - sqlsybase_ats_40
65 /   - sqlsybase_ats_41
66 /   - sqlsybase_ats_42
67 /   - sqlsybase_ats_43
68 /   - sqlsybase_ats_44
69 /   - sqlsybase_ats_45
70 /   - sqlsybase_ats_46
71 /   - sqlsybase_ats_47
72 /   - sqlsybase_ats_48
73 /   - sqlsybase_ats_49
74 /   - sqlsybase_ats_50
75 /   - sqlsybase_ats_51
76 /   - sqlsybase_ats_52
77 /   - sqlsybase_ats_53
78 /   - sqlsybase_ats_54
79 /   - sqlsybase_ats_55
80 /   - sqlsybase_ats_56
81 /   - sqlsybase_ats_57
82 /   - sqlsybase_ats_58
83 /   - sqlsybase_ats_59
84 /   - sqlsybase_ats_60
85 /   - sqlsybase_ats_61
86 /   - sqlsybase_ats_62
87 /   - sqlsybase_ats_63
88 /   - sqlsybase_ats_64
89 /   - sqlsybase_ats_65
90 /   - sqlsybase_ats_66
91 /   - sqlsybase_ats_67
92 /   - sqlsybase_ats_68
93 /   - sqlsybase_ats_69
94 /   - sqlsybase_ats_70
95 /   - sqlsybase_ats_71
96 /   - sqlsybase_ats_72
97 /   - sqlsybase_ats_73
98 /   - sqlsybase_ats_74
99 /   - sqlsybase_ats_75
100 /   - sqlsybase_ats_76
101 /   - sqlsybase_ats_77
102 /   - sqlsybase_ats_78
103 /   - sqlsybase_ats_79
104 /   - sqlsybase_ats_80
105 /   - sqlsybase_ats_81
106 /   - sqlsybase_ats_82
107 /   - sqlsybase_ats_83
108 /   - sqlsybase_ats_84
109 /   - sqlsybase_ats_85
110 /   - sqlsybase_ats_86
111 /   - sqlsybase_ats_87
112 /   - sqlsybase_ats_88
113 /   - sqlsybase_ats_89
114 /   - sqlsybase_ats_90
115 /   - sqlsybase_ats_91
116 /   - sqlsybase_ats_92
117 /   - sqlsybase_ats_93
118 /   - sqlsybase_ats_94
119 /   - sqlsybase_ats_95
120 /   - sqlsybase_ats_96
121 /   - sqlsybase_ats_97
122 /   - sqlsybase_ats_98
123 /   - sqlsybase_ats_99
124 /   - sqlsybase_ats_100
125 /   - sqlsybase_ats_101
126 /   - sqlsybase_ats_102
127 /   - sqlsybase_ats_103
128 /   - sqlsybase_ats_104
129 /   - sqlsybase_ats_105
130 /   - sqlsybase_ats_106
131 /   - sqlsybase_ats_107
132 /   - sqlsybase_ats_108
133 /   - sqlsybase_ats_109
134 /   - sqlsybase_ats_110
135 /   - sqlsybase_ats_111
136 /   - sqlsybase_ats_112
137 /   - sqlsybase_ats_113
138 /   - sqlsybase_ats_114
139 /   - sqlsybase_ats_115
140 /   - sqlsybase_ats_116
141 /   - sqlsybase_ats_117
142 /   - sqlsybase_ats_118
143 /   - sqlsybase_ats_119
144 /   - sqlsybase_ats_120
145 /   - sqlsybase_ats_121
146 /   - sqlsybase_ats_122
147 /   - sqlsybase_ats_123
148 /   - sqlsybase_ats_124
149 /   - sqlsybase_ats_125
150 /   - sqlsybase_ats_126
151 /   - sqlsybase_ats_127
152 /   - sqlsybase_ats_128
153 /   - sqlsybase_ats_129
154 /   - sqlsybase_ats_130
155 /   - sqlsybase_ats_131
156 /   - sqlsybase_ats_132
157 /   - sqlsybase_ats_133
158 /   - sqlsybase_ats_134
159 /   - sqlsybase_ats_135
160 /   - sqlsybase_ats_136
161 /   - sqlsybase_ats_137
162 /   - sqlsybase_ats_138
163 /   - sqlsybase_ats_139
164 /   - sqlsybase_ats_140
165 /   - sqlsybase_ats_141
166 /   - sqlsybase_ats_142
167 /   - sqlsybase_ats_143
168 /   - sqlsybase_ats_144
169 /   - sqlsybase_ats_145
170 /   - sqlsybase_ats_146
171 /   - sqlsybase_ats_147
172 /   - sqlsybase_ats_148
173 /   - sqlsybase_ats_149
174 /   - sqlsybase_ats_150
175 /   - sqlsybase_ats_151
176 /   - sqlsybase_ats_152
177 /   - sqlsybase_ats_153
178 /   - sqlsybase_ats_154
179 /   - sqlsybase_ats_155
180 /   - sqlsybase_ats_156
181 /   - sqlsybase_ats_157
182 /   - sqlsybase_ats_158
183 /   - sqlsybase_ats_159
184 /   - sqlsybase_ats_160
185 /   - sqlsybase_ats_161
186 /   - sqlsybase_ats_162
187 /   - sqlsybase_ats_163
188 /   - sqlsybase_ats_164
189 /   - sqlsybase_ats_165
190 /   - sqlsybase_ats_166
191 /   - sqlsybase_ats_167
192 /   - sqlsybase_ats_168
193 /   - sqlsybase_ats_169
194 /   - sqlsybase_ats_170
195 /   - sqlsybase_ats_171
196 /   - sqlsybase_ats_172
197 /   - sqlsybase_ats_173
198 /   - sqlsybase_ats_174
199 /   - sqlsybase_ats_175
200 /   - sqlsybase_ats_176
201 /   - sqlsybase_ats_177
202 /   - sqlsybase_ats_178
203 /   - sqlsybase_ats_179
204 /   - sqlsybase_ats_180
205 /   - sqlsybase_ats_181
206 /   - sqlsybase_ats_182
207 /   - sqlsybase_ats_183
208 /   - sqlsybase_ats_184
209 /   - sqlsybase_ats_185
210 /   - sqlsybase_ats_186
211 /   - sqlsybase_ats_187
212 /   - sqlsybase_ats_188
213 /   - sqlsybase_ats_189
214 /   - sqlsybase_ats_190
215 /   - sqlsybase_ats_191
216 /   - sqlsybase_ats_192
217 /   - sqlsybase_ats_193
218 /   - sqlsybase_ats_194
219 /   - sqlsybase_ats_195
220 /   - sqlsybase_ats_196
221 /   - sqlsybase_ats_197
222 /   - sqlsybase_ats_198
223 /   - sqlsybase_ats_199
224 /   - sqlsybase_ats_200
225 /   - sqlsybase_ats_201
226 /   - sqlsybase_ats_202
227 /   - sqlsybase_ats_203
228 /   - sqlsybase_ats_204
229 /   - sqlsybase_ats_205
230 /   - sqlsybase_ats_206
231 /   - sqlsybase_ats_207
232 /   - sqlsybase_ats_208
233 /   - sqlsybase_ats_209
234 /   - sqlsybase_ats_210
235 /   - sqlsybase_ats_211
236 /   - sqlsybase_ats_212
237 /   - sqlsybase_ats_213
238 /   - sqlsybase_ats_214
239 /   - sqlsybase_ats_215
240 /   - sqlsybase_ats_216
241 /   - sqlsybase_ats_217
242 /   - sqlsybase_ats_218
243 /   - sqlsybase_ats_219
244 /   - sqlsybase_ats_220
245 /   - sqlsybase_ats_221
246 /   - sqlsybase_ats_222
247 /   - sqlsybase_ats_223
248 /   - sqlsybase_ats_224
249 /   - sqlsybase_ats_225
250 /   - sqlsybase_ats_226
251 /   - sqlsybase_ats_227
252 /   - sqlsybase_ats_228
253 /   - sqlsybase_ats_229
254 /   - sqlsybase_ats_230
255 /   - sqlsybase_ats_231
256 /   - sqlsybase_ats_232
257 /   - sqlsybase_ats_233
258 /   - sqlsybase_ats_234
259 /   - sqlsybase_ats_235
260 /   - sqlsybase_ats_236
261 /   - sqlsybase_ats_237
262 /   - sqlsybase_ats_238
263 /   - sqlsybase_ats_239
264 /   - sqlsybase_ats_240
265 /   - sqlsybase_ats_241
266 /   - sqlsybase_ats_242
267 /   - sqlsybase_ats_243
268 /   - sqlsybase_ats_244
269 /   - sqlsybase_ats_245
270 /   - sqlsybase_ats_246
271 /   - sqlsybase_ats_247
272 /   - sqlsybase_ats_248
273 /   - sqlsybase_ats_249
274 /   - sqlsybase_ats_250
275 /   - sqlsybase_ats_251
276 /   - sqlsybase_ats_252
277 /   - sqlsybase_ats_253
278 /   - sqlsybase_ats_254
279 /   - sqlsybase_ats_255
280 /   - sqlsybase_ats_256
281 /   - sqlsybase_ats_257
282 /   - sqlsybase_ats_258
283 /   - sqlsybase_ats_259
284 /   - sqlsybase_ats_260
285 /   - sqlsybase_ats_261
286 /   - sqlsybase_ats_262
287 /   - sqlsybase_ats_263
288 /   - sqlsybase_ats_264
289 /   - sqlsybase_ats_265
290 /   - sqlsybase_ats_266
291 /   - sqlsybase_ats_267
292 /   - sqlsybase_ats_268
293 /   - sqlsybase_ats_269
294 /   - sqlsybase_ats_270
295 /   - sqlsybase_ats_271
296 /   - sqlsybase_ats_272
297 /   - sqlsybase_ats_273
298 /   - sqlsybase_ats_274
299 /   - sqlsybase_ats_275
300 /   - sqlsybase_ats_276
301 /   - sqlsybase_ats_277
302 /   - sqlsybase_ats_278
303 /   - sqlsybase_ats_279
304 /   - sqlsybase_ats_280
305 /   - sqlsybase_ats_281
306 /   - sqlsybase_ats_282
307 /   - sqlsybase_ats_283
308 /   - sqlsybase_ats_284
309 /   - sqlsybase_ats_285
310 /   - sqlsybase_ats_286
311 /   - sqlsybase_ats_287
312 /   - sqlsybase_ats_288
313 /   - sqlsybase_ats_289
314 /   - sqlsybase_ats_290
315 /   - sqlsybase_ats_291
316 /   - sqlsybase_ats_292
317 /   - sqlsybase_ats_293
318 /   - sqlsybase_ats_294
319 /   - sqlsybase_ats_295
320 /   - sqlsybase_ats_296
321 /   - sqlsybase_ats_297
322 /   - sqlsybase_ats_298
323 /   - sqlsybase_ats_299
324 /   - sqlsybase_ats_300
325 /   - sqlsybase_ats_301
326 /   - sqlsybase_ats_302
327 /   - sqlsybase_ats_303
328 /   - sqlsybase_ats_304
329 /   - sqlsybase_ats_305
330 /   - sqlsybase_ats_306
331 /   - sqlsybase_ats_307
332 /   - sqlsybase_ats_308
333 /   - sqlsybase_ats_309
334 /   - sqlsybase_ats_310
335 /   - sqlsybase_ats_311
336 /   - sqlsybase_ats_312
337 /   - sqlsybase_ats_313
338 /   - sqlsybase_ats_314
339 /   - sqlsybase_ats_315
340 /   - sqlsybase_ats_316
341 /   - sqlsybase_ats_317
342 /   - sqlsybase_ats_318
343 /   - sqlsybase_ats_319
344 /   - sqlsybase_ats_320
345 /   - sqlsybase_ats_321
346 /   - sqlsybase_ats_322
347 /   - sqlsybase_ats_323
348 /   - sqlsybase_ats_324
349 /   - sqlsybase_ats_325
350 /   - sqlsybase_ats_326
351 /   - sqlsybase_ats_327
352 /   - sqlsybase_ats_328
353 /   - sqlsybase_ats_329
354 /   - sqlsybase_ats_330
355 /   - sqlsybase_ats_331
356 /   - sqlsybase_ats_332
357 /   - sqlsybase_ats_333
358 /   - sqlsybase_ats_334
359 /   - sqlsybase_ats_335
360 /   - sqlsybase_ats_336
361 /   - sqlsybase_ats_337
362 /   - sqlsybase_ats_338
363 /   - sqlsybase_ats_339
364 /   - sqlsybase_ats_340
365 /   - sqlsybase_ats_341
366 /   - sqlsybase_ats_342
367 /   - sqlsybase_ats_343
368 /   - sqlsybase_ats_344
369 /   - sqlsybase_ats_345
370 /   - sqlsybase_ats_346
371 /   - sqlsybase_ats_347
372 /   - sqlsybase_ats_348
373 /   - sqlsybase_ats_349
374 /   - sqlsybase_ats_350
375 /   - sqlsybase_ats_351
376 /   - sqlsybase_ats_352
377 /   - sqlsybase_ats_353
378 /   - sqlsybase_ats_354
379 /   - sqlsybase_ats_355
380 /   - sqlsybase_ats_356
381 /   - sqlsybase_ats_357
382 /   - sqlsybase_ats_358
383 /   - sqlsybase_ats_359
384 /   - sqlsybase_ats_360
385 /   - sqlsybase_ats_361
386 /   - sqlsybase_ats_362
387 /   - sqlsybase_ats_363
388 /   - sqlsybase_ats_364
389 /   - sqlsybase_ats_365
390 /   - sqlsybase_ats_366
391 /   - sqlsybase_ats_367
392 /   - sqlsybase_ats_368
393 /   - sqlsybase_ats_369
394 /   - sqlsybase_ats_370
395 /   - sqlsybase_ats_371
396 /   - sqlsybase_ats_372
397 /   - sqlsybase_ats_373
398 /   - sqlsybase_ats_374
399 /   - sqlsybase_ats_375
400 /   - sqlsybase_ats_376
401 /   - sqlsybase_ats_377
402 /   - sqlsybase_ats_378
403 /   - sqlsybase_ats_379
404 /   - sqlsybase_ats_380
405 /   - sqlsybase_ats_381
406 /   - sqlsybase_ats_382
407 /   - sqlsybase_ats_383
408 /   - sqlsybase_ats_384
409 /   - sqlsybase_ats_385
410 /   - sqlsybase_ats_386
411 /   - sqlsybase_ats_387
412 /   - sqlsybase_ats_388
413 /   - sqlsybase_ats_389
414 /   - sqlsybase_ats_390
415 /   - sqlsybase_ats_391
416 /   - sqlsybase_ats_392
417 /   - sqlsybase_ats_393
418 /   - sqlsybase_ats_394
419 /   - sqlsybase_ats_395
420 /   - sqlsybase_ats_396
421 /   - sqlsybase_ats_397
422 /   - sqlsybase_ats_398
423 /   - sqlsybase_ats_399
424 /   - sqlsybase_ats_400
425 /   - sqlsybase_ats_401
426 /   - sqlsybase_ats_402
427 /   - sqlsybase_ats_403
428 /   - sqlsybase_ats_404
429 /   - sqlsybase_ats_405
430 /   - sqlsybase_ats_406
431 /   - sqlsybase_ats_407
432 /   - sqlsybase_ats_408
433 /   - sqlsybase_ats_409
434 /   - sqlsybase_ats_410
435 /   - sqlsybase_ats_411
436 /   - sqlsybase_ats_412
437 /   - sqlsybase_ats_413
438 /   - sqlsybase_ats_414
439 /   - sqlsybase_ats_415
440 /   - sqlsybase_ats_416
441 /   - sqlsybase_ats_417
442 /   - sqlsybase_ats_418
443 /   - sqlsybase_ats_419
444 /   - sqlsybase_ats_420
445 /   - sqlsybase_ats_421
446 /   - sqlsybase_ats_422
447 /   - sqlsybase_ats_423
448 /   - sqlsybase_ats_424
449 /   - sqlsybase_ats_425
450 /   - sqlsybase_ats_426
451 /   - sqlsybase_ats_427
452 /   - sqlsybase_ats_428
453 /   - sqlsybase_ats_429
454 /   - sqlsybase_ats_430
455 /   - sqlsybase_ats_431
456 /   - sqlsybase_ats_432
457 /   - sqlsybase_ats_433
458 /   - sqlsybase_ats_434
459 /   - sqlsybase_ats_435
460 /   - sqlsybase_ats_436
461 /   - sqlsybase_ats_437
462 /   - sqlsybase_ats_438
463 /   - sqlsybase_ats_439
464 /   - sqlsybase_ats_440
465 /   - sqlsybase_ats_441
466 /   - sqlsybase_ats_442
467 /   - sqlsybase_ats_443
468 /   - sqlsybase_ats_444
469 /   - sqlsybase_ats_445
470 /   - sqlsybase_ats_446
471 /   - sqlsybase_ats_447
472 /   - sqlsybase_ats_448
473 /   - sqlsybase_ats_449
474 /   - sqlsybase_ats_450
475 /   - sqlsybase_ats_451
476 /   - sqlsybase_ats_452
477 /   - sqlsybase_ats_453
478 /   - sqlsybase_ats_454
479 /   - sqlsybase_ats_455
480 /   - sqlsybase_ats_456
481 /   - sqlsybase_ats_457
482 /   - sqlsybase_ats_458
483 /   - sqlsybase_ats_459
484 /   - sqlsybase_ats_460
485 /   - sqlsybase_ats_461
486 /   - sqlsybase_ats_462
487 /   - sqlsybase_ats_463
488 /   - sqlsybase_ats_464
489 /   - sqlsybase_ats_465
490 /   - sqlsybase_ats_466
491 /   - sqlsybase_ats_467
492 /   - sqlsybase_ats_468
493 /   - sqlsybase_ats_469
494 /   - sqlsybase_ats_470
495 /   - sqlsybase_ats_471
496 /   - sqlsybase_ats_472
497 /   - sqlsybase_ats_473
498 /   - sqlsybase_ats_474
499 /   - sqlsybase_ats_475
500 /   - sqlsybase_ats_476
501 /   - sqlsybase_ats_477
502 /   - sqlsybase_ats_478
503 /   - sqlsybase_ats_479
504 /   - sqlsybase_ats_480
505 /   - sqlsybase_ats_481
506 /   - sqlsybase_ats_482
507 /   - sqlsybase_ats_483
508 /   - sqlsybase_ats_484
509 /   - sqlsybase_ats_485
510 /   - sqlsybase_ats_486
511 /   - sqlsybase_ats_487
512 /   - sqlsybase_ats_488
513 /   - sqlsybase_ats_489
514 /   - sqlsybase_ats_490
515 /   - sqlsybase_ats_491
516 /   - sqlsybase_ats_492
517 /   - sqlsybase_ats_493
518 /   - sqlsybase_ats_494
519 /   - sqlsybase_ats_495
520 /   - sqlsybase_ats_496
521 /   - sqlsybase_ats_497
522 /   - sqlsybase_ats_498
523 /   - sqlsybase_ats_499
524 /   - sqlsybase_ats_500
525 /   - sqlsybase_ats_501
526 /   - sqlsybase_ats_502
527 /   - sqlsybase_ats_503
528 /   - sqlsybase_ats_504
529 /   - sqlsybase_ats_505
530 /   - sqlsybase_ats_506
531 /   - sqlsybase_ats_507
532 /   - sqlsybase_ats_508
533 /   - sqlsybase_ats_509
534 /   - sqlsybase_ats_510
535 /   - sqlsybase_ats_511
536 /   - sqlsybase_ats_512
537 /   - sqlsybase_ats_513
538 /   - sqlsybase_ats_514
539 /   - sqlsybase_ats_515
540 /   - sqlsybase_ats_516
541 /   - sqlsybase_ats_517
542 /   - sqlsybase_ats_518
543 /   - sqlsybase_ats_519
544 /   - sqlsybase_ats_520
545 /   - sqlsybase_ats_521
546 /   - sqlsybase_ats_522
547 /   - sqlsybase_ats_523
548 /   - sqlsybase_ats_524
549 /   - sqlsybase_ats_525
550 /   - sqlsybase_ats_526
551 /   - sqlsybase_ats_527
552 /   - sqlsybase_ats_528
553 /   - sqlsybase_ats_529
554 /   - sqlsybase_ats_530
555 /   - sqlsybase_ats_531
556 /   - sqlsybase_ats_532
557 /   - sqlsybase_ats_533
558 /   - sqlsybase_ats_534
559 /   - sqlsybase_ats_535
560 /   - sqlsybase_ats_536
561 /   - sqlsybase_ats_537
562 /   - sqlsybase_ats_538
563 /   - sqlsybase_ats_539
564 /   - sqlsybase_ats_540
565 /   - sqlsybase_ats_541
566 /   - sqlsybase_ats_542
567 /   - sqlsybase_ats_543
568 /   - sqlsybase_ats_544
569 /   - sqlsybase_ats_545
570 /   - sqlsybase_ats_546
571 /   - sqlsybase_ats_547
572 /   - sqlsybase_ats_548
573 /   - sqlsybase_ats_549
574 /   - sqlsybase_ats_550
575 /   - sqlsybase_ats_551
576 /   - sqlsybase_ats_552
577 /   - sqlsybase_ats_553
578 /   - sqlsybase_ats_554
579 /   - sqlsybase_ats_555
580 /   - sqlsybase_ats_556
581 /   - sqlsybase_ats_557
582 /   - sqlsybase_ats_558
583 /   - sqlsybase_ats_559
584 /   - sqlsybase_ats_560
585 /   - sqlsybase_ats_561
586 /   - sqlsybase_ats_562
587 /   - sqlsybase_ats_563
588 /   - sqlsybase_ats_564
589 /   - sqlsybase_ats_565
590 /   - sqlsybase_ats_566
591 /   - sqlsybase_ats_567
592 /   - sqlsybase_ats_568
593 /   - sqlsybase_ats_569
594 /   - sqlsybase_ats_570
595 /   - sqlsybase_ats_571
596 /   - sqlsybase_ats_572
597 /   - sqlsybase_ats_573
598 /   - sqlsybase_ats_574
599 /   - sqlsybase_ats_575
600 /   - sqlsybase_ats_576
601 /   - sqlsybase_ats_577
602 /   - sqlsybase_ats_578
603 /   - sqlsybase_ats_579
604 /   - sqlsybase_ats_580
605 /   - sqlsybase_ats_581
606 /   - sqlsybase_ats_582
607 /   - sqlsybase_ats_583
608 /   - sqlsybase_ats_584
609 /   - sqlsybase_ats_585
610 /   - sqlsybase_ats_586
611 /   - sqlsybase_ats_587
612 /   - sqlsybase_ats_588
613 /   - sqlsybase_ats_589
614 /   - sqlsybase_ats_590
615 /   - sqlsybase_ats_591
616 /   - sqlsybase_ats_592
617 /   - sqlsybase_ats_593
618 /   - sqlsybase_ats_594
619 /   - sqlsybase_ats_595
620 /   - sqlsybase_ats_596
621 /   - sqlsybase_ats_597
622 /   - sqlsybase_ats_598
623 /   - sqlsybase_ats_599
624 /   - sqlsybase_ats_600
625 /   - sqlsybase_ats_601
626 /   - sqlsybase_ats_602
627 /   - sqlsybase_ats_603
628 /   - sqlsybase_ats_604
629 /   - sqlsybase_ats_605
630 /   - sqlsybase_ats_606
631 /   - sqlsybase_ats_607
632 /   - sqlsybase_ats_608
633 /   - sqlsybase_ats_609
634 /   - sqlsybase_ats_610
635 /   - sqlsybase_ats_611
636 /   - sqlsybase_ats_612
637 /   - sqlsybase_ats_613
638 /   - sqlsybase_ats_614
639 /   - sqlsybase_ats_615
640 /   - sqlsybase_ats_616
641 /   - sqlsybase_ats_617
642 /   - sqlsybase_ats_618
643 /   - sqlsybase_ats_619
644 /   - sqlsybase_ats_620
645 /   - sqlsybase_ats_621
646 /   - sqlsybase_ats_622
647 /   - sqlsybase_ats_623
648 /   - sqlsybase_ats_624
649 /   - sqlsybase_ats_625
650 /   - sqlsybase_ats_626
651 /   - sqlsybase_ats_627
652 /   - sqlsybase_ats_628
653 /   - sqlsybase_ats_629
654 /   - sqlsybase_ats_630
655 /   - sqlsybase_ats_631
656 /   - sqlsybase_ats_632
657 /   - sqlsybase_ats_633
658 /   - sqlsybase_ats_634
659 /   - sqlsybase_ats_635
660 /   - sqlsybase_ats_636
661 /   - sqlsybase_ats_637
662 /   - sqlsybase_ats_638
663 /   - sqlsybase_ats_639
664 /   - sqlsybase_ats_640
665 /   - sqlsybase_ats_641
666 /   - sqlsybase_ats_642
667 /   - sqlsybase_ats_643
668 /   - sqlsybase_ats_644
669 /   - sqlsybase_ats_645
670 /   - sqlsybase_ats_646
671 /   - sqlsybase_ats_647
672 /   - sqlsybase_ats_648
673 /   - sqlsybase_ats_649
674 /   - sqlsybase_ats_650
675 /   - sqlsybase_ats_651
676 /   - sqlsybase_ats_652
677 /   - sqlsybase_ats_653
678 /   - sqlsybase_ats_654
679 /   - sqlsybase_ats_655
680 /   - sqlsybase_ats_656
681 /   - sqlsybase_ats_657
682 /   - sqlsybase_ats_658
683 /   - sqlsybase_ats_659
684 /   - sqlsybase_ats_660
685 /   - sqlsybase_ats_661
686 /   - sqlsybase_ats_662
687 /   - sqlsybase_ats_663
688 /   - sqlsybase_ats_664
689 /   - sqlsybase_ats_665
690 /   - sqlsybase_ats_666
691 /   - sqlsybase_ats_667
692 /   - sqlsybase_ats_668
693 /   - sqlsybase_ats_669
694 /   - sqlsybase_ats_670
695 /   - sqlsybase_ats_671
696 /   - sqlsybase_ats_672
697 /   - sqlsybase_ats_673
698 /   - sqlsybase_ats_674
699 /   - sqlsybase_ats_675
700 /   - sqlsybase_ats_676
701 /   - sqlsybase_ats_677
702 /   - sqlsybase_ats_678
703 /   - sqlsybase_ats_679
704 /   - sqlsybase_ats_680
705 /   - sqlsybase_ats_681
706 /   - sqlsybase_ats_682
707 /   - sqlsybase_ats_683
708 /   - sqlsybase_ats_684
709 /   - sqlsybase_ats_685
710 /   - sqlsybase_ats_686
711 /   - sqlsybase_ats_687
712 /   - sqlsybase_ats_688
713 /   - sqlsybase_ats_689
714 /   - sqlsybase_ats_690
715 /   - sqlsybase_ats_691
716 /   - sqlsybase_ats_692
717 /   - sqlsybase_ats_693
718 /   - sqlsybase_ats_694
719 /   - sqlsybase_ats_695
720 /   - sqlsybase_ats_696
721 /   - sqlsybase_ats_697
722 /   - sqlsybase_ats_698
723 /   - sqlsybase_ats_699
724 /   - sqlsybase_ats_700
725 /   - sqlsybase_ats_701
726 /   - sqlsybase_ats_702
727 /   - sqlsybase_ats_703
728 /   - sqlsybase_ats_704
729 /   - sqlsybase_ats_705
730 /   - sqlsybase_ats_706
731 /   - sqlsybase_ats_707
732 /   - sqlsybase_ats_708
733 /   - sqlsybase_ats_709
734 /   - sqlsybase_ats_710
735 /   - sqlsybase_ats_711
736 /   - sqlsybase_ats_712
737 /   - sqlsybase_ats_713
738 /   - sqlsybase_ats_714
739 /   - sqlsybase_ats_715
740 /   - sqlsybase_ats_716
741 /   - sqlsybase_ats_717
742 /   - sqlsybase_ats_718
743 /   - sqlsybase_ats_719
744 /   - sqlsybase_ats_720
745 /   - sqlsybase_ats_721
746 /   - sqlsybase_ats_722
747 /   - sqlsybase_ats_723
748 /   - sqlsybase_ats_724
749 /   - sqlsybase_ats_725
750 /   - sqlsybase_ats_726
751 /   - sqlsybase_ats_727
752 /   - sqlsybase_ats_728
753 /   - sqlsybase_ats_729
754 /   - sqlsybase_ats_730
755 /   - sqlsybase_ats_731
756 /   - sqlsybase_ats_732
757 /   - sqlsybase_ats_733
758 /   - sqlsybase_ats_734
759 /   - sqlsybase_ats_735
760 /   - sqlsybase_ats_736
7
```



- i. Kemudian temukan kode seperti berikut pada file database.php.

```
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => '',
80     'password' => '',
81     'database' => '',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97
```

- j. Kemudian tambahkan kode tersebut seperti gambar berikut.

```

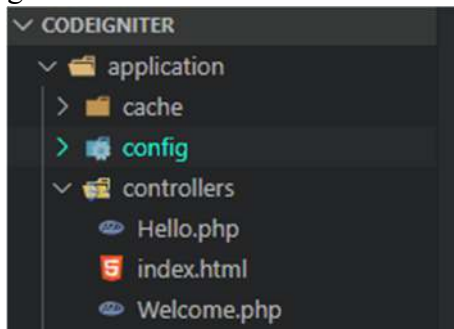
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost', //hostname
79     'username' => 'root', //username
80     'password' => '', //password
81     'database' => 'database_name', //nama database
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97

```

#### A. Hello World Codeigniter

Untuk menguji pemahaman pada codeigniter, kita akan mencoba dengan menampilkan text “Hello World” pada browser menggunakan controller.

Buat sebuah controller dengan nama Hello.php seperti gambar berikut.



Setelah itu, ketikkan kode seperti gambar berikut.

```
application > controllers > Hello.php
1  <?php
2  class Hello extends CI_Controller{
3      function index(){
4          echo "Hello World!";
5      }
6  }
7  <?>
```

Jika berhasil maka hasilnya akan seperti gambar berikut.

