

**PANDUAN LENGKAP MENGENAI SISTEM DASHBOARD
MONITORING KINERJA MAHASISWA INTERNSHIP
BERBASIS GPS (GLOBAL POSITIONING SYSTEM)**

**PANDUAN LENGKAP MENGENAI SISTEM DASHBOARD
MAHASISWA INTERNSHIP BERBASIS GPS (GLOBAL
POSITIONING SYSTEM)**

Aip Suprpto Munari

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 *dashboard* monitoring kinerja mahasiswa *internship* berbasis gps (*global positioning system*)

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 Preprocessing / 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 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* “`<`”, *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* jugadihapus 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), Eiffel 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.

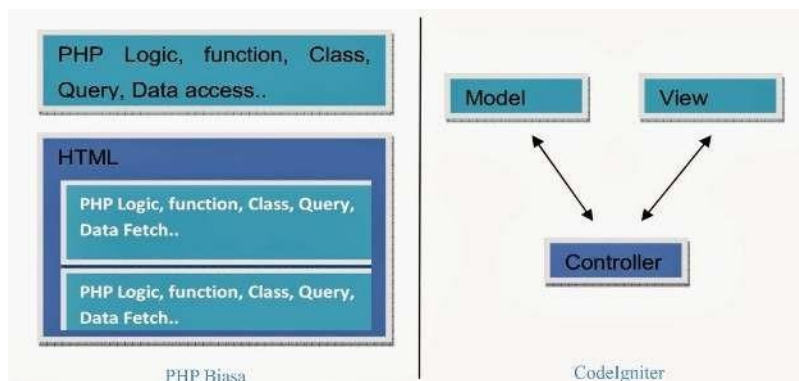


1.11 Pengertian CodeIgniter

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.



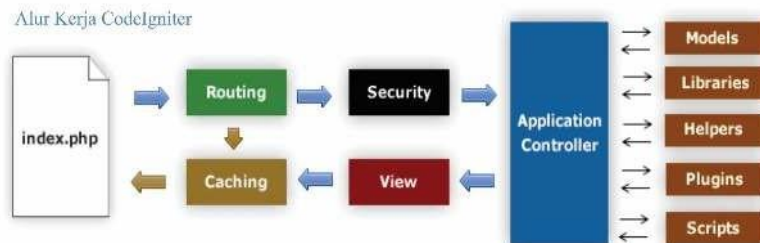
MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi, diantaranya :

- 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

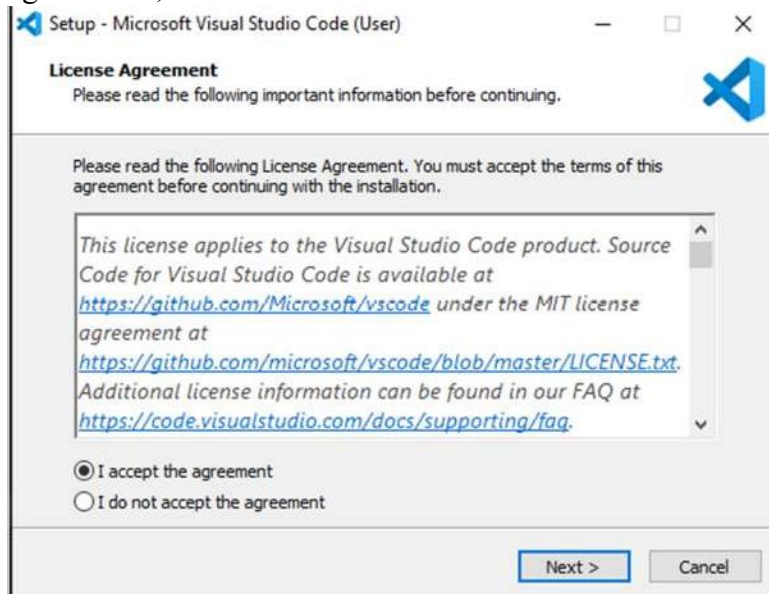
Visual studio code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node.js, serta 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.

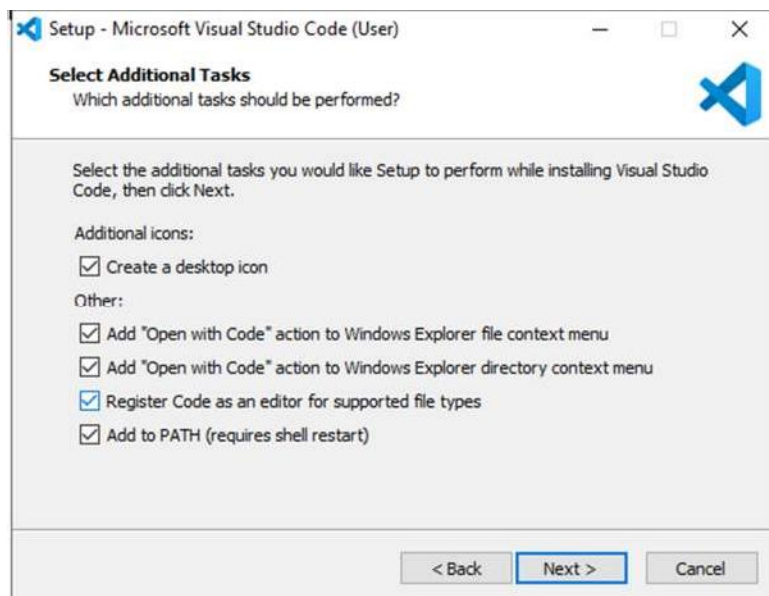
- a. Pertama, *download* terlebih dahulu file installer visual studio code melalui situs resminya pada link berikut ini <https://code.visualstudio.com/download>



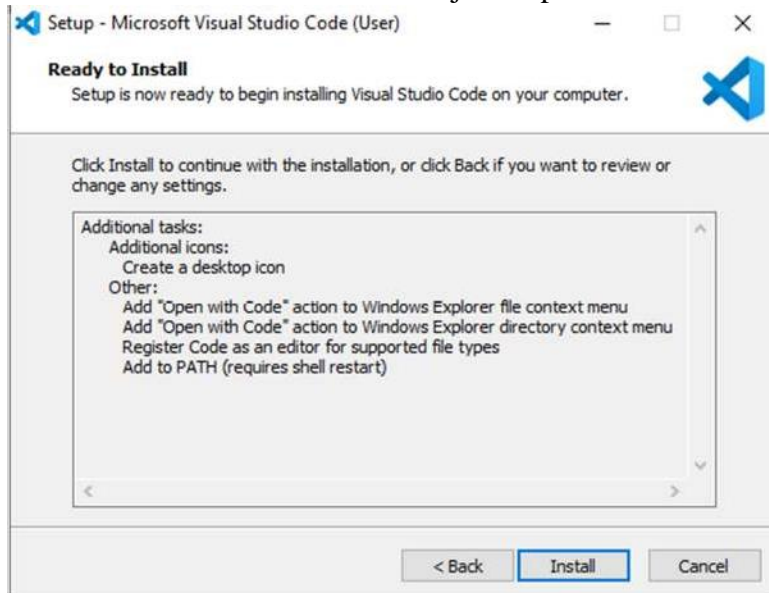
- b. Terdapat 2 pilihan untuk sistem operasi windows, diantaranya ada yang 64 bit dan 32 bit. Pilihlah installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Jika, sudah berhasil di *download*, lanjut pada proses instalasi, pertama klik pada file installer vs code.
- d. Pilih “I accept the agreement”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



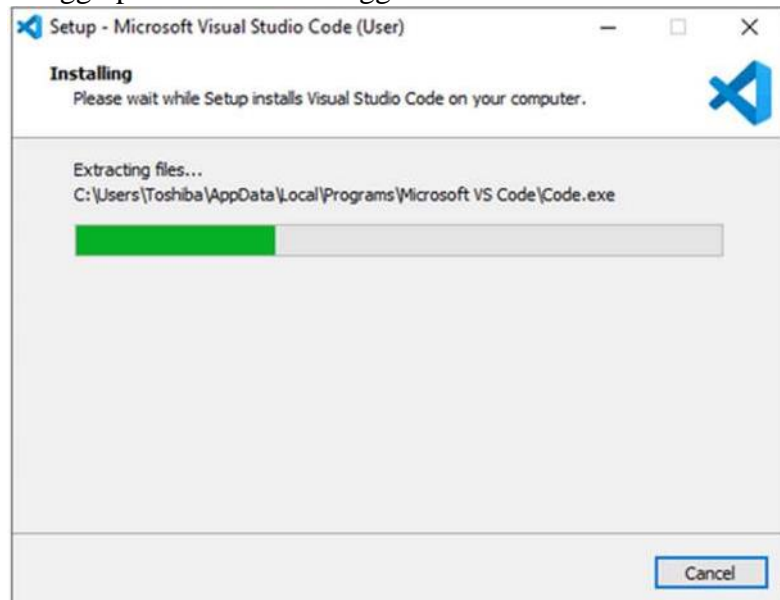
- e. Pada bagian select additional tasks, centang semuanya kemudian klik next.



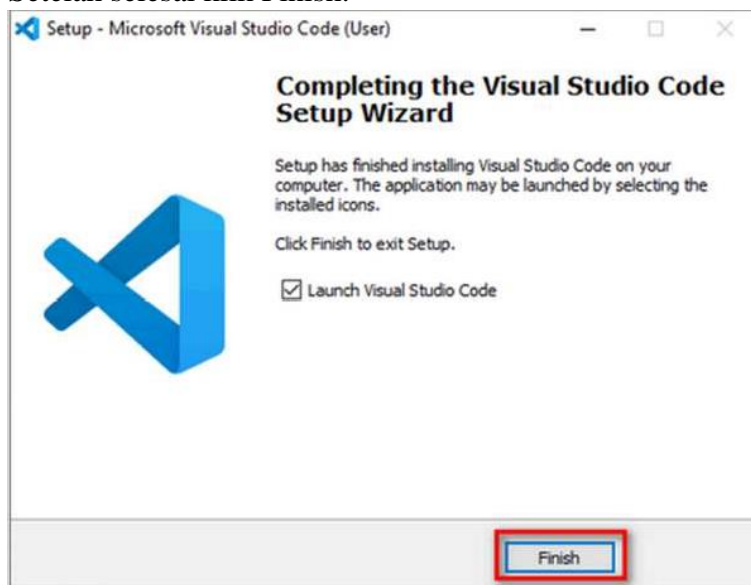
f. Kemudian klik install untuk melanjutkan proses instalasi.



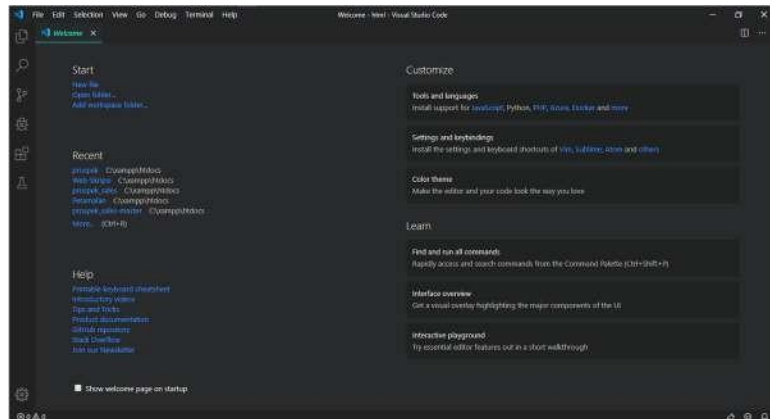
g. Tunggu proses instalasi hingga selesai.



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*).

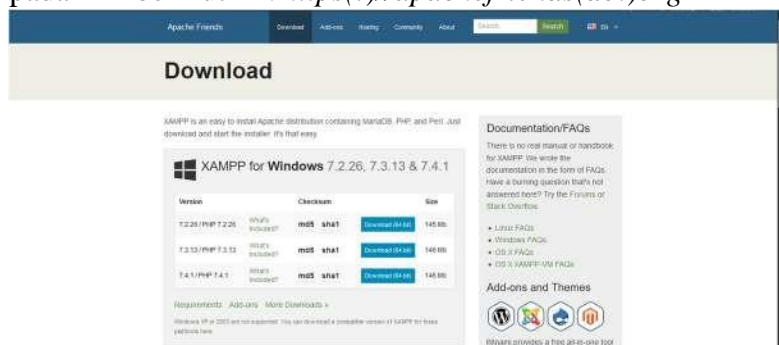
M = MySQL / MariaDB, MySQL atau MariaDB merupakan aplikasi *database server* yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui *database*.

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.

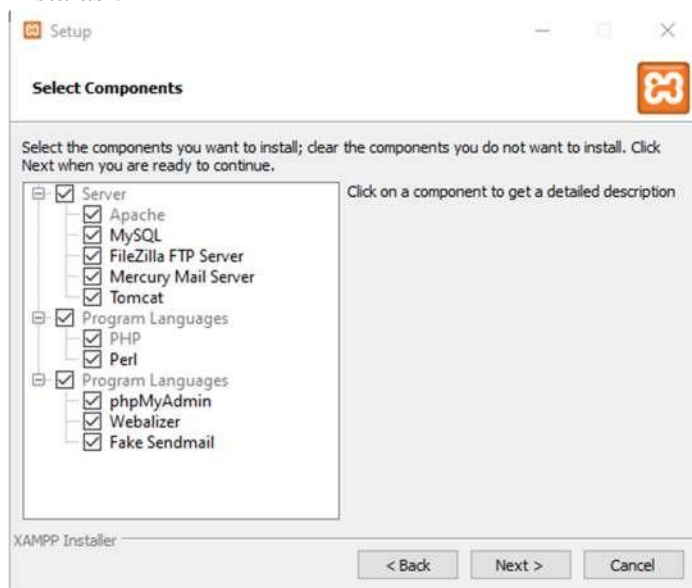
- a. Pertama, *download* terlebih dahulu file installer xampp pada link berikut ini. <https://apachefriends.org>



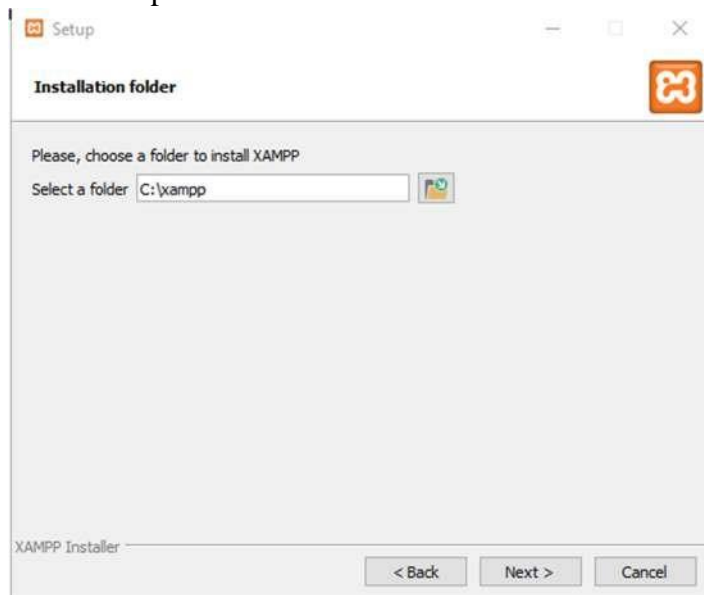
- b. Pilih file installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Klik file xampp yang sudah berhasil di *download*, maka akan muncul tampilan seperti ini.



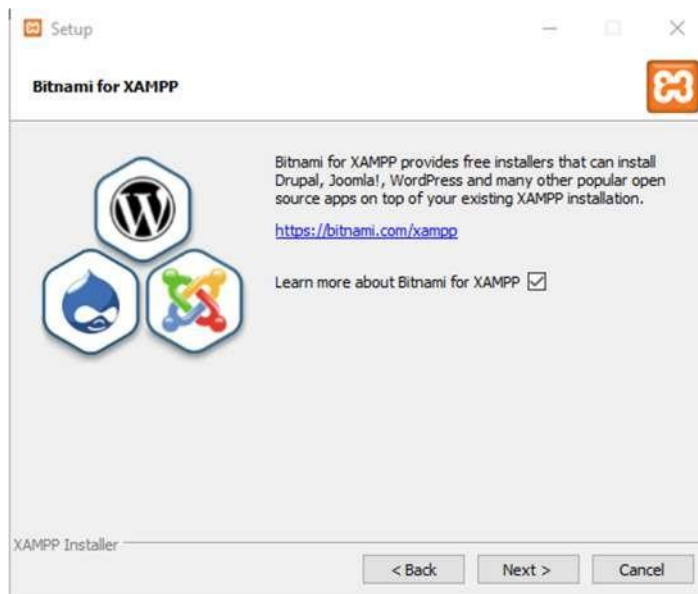
- d. Kemudian, pilih next untuk melanjutkan pada proses instalasi.



- 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.



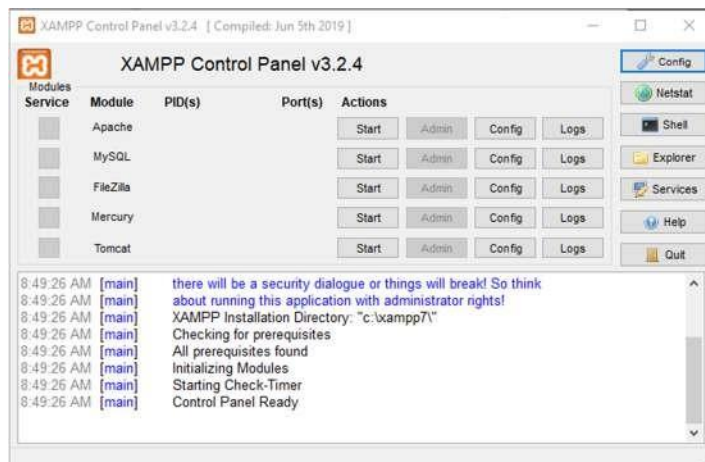
- h. Kemudian tunggu hingga proses instalasi selesai.



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



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



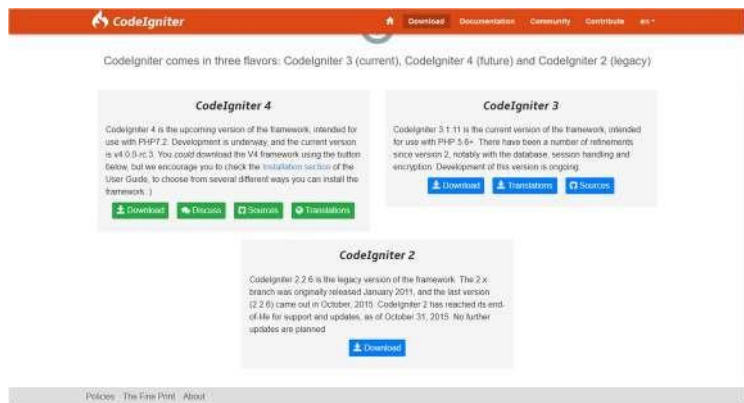
k. Jika berhasil, maka tampilan utama *control panel* dari xampp seperti berikut.

3. Codeigniter

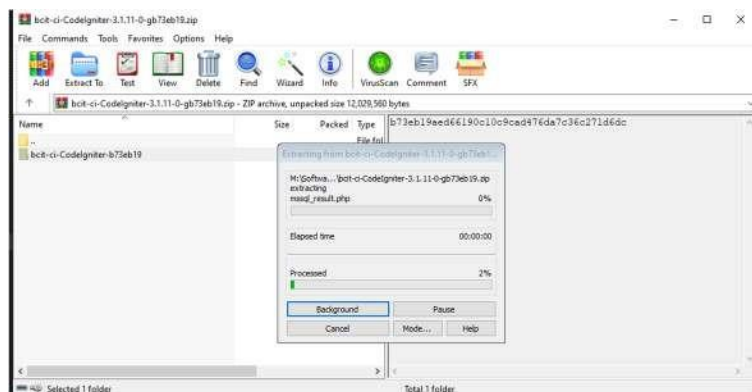
Codeigniter (CI) merupakan sebuah *web application framework* yang bersifat *open source* dimana digunakan untuk membangun aplikasi php dinamis. *Framework* itu sendiri merupakan abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh kode yang dibuat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

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

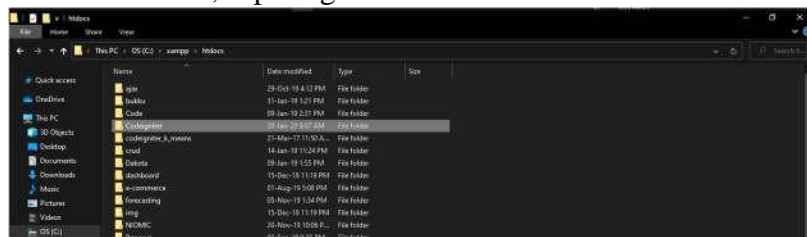
a. Pertama, *download* terlebih dahulu codeigniter pada *website* resminya pada link berikut.
<https://codeigniter.com/download>.



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



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



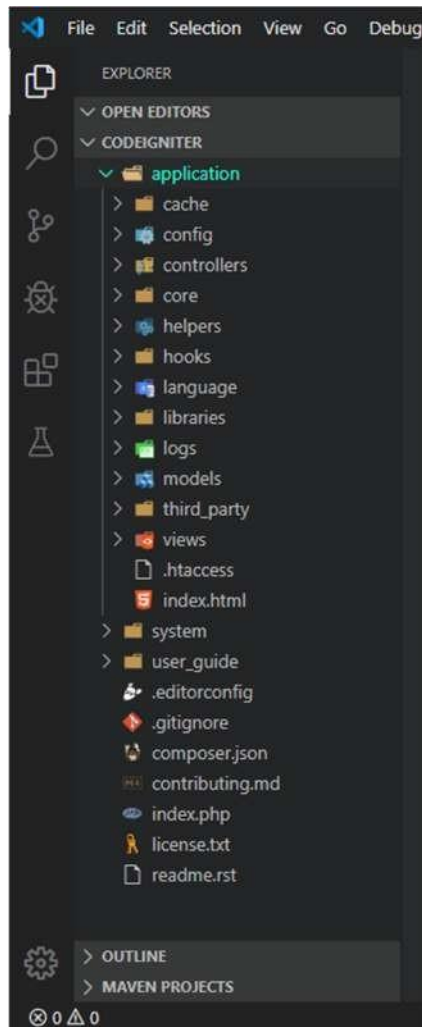
- d. Untuk menjalankannya, kita menggunakan xampp, untuk itu silakan jalankan terlebih dahulu aplikasi xampp-nya seperti berikut.
- e. Buka aplikasi xamppnya, kemudian klik tombol start pada kolom actions baris pertama untuk menjalankan apache, dan klik tombol start pada kolom actions baris kedua untuk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

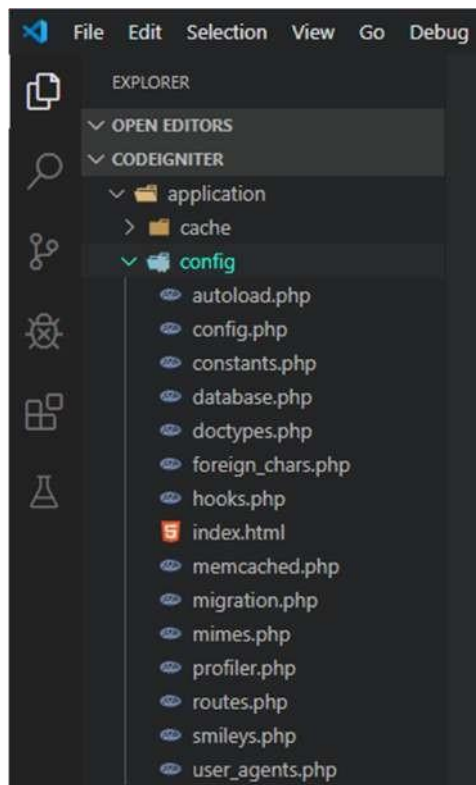


- a) *Application*, merupakan folder yang pada dasarnya menyimpan aplikasi yang sedang kita buat.
- b) *Cache*, merupakan folder yang menyimpan semua cache yang dibuat oleh cache library.

- 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 .....
6 AUTO-LOADER
7 .....
8 This file specifies which systems should be loaded by default.
9
10 In order to keep the framework as light-weight as possible only the
11 absolute minimum resources are loaded by default. For example,
12 the database is not connected to automatically since no assumption
13 is made regarding whether you intend to use it. This file lets
14 you globally define which systems you would like loaded with every
15 request.
16
17 .....
18 Instructions
19 .....
20 These are the things you can load automatically:
21
22 1. Packages
23 2. Libraries
24 3. Drivers
25 4. Helper files
26 5. Custom config files
27
```

- 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.

```
autoload.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, CodeIgniter will try guess the protocol and path
17 // your installation, but due to security concerns the hostname will be set
18 // to X-MISSING (lower case) 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 PHP script and you can only 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.

```
autoload.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 EXPLANATION OF VARIABLES
14 .....
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, mssql, mysql, mysqli, oci8,
26 odbc, pdo, pdo_mysql, sqlsrv, sqls,
27 ['dbprefix'] You can add an optional prefix, which will be added
28
```


- 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.

```

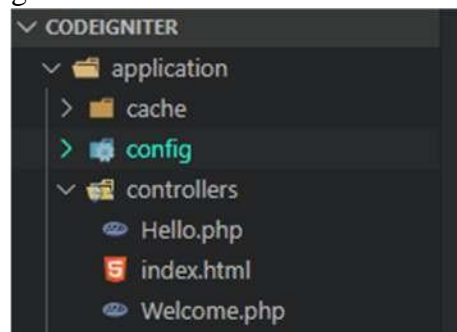
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost', //hostname
79     'username' => 'root', //username
80     'password' => '', //password
81     'database' => 'database_name', //nama database
82     'dbdriver' => '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.

