

**PERANGANCAN SISTEM PROSPEK
SALES MENGGUNAKAN REGRESI
LINIER BERGANDA**

**PERANCANGAN 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, kedependekan dari *Hypertext Prerocessing / Form Interpreter*.

Dengan perilisan 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 ke dalam 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* “ \Leftrightarrow ”, *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), 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 maintain 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](http://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.

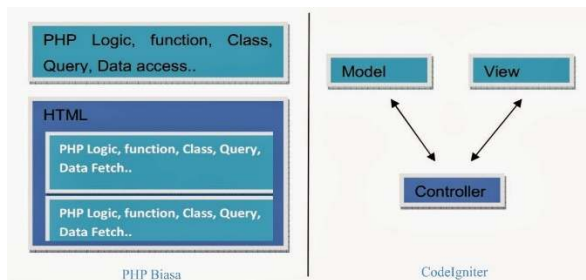


1.11 Pengertian CodeIgniter

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.

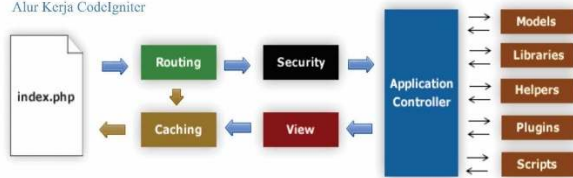


MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi, diantaranya :

- 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 :

Alur Kerja CodeIgniter



- 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 pembuaan *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).

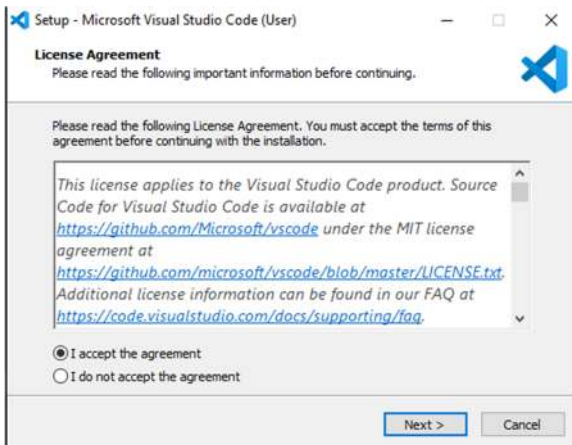
Berikut ini merupakan langkah-langkah instalasi visual studio code pada sistem operasi windows.

- a. Pertama, *download* terlebih dahulu file installer visual studio code melalui situs

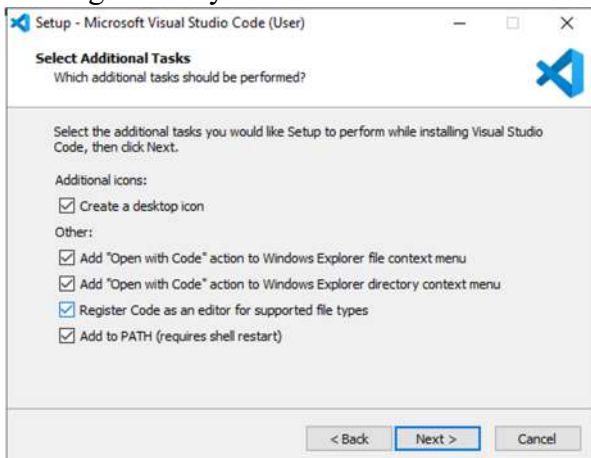
resminya pada link berikut ini
<https://code.visualstudio.com/download>



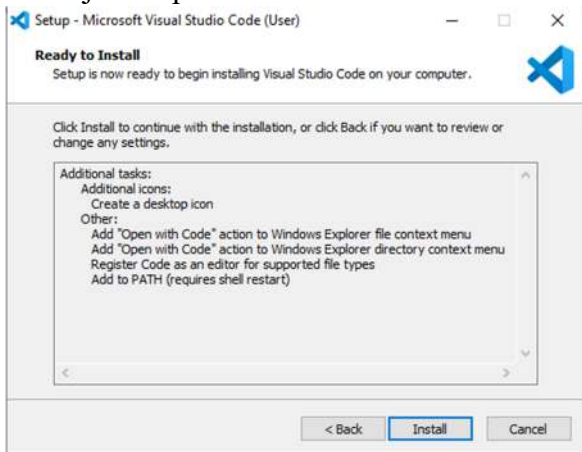
- b. Terdapat 2 pilihan untuk sistem operasi windows, diantaranya ada yang 64 bit dan 32 bit. Pilihlah installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Jika, sudah berhasil di *download*, lanjut pada proses instalasi, pertama klik pada file installer vs code.
- d. Pilih “I accept the agreement”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



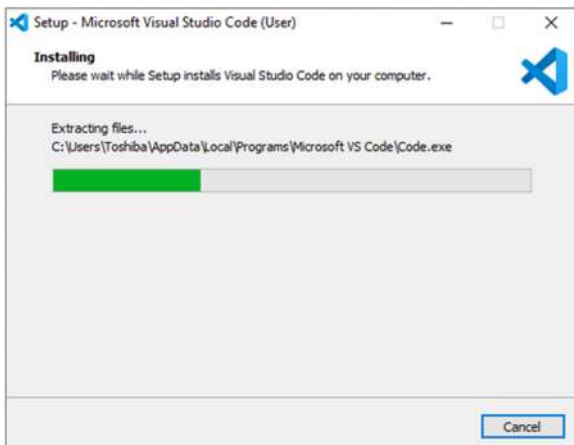
- e. Pada bagian select additional tasks, centang semuanya kemudian klik next.



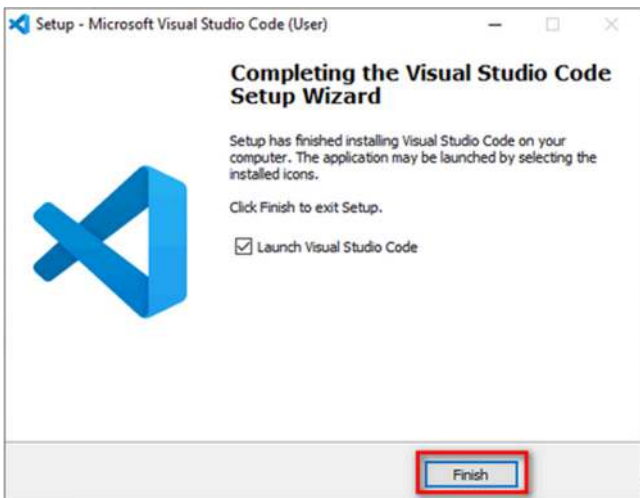
- f. Kemudian klik install untuk melanjutkan proses instalasi.



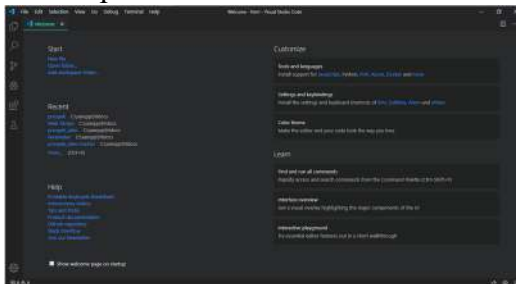
- g. Tunggu proses instalasi hingga selesai.



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*.

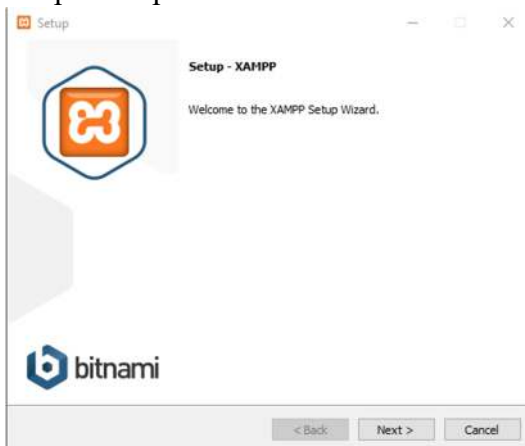
P = Perl, untuk huruf “P” selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi, sehingga sangat fleksibel dan banyak digunakan.

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

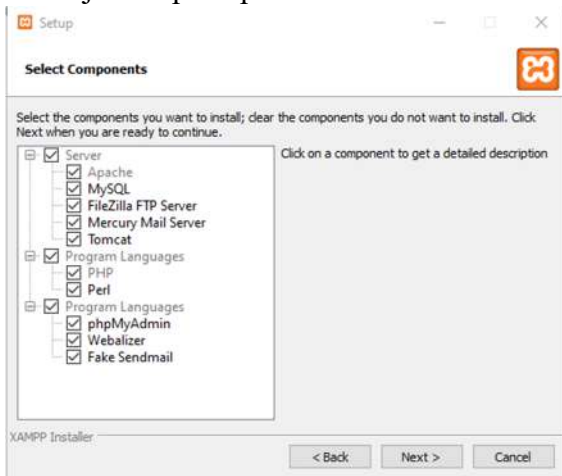
- a. Pertama, *download* terlebih dahulu file installer xampp pada link berikut ini.
<https://apachefriends.org>



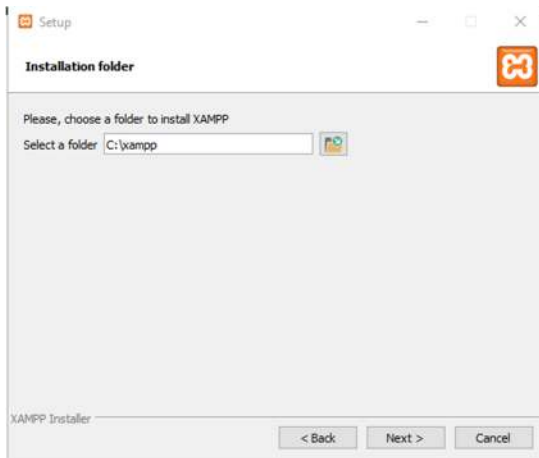
- b. Pilih file installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Klik file xampp yang sudah berhasil di *download*, maka akan muncul tampilan seperti ini.



- 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.



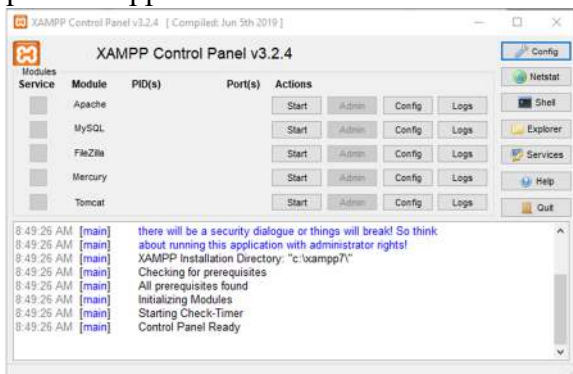
- h. Kemudian tunggu hingga proses instalasi selesai.



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



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



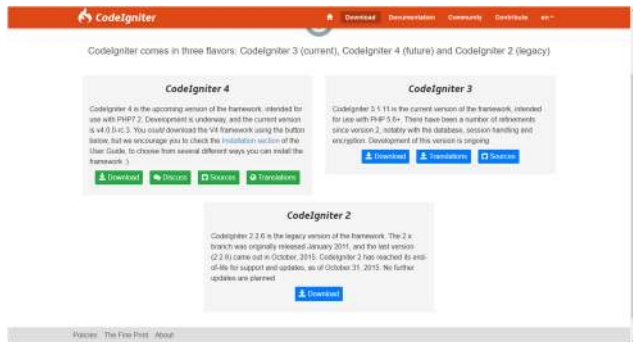
- k. Jika berhasil, maka tampilan utama *control panel* dari xampp seperti berikut.

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.

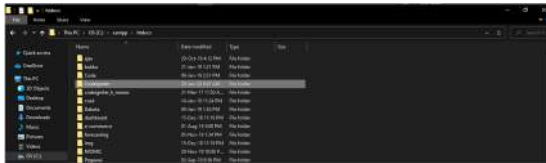
- a. Pertama, *download* terlebih dahulu codeigniter pada *website* resminya pada link berikut. *https(://codeigniter(dot)com/download*.



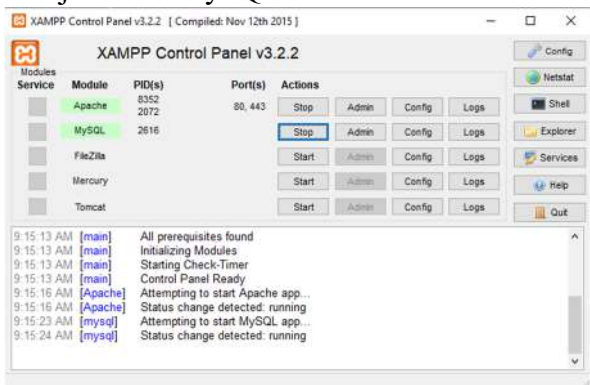
- b. Setelah berhasil di *download*, selanjutnya ekstrak file tersebut.



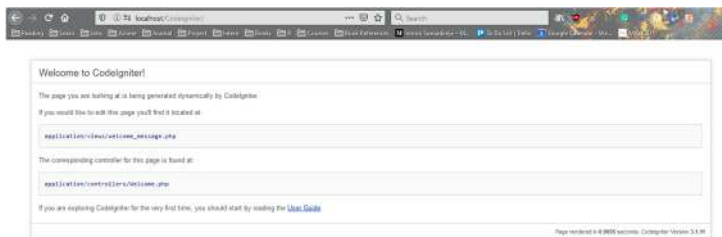
- c. Kemudian, *copy* file yang telah di ekstrak ke dalam direktorihtdocs, seperti gambar berikut.



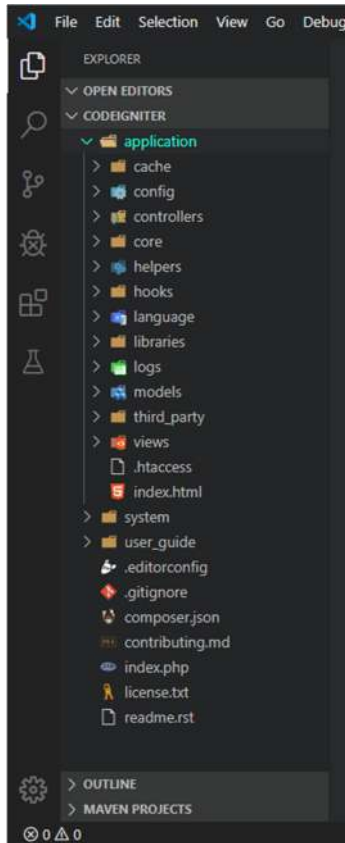
- d. Untuk menjalankannya, kita menggunakan xampp, untuk itu silakan jalankan terlebih dahulu aplikasi xampp-nya seperti berikut.
- e. Buka aplikasi xamppnya, kemudian klik tombol start pada kolom actions baris pertama untuk menjalankan apache, dan klik tombol start pada kolom actions baris kedua untuk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



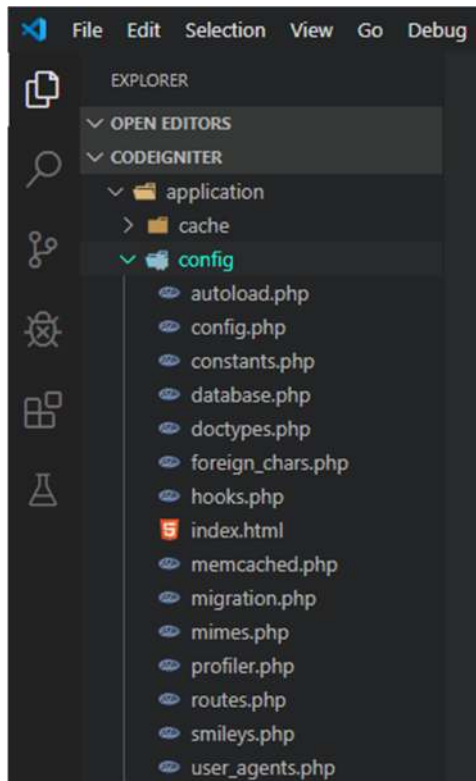
- a) *Application*, merupakan folder yang pada dasarnya menyimpan aplikasi yang sedang kita buat.

- b) *Cache*, merupakan folder yang menyimpan semua cache yang dibuat oleh cache library.
- c) *Config*, merupakan folder yang menyimpan informasi mengenai konfigurasi aplikasi seperti autoloader, 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.

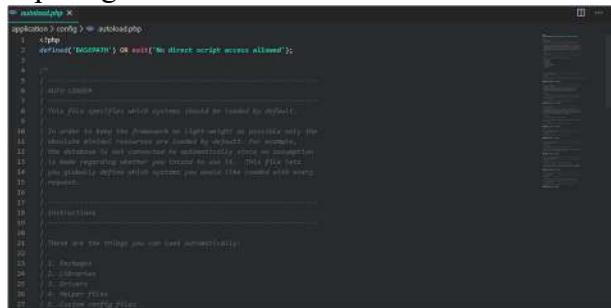
4. Konfigurasi dasar pada codeigniter

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

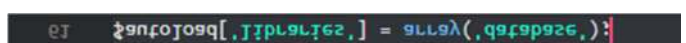
autoload.php, silakan buka file-nya seperti gambar berikut.

A screenshot of a code editor showing the contents of the 'autoload.php' file. The file is a PHP script that defines a function 'define('MAGNETO') or exit('No direct script access allowed');'. It also contains a large block of comments explaining the purpose of the file and the 'MAGNETO' constant. The code is written in a dark-themed editor with syntax highlighting.

b. Kemudian temukan kode berikut.

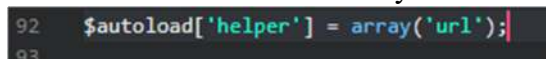
A screenshot of a code editor showing a single line of PHP code: `$autoload['JIPUL9L762'] = 9LL9L();`. The code is highlighted in a dark-themed editor.

c. Ubah kode tersebut menjadi seperti berikut.

A screenshot of a code editor showing a modified line of PHP code: `$autoload['JIPUL9L762'] = 9LL9L('q9f9p926');`. The code is highlighted in a dark-themed editor.

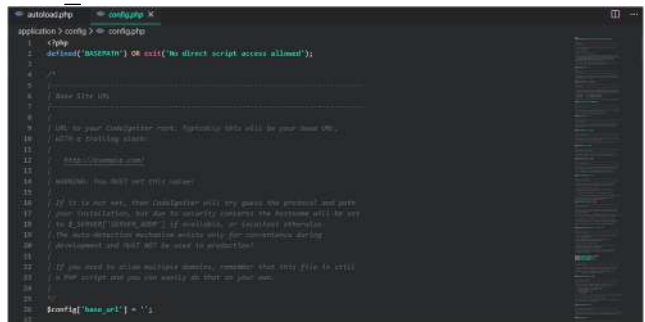
Kode tersebut dapat diartikan kita dapat meload library ‘database’ secara otomatis.

d. Selanjutnya, temukan kode berikut, dan tambahkan ‘url’ didalamnya.

A screenshot of a code editor showing two lines of PHP code. The first line is `$autoload['helper'] = array('url');` and the second line is `93`. The code is highlighted in a dark-themed editor.

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.



```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4
5
6 // Base URL (URI)
7
8 // Oh, to your developer mind, depending this will be your base URL.
9 // It's a trailing slash.
10
11 // Base URL (example.com)
12
13 // Example: You MUST use this value!
14
15 // If it is not set, your developer will try guess the protocol and path.
16 // Your initialization, but due to security concerns the baseurl will be set
17 // to $_SERVER['HTTP_HOST'] if available, or equivalent otherwise.
18 // The auto-detection mechanism works only for convenience during
19 // development and MUST NOT be used in production!
20
21 // If you need to allow multiple domains, remember that only file is valid.
22 // A PHP script can only exist at that on your own.
23
24
25 $config['base_url'] = '';
```

- f. Kemudian temukan kode berikut.

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

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

```
38 $config['base_url'] = 'http://localhost:8080/';
```

- 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.

A screenshot of a code editor with a dark theme. The file is named 'database.php' and is located in a directory named 'application > config'. The code defines a class 'Database' with a static property 'instance' and a static method 'get_instance()'. The 'get_instance()' method checks if an instance of 'Database' already exists; if not, it creates a new instance. The 'Database' class has several public properties: 'hostname', 'username', 'password', 'database', 'charset', 'collat', 'prefix', and 'debug'. These properties are used in the 'connect()' method to establish a database connection using 'mysqli_connect()'. The 'connect()' method returns the connection object or null if there is an error. The 'debug' property is used to display error messages if set to true. The code is as follows:

```
1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
263
```

```

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.

```

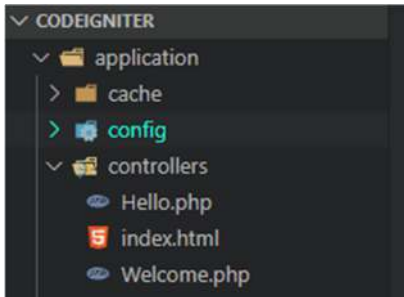
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost', //hostname
79     'username' => 'root', //username
80     'password' => '', //password
81     'database' => 'database_name', //nama database
82     'dbdriver' =>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



Jika berhasil maka hasilnya akan seperti gambar berikut.



BAB III

PERANCANGAN SISTEM

5.1 Perancangan

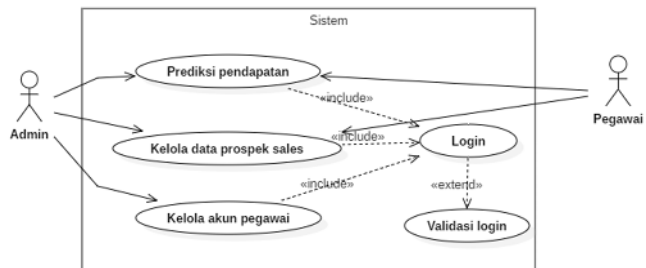
Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. Tahapan ini meliputi mengonfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

5.2.1 *Unified Modelling Language* (UML)

5.3.1 *Use Case Diagram*

Use case diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antara

aktor dengan aktivitas yang terdapat pada sistem. Adapun *use case diagram* pada aplikasi peramalan jumlah pendapatan PT Telkom dengan paket internet indihome ini, yaitu sebagai berikut:

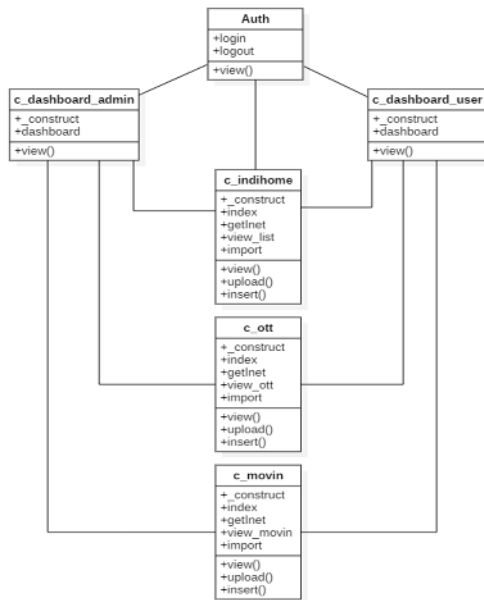


Gambar 5. 1 Use Case Diagram Prediksi Pendapatan

5.4.1 Class Diagram

Class diagram menggambarkan struktur dan hubungan antar objek-objek

yang ada pada sistem. Struktur itu meliputi atribut-atribut dan metode-metode yang ada pada masing-masing kelas. Adapau *class diagram* pada aplikasi peramalan ini, sebagai berikut:



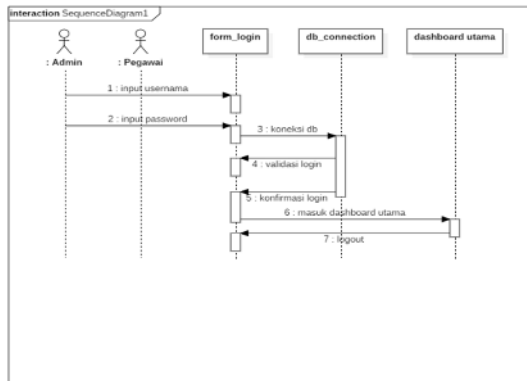
Gambar 5. 2 Class Diagram Prediksi Pendapatan

5.5.1 Sequence Diagram

Sequence diagram

menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. Adapun *sequence diagram* pada aplikasi peramalan ini, sebagai berikut:

5.5.1.1 Sequence Diagram Login

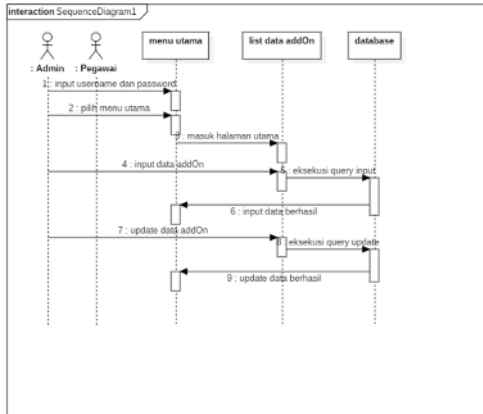


Gambar 5. 3 Sequence Diagram Login

Keterangan :

Pertama admin/pegawai masuk ke *form login* dengan memasukkan *username* dan *password* pada *form login*, lalu sistem akan memberikan koneksi ke *database* dan *database* memberikan validasi dan konfirmasi berhasil *login*, setelah itu masuk ke *dashboard* utama admin/pegawai, lalu *logout* jika telah selesai.

5.5.1.2 Sequence Diagram Kelola Data Prospek Sales



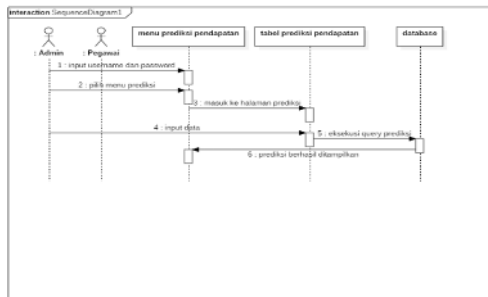
Gambar 5. 4 Sequence Diagram Kelola Data
AddOn

Keterangan :

Pertama admin/pegawai *login* terlebih dahulu pada sistem, sesudah *login* akan masuk pada halaman utama, setelah itu *upload* data sesuai dengan *addOn* yang telah dipilih, lalu *update* data *addOn* jika terjadi penambahan

data atau pengurangan data pelanggan pada *addOn* tertentu, setelah itu sistem akan mengeksekusinya dan data dari *database* akan berubah.

5.5.1.3 *Sequence Diagram* Prediksi Pendapatan



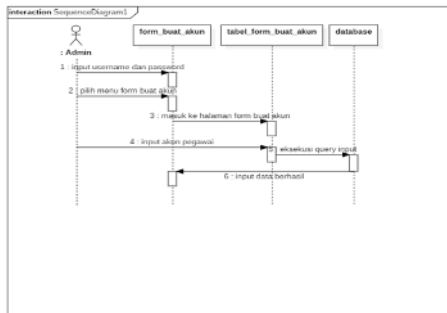
Gambar 5. 5 *Sequence Diagram* Prediksi Pendapatan

Keterangan :

Menu prediksi pendapatan ini dapat diakses oleh admin maupun pegawai. Pertama, aktor *login* terlebih dahulu, sesudah *login* memilih menu

prediksi pendapatan, lalu sistem akan menampilkan halaman prediksi pendapatan dan aktor dapat mengunggah data prediksi untuk dilakukan prediksi oleh sistem.

5.5.1.4 *Sequence Diagram* Kelola Data Akun



Gambar 5. 6 Sequence Diagram Kelola Data Pegawai

Keterangan :

Kelola data akun ini hanya dapat diakses oleh admin. Pertama,

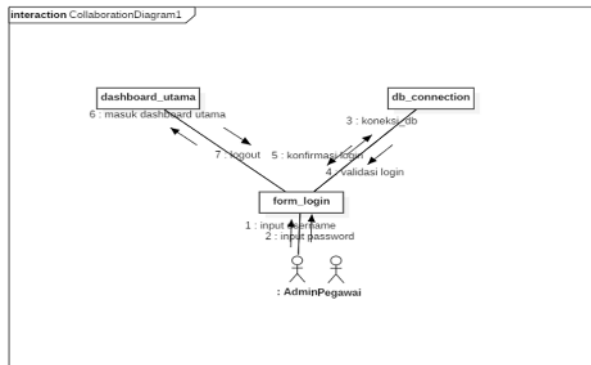
admin *login* terlebih dahulu, lalu pilih menu kelola akun pegawai dan akan masuk ke halaman kelola akun, admin dapat menginput, merubah, dan menghapus data di dalam kelola akun yang akan langsung terhubung masuk ke dalam *database*.

5.6.1 Collaboration Diagram

Collaboration diagram memperlihatkan interaksi dan hubungan terstruktur antar objek. Tipe *diagram* ini menekankan pada hubungan (*relationship*) antar objek, sedangkan *sequence diagram* menekankan ada urutan kejadian. Dalam satu *diagram* kolaborasi terdapat beberapa objek, *link*, dan *message*. *Diagram* kolaborasi digunakan sebagai alat untuk

menggambarkan interaksi yang mengungkapkan keputusan mengenai perilaku sistem.

5.6.1.1 Collaboration Diagram Login



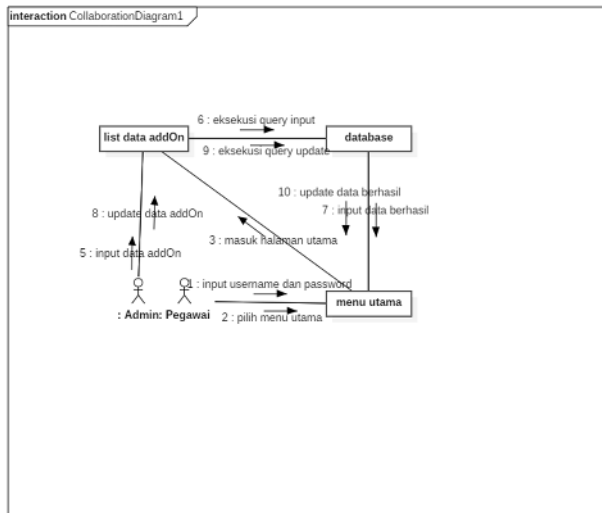
Gambar 5. 7 Collaboration Diagram Login

Keterangan :

Admin/pegawai melakukan login dengan menginput *username* dan *password*, memeriksa koneksi ke *database*, lalu validasi *username* dan *password*, apabila *username* dan *password* salah, maka sistem tidak akan

memberikan akses untuk masuk ke *dashboard* utama. Jika benar akan melakukan konfirmasi *login* dan akan masuk ke *dashboard* utama dari masing-masing aktor level yang *login*.

5.6.1.2 Collaboration Diagram Kelola Data Prospek Sales



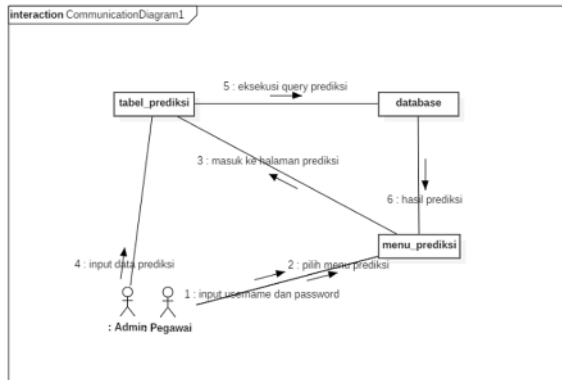
Gambar 5. 8 Collaboration Diagram Kelola Data AddOn

Keterangan :

Admin/pegawai dapat melakukan kelola data prospek sales, sebelumnya admin/pegawai akan *login* terlebih dahulu dan memilih menu utama, didalam menu utama *addOn* dapat mengelola data *addOn* untuk input data dan *update* data *addOn* jika terjadi perubahan jumlah data pelanggan, setiap perubahan pada sistem akan disimpan pada *database*.

5.6.1.3 Collaboration Diagram

Prediksi Pendapatan



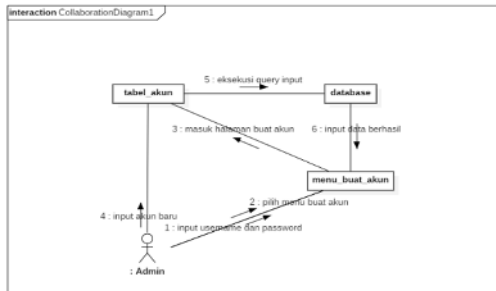
*Gambar 5. 9 Collaboration Diagram
Prediksi Pendapatan*

Keterangan :

Admin/pegawai dapat melakukan prediksi pendapatan dari produk *digital service* ini, dengan cara *login* terlebih dahulu untuk masuk ke *dashboard* utama, lalu memilih menu prediksi pendapatan, agar proses prediksi dapat dilakukan oleh sistem.

5.6.1.4 Collaboration Diagram

Kelola Data Akun



Gambar 5. 10 Collaboration Diagram Kelola Data Pegawai

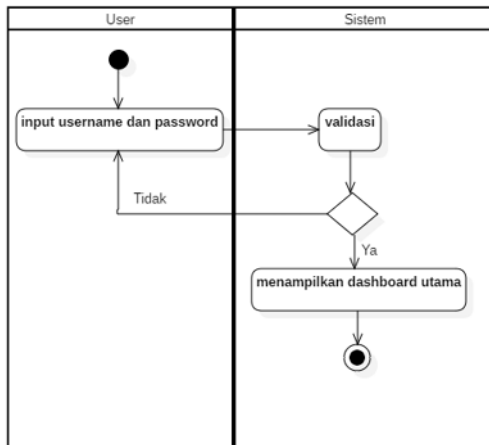
Keterangan :

Admin dapat melakukan kelola data akun, dengan cara *login* terlebih dahulu untuk masuk ke *dashboard* utama, lalu memilih menu kelola data akun agar dapat memberikan akses baru bagi pegawai yang ingin mengakses sistem prospek sales.

5.7.1 Activity Diagram

Activity diagram memodelkan aliran kerja atau *workflow* dari urutan aktivitas dalam suatu proses yang mengacu pada *use case diagram*. Adapun *activity diagram* pada aplikasi peramalan ini, sebagai berikut:

5.7.1.1 Activity Diagram Login

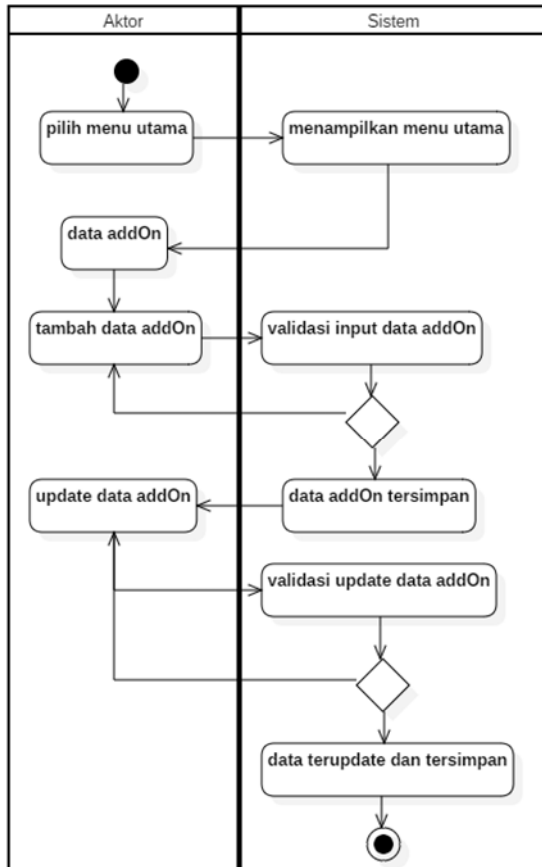


Gambar 5. 11 Activity Diagram Login

Keterangan :

Admin/pegawai menginput *username* dan *password* ke dalam sistem, dan sistem akan melakukan validasi *login*. Apabila valid maka akan menampilkan *dashboard* utama dari masing-masing user level, dan apabila tidak valid maka akan kembali ke *form login* untuk memasukkan *username* dan *password*.

5.7.1.2 Activity Diagram Kelola Data Prospek Sales

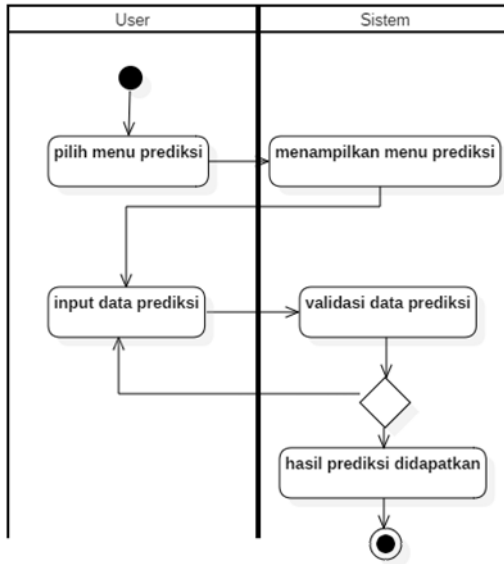


Gambar 5. 12 Activity Diagram Kelola Data AddOn

Keterangan :

Setelah admin/pegawai *login*, memilih menu utama, sistem akan menampilkan halaman utama yang berisi *list* data *addOn* produk *digital service*, admin/pegawai menambahkan data *addOn* lalu data akan di validasi, sistem akan mengeksekusi *query input*, lalu admin/pegawai akan meng-*update* data *addOn* jika terjadi penambahan data, setiap perubahan data akan tersimpan di *database*.

5.7.1.3 Activity Diagram Prediksi Pendapatan



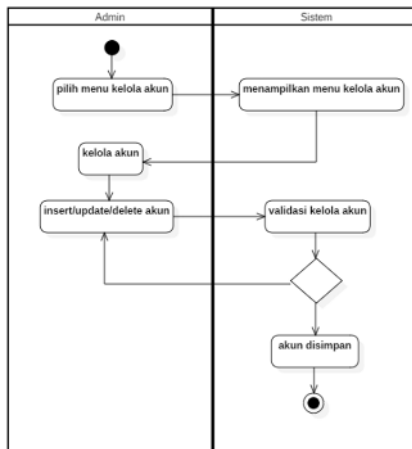
Gambar 5. 13 Activity Diagram Prediksi Pendapatan

Keterangan :

Data prediksi pendapatan dapat diakses oleh admin dan pegawai. Setelah aktor *login*, memilih menu prediksi pendapatan, sistem akan

menampilkan halaman prediksi pendapatan, aktor dapat mengunggah data prediksi pendapatan, lalu akan di validasi oleh sistem, jika benar akan menghasilkan prediksi yang telah diolah.

5.7.1.4 *Activity Diagram Kelola Data Akun*



Gambar 5. 14 Activity Diagram Kelola Data Pegawai

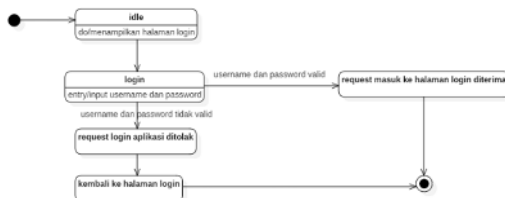
Keterangan :

Data kelola akun dapat diakses oleh admin/pegawai. Setelah aktor *login*, memilih menu kelola data akun, sistem akan menampilkan halaman data akun, aktor dapat mengelola data akun seperti menambahkan, mengubah, dan menghapus data yang berada di tabel data akun, data tersebut akan masuk di tabel data akun dan akan di validasi ,sistem akan mengeksekusi *query* tambah/*input*, edit/*update*, dan hapus/*delete*. Setelah aktor melakukan beberapa perintah yang dibutuhkan, data terbaru akan tersimpan di *database*.

5.8.1 Statechart Diagram

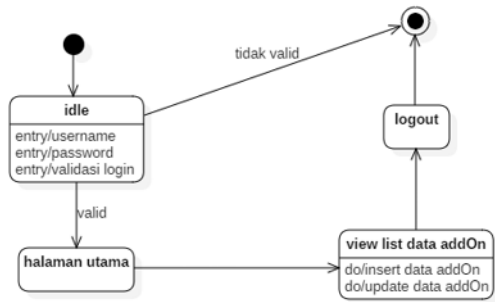
Statechart diagram adalah suatu *diagram* yang menggambarkan daur hidup (*behavior pattern*) dari sebuah objek, dari awal objek tersebut diinisialisasi sampai di hancurkan. Menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimulasi yang diterima.

5.10.1.1 Statechart Diagram Login



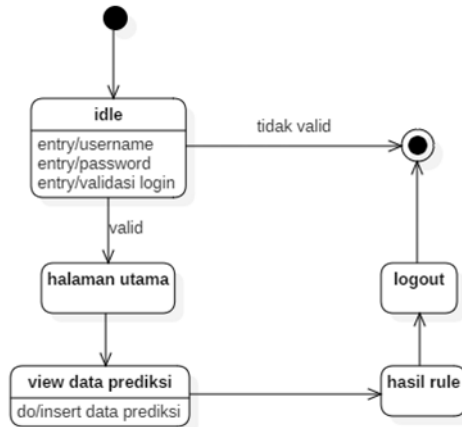
Gambar 5. 15 Statechart Diagram Login

5.10.1.2 Statechart Diagram Kelola Data Prospek



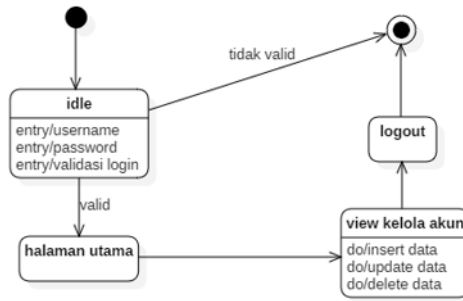
Gambar 5. 16 Statechart Diagram Kelola Data AddOn

5.10.1.3 Statechart Diagram Prediksi Pendapatan



Gambar 5. 17 Statechart Diagram Prediksi Pendapatan

5.10.1.4 Statechart Diagram Kelola Data Akun



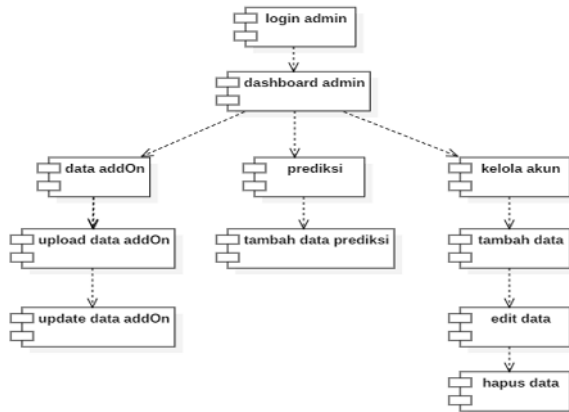
Gambar 5. 18 Statechart Diagram Kelola Data Pegawai

5.9.1 Component Diagram

Component diagram adalah *diagram* yang digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen *software* sistem. *Component diagram* berguna untuk memodelkan komponen objek. Adapun *component diagram* berguna untuk memodelkan komponen objek. *Component diagram* pada

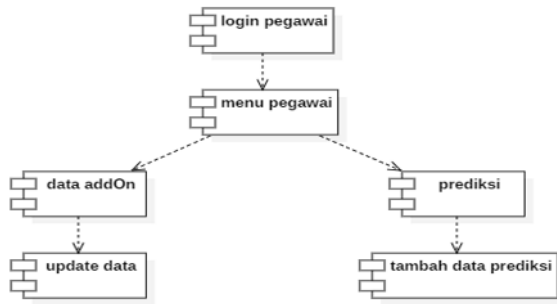
aplikasi peramalan ini, yaitu sebagai berikut:

5.9.1.1 *Component Diagram Admin*



Gambar 5. 19 Component Diagram Admin

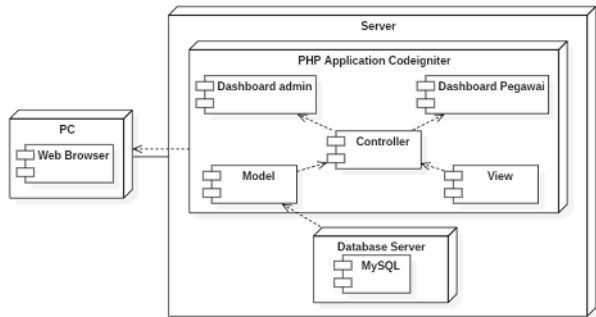
5.9.1.2 *Component Diagram Pegawai*



Gambar 5. 20 Component Diagram Pegawai

5.10.1 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal seperti sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*, sistem *client/server*, sistem terdistribusi murni, dan rekayasa ulang aplikasi.




Gambar 5. 21 Deployment Diagram

5.2 Perancangan *User Interface* Sistem

Setelah melakukan analisis, masuk pada tahap kedua dalam penerapan metode regresi linier berganda, yaitu tahap desain. Pada tahap ini penulis masih melakukan strategi diskusi dengan pembimbing untuk mengetahui bagaimana bentuk perancangan antarmuka yang digunakan pada sistem informasi prediksi pendapatan nantinya.

Rancangan *interface* adalah rancangan pembangunan dari komunikasi antar pemakai (aktor) dengan komputer. Antarmuka (*interface*) ini terdiri dari proses memasukkan data ke sistem dan menampilkan *output* informasi kepada aktor. Berikut beberapa gambar bentuk rancangan *user interface* yang sesuai dengan ruang lingkup penulis:

5.4.1 *Login Form*

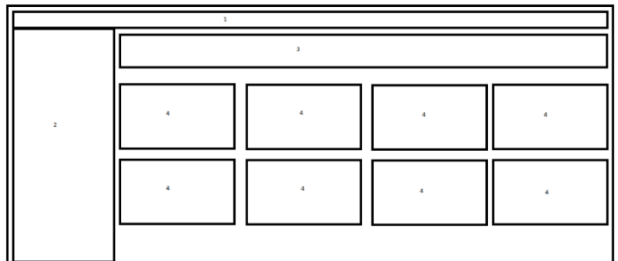


Gambar 5. 22 User Interface Login Form

Keterangan :

1. *Image : Profile Person*
2. *Label : Login here*
3. *Textbox 1 : Username*
4. *Textbox 2 : Password*
5. *Button 1 : Login*

5.4.2 Dashboard Utama

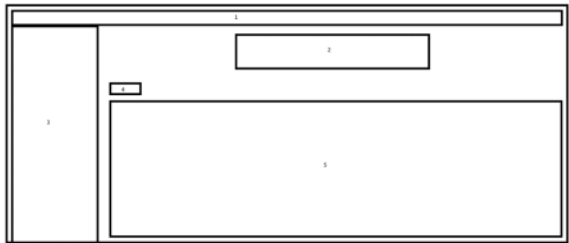


Gambar 5. 23 Halaman Dashboard Utama

Keterangan :

1. *Header*
2. *Label 2 : Judul Halaman*
3. *Side Bar*
4. *List Box*

5.4.3 Halaman Prediksi Pendapatan

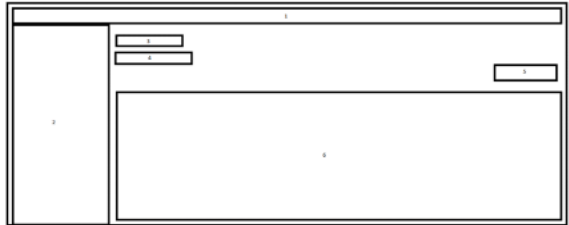


Gambar 5. 24 Halaman Prediksi Pendapatan

Keterangan :

1. *Header*
2. *Label 1 : Judul Halaman*
3. *Side Bar*
4. *Button 1 : Upload data*
5. *Table Data*

5.4.4 Halaman Kelola Data Akun



Gambar 5. 25 Halaman Kelola Data Pegawai

Keterangan :

1. *Header*
2. *Side Bar*
3. *Label 1 : Tampil data user*
4. *Button 1 : Tambah data pegawai*
5. *Textbox : Input Nomor Internet*
6. *Table Data*