

**PANDUAN LENGKAP MENGENAI APLIKASI
PROSPEK SALES**

PANDUAN LENGKAP MENGENAI APLIKASI PROSPEK SALES

Andri Fajar Sunandhar

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 aplikasi prospek sales yang sedang dibangun.

ACKNOWLEDGMENTS

ACRONYMS

SYMBOLS

INTRODUCTION

BAB I

1.1 Codeigniter

1.1.1 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 berusaha mempermudah penulisan kode-kode program PHP tersebut menjadi lebih singkat. 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* yang membantu perkembangan *Codeigniter* sehingga menjadi *framework* terpopuler di tahun 2006.

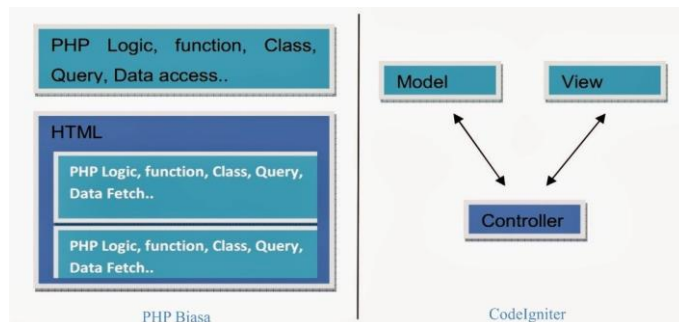


Gambar 1. Logo Codeigniter

1.1.2 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.

Untuk membuat website dinamis bisa menggunakan codeigniter karena menjadi salah satu framework php dengan konsep MVC (*Model, View, Controller*) yang dapat mempermudah developer dalam pembuatan aplikasi web. Selain ringan dan cepat, codeigniter juga memiliki dokumentasi yang lengkap disertai dengan contoh implementasi kodenya. Dengan dokumentasi yang lengkap akan menjadi salah satu alasan kuat mengapa banyak orang memilih *codeigniter* sebagai *framework* pilihannya. Berikut merupakan gambar perbandingan PHP biasa dengan Codeigniter.

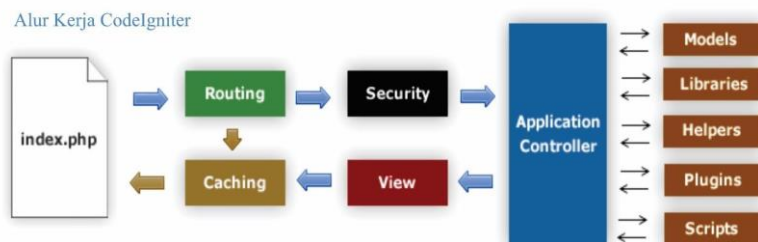


Gambar 2. Perbandingan PHP dengan Codeigniter

Konsep MVC memisahkan sistem berdasarkan komponen utama yang membangun sistem seperti manipulasi data, user interface, dan bagian pengontrol sistem. 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, berfungsi mengatur hubungan antara model dan *view* supaya bisa saling berkomunikasi, controller tersebut yang mengatur segala proses dalam aplikasi.
- c. *View*, merupakan bagian yang menangani *presentation*. Pada bagian *view* biasanya berupa file dengan template HTML, yang diatur oleh *controller*. *View* akan menerima kemudian mempresentasikan data kepada *user*. Pada bagian ini tidak memiliki akses langsung terhadap bagian model, untuk bisa mengakses ke bagian model maka harus *request* terlebih dahulu ke *controller*.

Alur kerja *framework* Codeigniter dapat digambarkan sebagai berikut:



Gambar 3. Alur Kerja *Framework* 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 *controller* di load secara keseluruhan, maka data yang disubmit oleh *user* dalam bentuk request HTTP akan di periksa terlebih dahulu melalui *security* yang dimiliki oleh *Codeigniter*.
- e. *Controller*: *controller* akan membuka file bagian model, *core libraries*, *helper* dan semua *resources* yang dibutuhkan dalam program tersebut.
- f. *View*: bagian terakhir akan dilakukan pengecekan semua program yang ada dalam *view* kemudian file akan 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.

1.1.3 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.

1.1.4 Keunggulan dan kekurangan Codeigniter

1.1.4.1 Keunggulan Codeigniter

- 1. Performa yang sangat cepat dibandingkan dengan framework yang lainnya.
- 2. Konfigurasi yang sangat minim, untuk menyesuaikan dengan database terlebih dahulu merubah beberapa konfigurasi pada bagian file *database.php* dan *autoload.php*.

3. Dokumentasi yang sangat lengkap, setiap paket pada instalasi codeigniter sudah disertai *user guide* dan bagus untuk dijadikan permulaan, serta bahasanya mudah untuk dipahami.

1.1.4.2 Kekurangan Codeigniter

1. Codeigniter dikembangkan oleh Ellis lab dan bukan oleh suatu komunitas, yang menyebabkan *update code engine*-nya tidak secepat *framework* lain.
2. Tidak direkomendasikan untuk pembuatan *web* dengan skala besar (*enterprise*) walaupun tersedia banyak *library*.
3. Masih menemukan banyak kekurangan dalam menambahkan file, sehingga banyak file yang tidak penting dengan mudah bisa tersimpan.

1.2 Framework

1.2.1 Pengertian Framework

Framework adalah kerangka kerja. *Framework* merupakan sekumpulan *script class* dan *function* yang dapat memudahkan *developer/programmer* dalam menangani suatu permasalahan dalam pembuatan aplikasi seperti pemanggilan *variable*, koneksi ke *database*, dll. sehingga *developer* bisa lebih fokus dan lebih cepat dalam membangun sebuah aplikasi.

Framework bisa disebut komponen pemrograman yang siap digunakan kembali kapan saja, sehingga *programmer* tidak harus membuat *script* yang sama secara berulang-ulang.

Secara sederhana dapat dijelaskan bahwa *framework* merupakan sekumpulan fungsi (*libraries*), maka seorang *programmer* tidak perlu lagi membuat fungsi-fungsi (biasanya disebut kumpulan *Library*) dari awal, *programmer* tinggal memanggil kumpulan *library* atau fungsi yang sudah ada didalam *framework*, tentunya cara menggunakan fungsi-fungsi itu sudah ditentukan oleh *framework*. Berikut merupakan beberapa contoh fungsi standar *framework* diantaranya fungsi

grafik, tabel bergaya zebra, validasi, *upload*, *captcha*, proteksi terhadap XSS (XSS filtering), *template*, kompresi, XML, *paging*, enkripsi, email, SEO, *session*, *security*, kalender, bahasa, manipulasi gambar, dan lain-lain.

1.2.2 Fungsi *Framework*

1. Dapat membantu kerja *developer* dalam membangun aplikasi sehingga aplikasi bisa diselesaikan dalam waktu yang singkat.
2. Penerapan *design patterns* memudahkan dalam rancangan, pengembangan dan pemeliharaan sistem.
3. *Stability* dan *Reability* yang dibangun didalam sistem lebih stabil dan handal karena berbasis pada *framework* yang sudah teruji stabilitas dan keandalannya.
4. *Coding Style* konsisten, memudahkan dalam membaca kode dalam menemukan *bugs*.
5. *Security Concern* dan *framework* akan mengantisipasi dengan memasang perisai terhadap adanya berbagai masalah keamanan yang mungkin terjadi.
6. Dokumentasi, *framework* dapat mendisiplinkan kita untuk menulis dokumen apa yang kita tulis.

1.2.3 Kelebihan dan Kelemahan *Framework*

1.2.3.1 Kelebihan *Framework*

1. Lebih cepat dan efisien
Jika mengerjakan proyek besar, maka penggunaan *framework* dapat membantu mempercepat proses pengembangan. Pada umumnya, *framework* memiliki beragam fungsi dan plugin yang bisa dimanfaatkan. Dalam kerangka kerja ini, maka proses pengembangan proyek jauh lebih cepat daripada harus menulis kode dari awal. Selain itu, tidak perlu menulis berulang-ulang untuk kode yang bersifat repetitif.
2. Menghemat biaya

Sebagian besar bersifat open source dan gratis untuk bisa digunakan. Biaya yang harus dikeluarkan oleh client juga akan menjadi lebih kecil karena proses pengerjaan yang lebih simple dan lebih cepat.

3. Memperhatikan faktor keamanan

Framework telah banyak digunakan oleh *developer*, dan kemungkinan adanya masalah keamanan maupun bug yang telah diperbaiki. Selain itu, *framework* biasanya memiliki komunitas dalam jumlah cukup banyak yang dapat berperan dalam jangka panjang. Setiap kali pengguna menemukan celah keamanan, maka mereka dapat memberi tahu tim untuk segera memperbaikinya.

1.2.3.2 Kelemahan *Framework*

1. Kurangnya pemahaman bahasa pemrograman

Jika bekerja menggunakan kerangka kerja kemudian hanya mengetahui sedikit tentang bahasa pemrograman yang digunakan, maka *developer* hanya mempelajari mengenai kerangka tersebut. Sehingga pemahaman mengenai bahasa pemrograman menjadi tidak berkembang.

2. Memiliki batasan

Dalam penggunaannya, kerangka ini juga memiliki beberapa batasan yang tidak dapat Anda modifikasi. Sehingga Anda harus bekerja sesuai dengan standar yang digunakan di dalamnya. Oleh karena itu, ketika mengembangkan sebuah aplikasi, maka harus menggunakan kerangka kerja yang sesuai dengan kebutuhan *developer*.

3. Kode Publik

Karena bersifat publik, maka kode dapat digunakan siapa saja termasuk pihak-pihak yang mempunyai niat buruk.

1.3 *Object Oriented Programming* (OOP)

1.3.1 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.

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 bawaan dari C++ yaitu C# (C-Sharp), kemudian disusul dengan VB.Net dengan fitur OOP yang merupakan penyempurnaan dari bahasa VB 0.6 yang tidak mendukung fitur OOP.

1.3.2 Pengertian OOP

Object Oriented Programming (OOP) merupakan paradigma dalam melakukan pemrograman yang berorientasi objek, sehingga pengolahan data disatukan dalam *class* dan *object*.

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.

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

1.3.3 Jenis-jenis OOP pada bahasa pemrograman

Konsep OOP dibagi menjadi 3 jenis, yaitu :

- a. Bahasa OOP murni, merupakan sebuah bahasa yang mengharuskan program ditulis hanya berupa objek saja. Contoh – Eiffel, Smalltalk, 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*.

1.3.4 Konsep OOP

- 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 dari *class* yang berbeda untuk menyediakan penggunaan yang berbedabagi interface (publik) yang sama.

1.3.5 Keunggulan dan kekurangan OOP

1.3.5.1 Keunggulan OOP

- 1. OOP menyediakan struktur yang sangat jelas untuk program, sehingga OOP sangat bagus digunakan untuk mendefinisikan tipe data.
- 2. OOP akan mempermudah dalam melakukan *maintenance* dan memodifikasi kode yang sudah ada. Objek yang baru dapat dibuat tanpa harus mengubah kode yang sudah ada.
- 3. OOP menyediakan sebuah *framework* dimana komponen *software* yang tersedia dalam *framework tersebut* dapat dengan mudah diadaptasi dan dimodifikasi oleh programmer. Hal ini sering digunakan dalam pengembangan *Graphical User Interfaces* (GUI).

1.3.5.2 Kekurangan OOP

- 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.

1.4 *Hypertext Preprocessor (PHP)*

1.4.1 Sejarah PHP

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

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

Dalam perilsan kode sumber ini menjadi *open source*, maka dari itu banyak *programmer* yang tertarik untuk 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)*. Selanjutnya *interpreter* PHP sudah diimplementasikan dalam program C. Dalam perilsan ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP / FI secara signifikan. PHP versi 2.0 ini telah menarik banyak perahitian *programmer*, namun bahasa ini memiliki masalah dengan kestabilan yang kurang bisa diandalkan. Hal ini dikarenakan Lerdorf hanya bekerja sendiri untuk mengembangkan PHP.

Pada saat itu Andi Gutmans dan Zeev Suraski, mengambil bagian dan membuat ulang *parsing engine* yang menjadi dasar dari PHP supaya 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*. Selanjutnya The PHP Group yang merupakan kumpulan programmer dari seluruh dunia bersatu untuk mengembangkan PHP.

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

PHP 4.0 adalah versi PHP yang paling banyak digunakan pada awal abad ke-21. Versi ini banyak digunakan dikarenakan kemampuannya untuk membangun aplikasi *web* sangat kompleks, akan tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, *Zend* kembali merilis PHP 5.0. Dalam versi 5.0 ini, mengalami perubahan yang cukup besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk mengetahui perkembangannya. Beberapa penambahan fitur meliputi *PHP Data Objects* (PDO) untuk mengakses *database*, *closures*, *trait*, dan *namespaces*.

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.

1.4.2 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 secara bersamaan dengan bahasa pemrograman lainnya seperti bahasa pemrograman HTML, dan Javascript.

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

1.4.3 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 tertulis dalam pembatas yang telah ditentukan oleh 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 yaitu untuk memisahkan kode PHP dari kode PHP lainnya, seperti HTML, dan Javascript.

b. Variabel

Variabel dalam PHP diawali dengan simbol dolar “\$”. Pada PHP versi 5.0 diperkenalkan jenis isyarat yang memungkinkan fungsi untuk 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.

1.4.4 Keunggulan dan kekurangan PHP

1.4.4.1 Keunggulan PHP

1. Bahasa pemrograman PHP merupakan *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. Dari sisi pemahamannya, PHP merupakan bahasa *scripting* yang sangat mudah digunakan karena memiliki referensi yang cukup banyak.
5. PHP merupakan bahasa pemrograman yang bersifat *open source* (Gratis) yang bisa digunakan di berbagai sistem operasi seperti Windows, Linux, Unix, Macintosh dan dapat dijalankan secara *runtime* melalui *console* serta dapat menjalankan perintah-perintah sistem.

1.4.4.2 Kekurangan PHP

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 memiliki kelemahan dalam hal keamanan tertentu, dimana *programmer* sering tidak teliti dalam melakukan pemrograman dan kurang memperhatikan isu dan konfigurasi PHP.

1.5 Google Maps API

1.5.1 Pengertian Google Maps API

Google Maps API merupakan pengembangan teknologi dari google yang digunakan untuk menanamkan Google Map di suatu aplikasi yang tidak dibuat oleh Google. Google Maps API merupakan suatu library dengan bentuk *javascript* yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai dengan kebutuhan. Dalam perkembangannya Google Maps API diberikan kemampuan untuk mengambil gambar peta statis. Melakukan geocoding, dan memberikan penuntun arah. Google Maps API bersifat gratis untuk publik.

Penggunaan Google Maps API pada pengembangan aplikasi android dengan menggunakan Eclipse dan komputer menggunakan sistem operasi windows. Google Maps API terbagi menjadi 4 platform yaitu *web*, *Webservice*, *Android*, dan *iOS*.

1.5.2 Platform Google Maps API

1. Google Maps API for Android

- Google Maps Android API

<https://developers.google.com/maps/documentation/android-api/>

API ini digunakan untuk aplikasi yang menampilkan peta seperti GIS di Android. Fitur yang disediakan yaitu 3D Building, Custom Map, Custom Marker, Integrasi dengan webservice pihak ketiga dan yang lainnya. Intinya jika dalam pembuatan aplikasi GIS seperti peta masjid atau sebaran penduduk di Suatu daerah, maka inilah APInya. Bahasa yang dipakai adalah JAVA.

- Google Place API for Android

<https://developers.google.com/places/android-api/>

Digunakan untuk aplikasi yang membutuhkan fitur mendeteksi lokasi disekitar user di Android dengan

bahas JAVA. Misalkan daftar restoran terdekat atau pom bensin terdekat.

2. Google Maps API for iOS

API disini digunakan untuk membuat Aplikasi iPad atau iPhone

- Google Maps SDK for iOS

<https://developers.google.com/maps/documentation/ios-sdk/>

Sama dengan Google Maps Android API yang membedakan yaitu API ini untuk Platform Apple. Bahasa yang digunakan adalah Objective-C.

- Google Place API for iOS

<https://developers.google.com/places/ios-api/>

Sama dengan Place API for Android yang membedakan API ini untuk Platform Apple. Bahasa yang digunakan adalah Objective-C.

3. Google Maps API for Web

Kumpulan API ini digunakan untuk membuat aplikasi WEB seperti GIS dengan bahasa pemrograman *Javascript* dan HTML. API untuk web ini dapat dilihat dokumentasinya di

<https://developers.google.com/maps/web/>.

API untuk web ini terbagi menjadi :

- Google Map Javascript API, digunakan untuk menampilkan Peta di webApp atau website yang dibuat dengan custom UI, Marker, Infowindows dan integrasi.
- Google Maps Embedded API, digunakan untuk menampilkan peta suatu lokasi tanpa menggunakan bahasa pemrograman, peta ini hampir sama dengan menampilkan peta Google di website yang dibuat secara interaktif.
- Google Street View Image API biasa digunakan untuk menampilkan lokasi jalan dari StreetView.

API ini sifatnya interaktif, sangat cocok untuk menampilkan lokasi dengan sudut pandang 360 derajat.

- Google Static Maps API ini biasa digunakan untuk menampilkan Peta Static sebuah lokasi. API ini hampir mirip dengan Embedded API, hanya yang membedakan peta nya tidak interaktif. (sudah dalam format Gambar).
- Google Place Javascript API, digunakan untuk melisting lokasi point of interest seperti hotel, restoran atau lokasi lokasi lain yang terdata di Google map, API ini fungsinya hampir sama dengan Google Place API for Android dan Google Place API for IOS.

4. Google Maps Webservice API

API ini merupakan layanan yang outputnya berupa JSON. Artinya sembarang bahasa pemrograman atau platform bisa mengaksesnya. Webservice lebih fokus dalam memberikan data dengan format outputnya diserahkan ke programmer. Jika akan membuat aplikasi Canggih berbasis Google Maps maka wajib di pelajari di

<https://developers.google.com/maps/web-services/>.

Berikut ini adalah komponennya :

- Google Maps GeoCoding API ini berfungsi untuk mengkonversi koordinat menjadi alamat jalan atau lokasi menjadi koordinat (reverse GeoCoding).
- Google Place Webservice API ini memiliki fungsi yang sama dengan Google Place API, yang membedakan yaitu webservice ini lebih luas pemakaiannya dan tidak dibatasi oleh bahasa pemrograman java, Objective-C atau javascript.

- Google Maps Elevation API ini memiliki fungsi menampilkan ketinggian suatu lokasi dari atas permukaan laut. Misalkan kota Yogyakarta ketinggian dari permukaan laut 10 Meter, sedangkan jika lokasi koordinatnya adalah puncak gunung merapi, ketinggiannya 3000 Meter.
- Google Maps Road API ini memiliki fungsi sangat spesifik, yaitu memastikan Programmer GPS tracker menampilkan log perjalanan persis di jalan yang dilalui. Biasanya dipakai perusahaan GPS tracking atau asset Tracking.
- Google Map GeoLocation API ini memiliki fungsi yang sangat canggih, dikarenakan bisa mendeteksi lokasi user walaupun GPS di smartphone dimatikan. GeoLocation API memanfaatkan sinyal wifi untuk mendeteksi/melacak keberadaan lokasi user. Artinya perangkat yang tidak memiliki GPS masih bisa dideteksi lokasinya walaupun tidak seakurat sensor GPS.
- Google Maps Direction API. Dipakai untuk menunjukkan jalur perjalanan dari lokasi A ke lokasi B. Biasanya lokasi A adalah lokasi User saat ini dan lokasi adalah Point of interest yang dituju.
- Google Maps Timezone API digunakan untuk mendeteksi Timezone suatu koordinat/lokasi. Misalkan input Jakarta, maka timezone GMT+7.
- Google Maps Distance Matrix API ini sangat cocok untuk mengkalkulasi jarak dan waktu tempuh ke suatu lokasi. API ini masih berhubungan dengan Direction API.

1.5.3 Kelebihan dan kekurangan Google Maps API

1.5.3.1 Kelebihan Google Maps API

1. Dukungan penuh yang dilakukan google sehingga terjamin dan bervariasi fitur yang ada pada Google Maps API
2. Banyak pengembang yang menggunakan Google Maps API sehingga mudah dalam mencari referensi dalam pengembangan aplikasi

1.5.3.2 Kekurangan Google Maps API

1. Jika ingin mengakses maka harus terkoneksi ke internet pada perangkat yang digunakan.

1.6 XAMPP

1.6.1 Sejarah XAMPP

XAMPP adalah pengembangan dari LAMP (*Linux Apache, MySQL, PHP and PERL*). XAMPP didirikan oleh Kai 'Oswald' Seidler dan Kay Vogelgesang pada tahun 2002. XAMPP tersebut merupakan salah satu project non-profit, project ini bertujuan untuk mempromosikan Apache web server kepada publik.

1.6.2 Pengertian XAMPP

XAMPP yaitu perangkat lunak yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. XAMPP merupakan tools yang menyediakan paket perangkat lunak. Dengan menginstall XAMPP maka tidak perlu melakukan kembali instalasi, konfigurasi web server Apache seperti PHP dan MySQL secara manual. XAMPP akan menginstallasi dan mengkonfigurasikannya secara otomatis.

XAMPP merupakan salah satu paket instalasi Apache, PHP dan MySQL yang dapat digunakan untuk membantu proses instalasi ketiga aplikasi tersebut. Selain paket instalasi instant XAMPP versi 1.6.4 juga memberikan fasilitas pilihan penggunaan PHP v4.0 atau PHP v5.0.

1.6.3 Penjelasan singkatan XAMPP

- **X** : Program ini bisa dijalankan di banyak sistem operasi seperti Windows, Linux, Mac OS, dan Solaris.
- **A** : Apache, server aplikasi Web.
- **M** : MySQL, merupakan aplikasi database yang berfungsi untuk mengolah/menyimpan data.
- **P** : PHP, merupakan bahasa pemrograman web.
- **P** : Perl, merupakan bahasa pemrograman untuk semua tujuan.

1.6.4 Fitur-fitur XAMPP

Berikut ini merupakan beberapa fitur-fitur dari xampp :

1. *Apache*
Apache dapat diartikan sebagai perangkat lunak sumber terbuka yang menjadi alternatif dari server web Netscape.
2. *MySQL*
MySQL atau sering disebut “My Structured Query Language”. MySQL ini berjalan sebagai server yang menyediakan multi-user untuk mengakses ke sejumlah database.
3. *PHP*
Bahasa pemrograman *script* sisi server yang didesain untuk pengembangan *web*

1.6.5 Bagian-bagian Xampp

Berikut ini adalah merupakan bagian-bagian dari XAMPP :

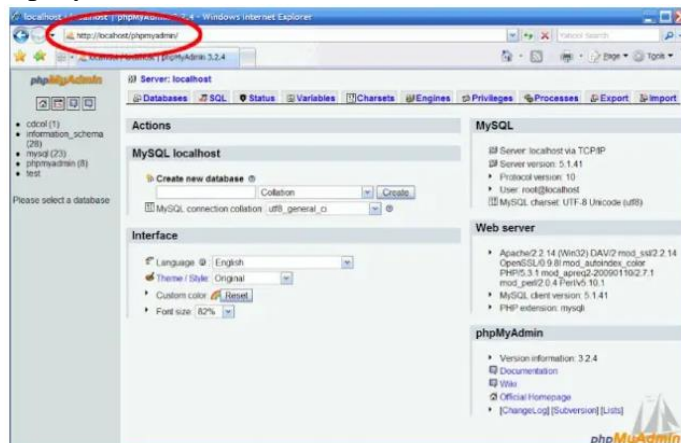
1. Htdocs

Name	Date modified	Type	Size
anonymous	15/12/2013 13:52	File folder	
apache	15/12/2013 13:52	File folder	
cgi-bin	15/12/2013 13:56	File folder	
contrib	15/12/2013 13:52	File folder	
FileZillaFTP	15/12/2013 13:56	File folder	
htdocs	15/12/2013 16:08	File folder	
img	15/12/2013 13:52	File folder	
install	15/12/2013 13:56	File folder	
licenses	15/12/2013 13:52	File folder	

Gambar x. Htdocs

Htdocs merupakan sebuah folder yang digunakan untuk menyimpan aplikasi yang akan dibuat.

2. PhpMyadmin



Gambar x. phpMyadmin

phpMyadmin merupakan sebuah tempat yang digunakan untuk mengelola database MySQL yang telah di install pada komputer atau laptop. Untuk mengakses phpMyadmin dengan mengetikan <http://localhost/phpMyadmin> pada browser maka akan muncul tampilannya seperti gambar x.

3. Control Panel



Gambar x. Control Panel

Control Panel merupakan sebuah layanan untuk mengelola XAMPP baik itu mengontrol (start atau stop XAMPP) serta layanan service lainnya.

1.6.6 Kelebihan dan Kekurangan XAMPP

Berikut ini terdapat beberapa kelebihan dan kekurangan pada XAMPP, yaitu :

1.6.7.1 Kelebihan Xampp

1. Database Storage Engine ini banyak digunakan oleh programmer karena sifatnya free (gratis).
2. Kemampuannya mempunyai kapasitas yang cukup mumpuni yaitu sekitar 60.000 tabel dengan jumlah record mencapai 5.000.000.000 bahkan untuk yang terbaru sudah lebih dari itu.
3. Keamanan penyimpanan data pada xampp sudah terbilang cukup aman.
4. Kelebihan paling utama dari xampp ini yaitu kecepatannya.

1.6.7.2 Kekurangan Xampp

1. Tidak cocok untuk menangani data dalam jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data.
2. Memiliki keterbatasan dalam kemampuan kinerja pada server ketika data yang disimpan telah melebihi batas maksimal kemampuan.

1.7 Metode *Prototype*

BAB II

INSTALASI APLIKASI YANG AKAN DIGUNAKAN

Alat yang dibutuhkan untuk membangun sistem pemetaan ini diantaranya :

- *Xampp*
- *Sublime text*
- *Codeigniter*

1. Xampp

Xampp merupakan 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*.

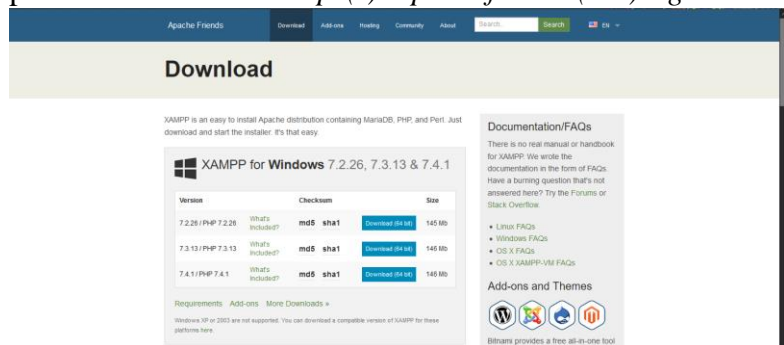
P = PHP, huruf “P” yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa

pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat *website dinamis*.

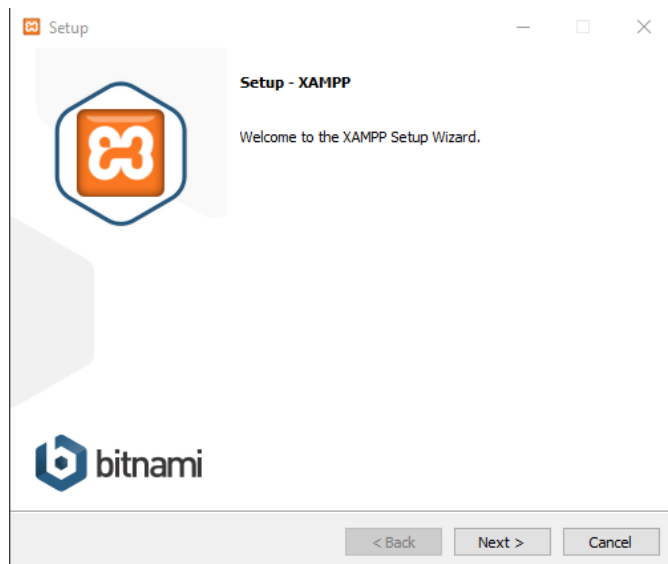
P = Perl, untuk huruf “P” selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi, sehingga sangat fleksibel dan banyak digunakan.

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

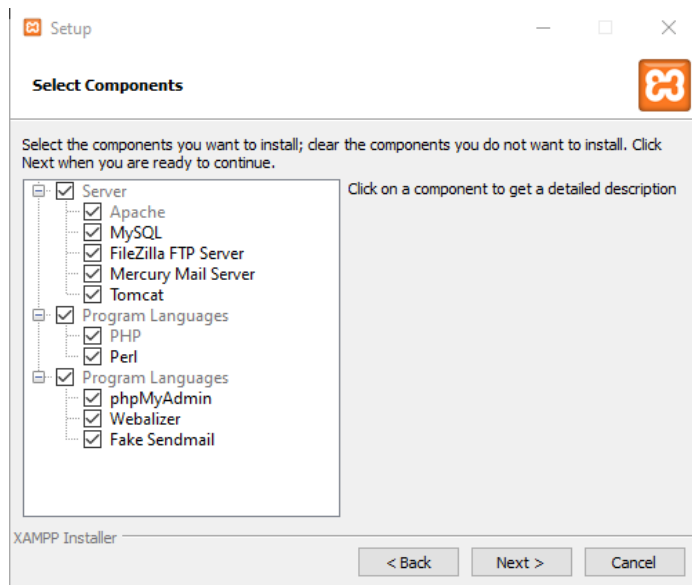
- a. Pertama, *download* terlebih dahulu file installer xampp pada link berikut ini. [https://apachefriends\(dot\)org](https://apachefriends(dot)org)



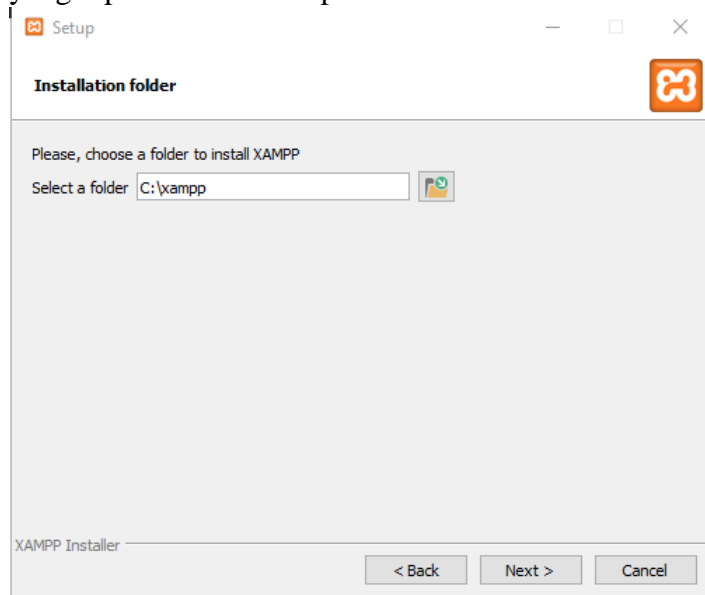
- b. Pilih file installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Klik file xampp yang sudah berhasil di *download*, maka akan muncul tampilan seperti ini.



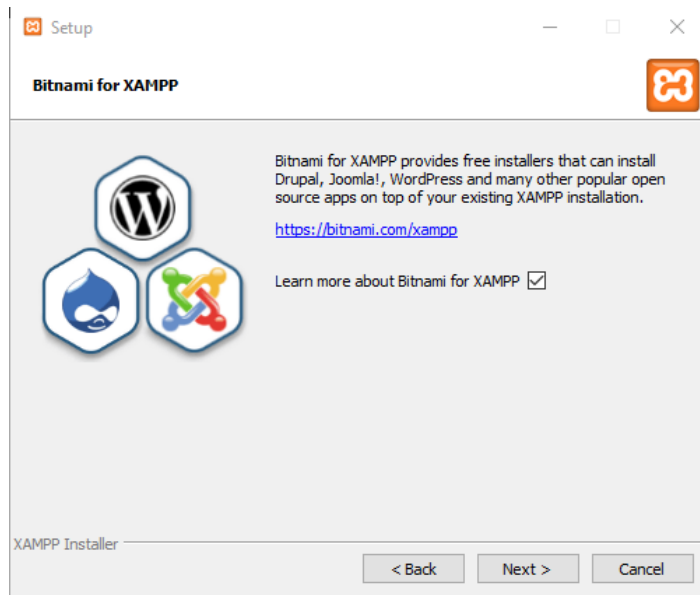
- d. Kemudian, pilih next untuk melanjutkan pada proses instalasi.



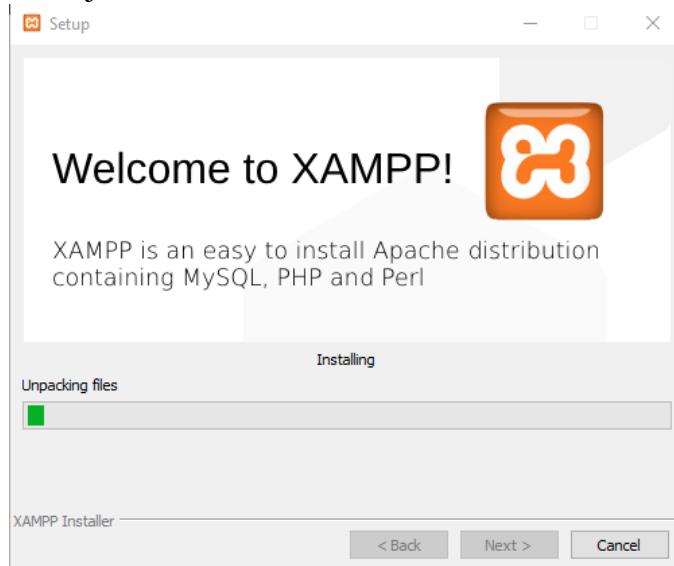
- e. Pada tampilan *select component*, centang semua pilihannya agar dapat menjalankan semua program yang dipilih. Kemudian pilih *next*.



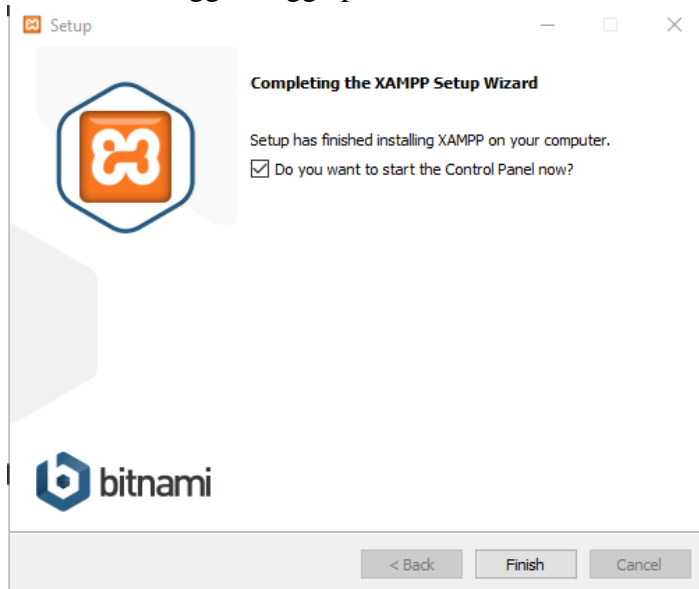
- f. Pada, *installation folder* secara *default* sistem akan menyimpannya pada direktori C. Jika sudah klik *next* untuk melanjutkan proses instalasi.



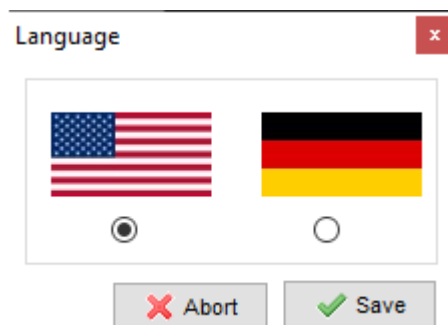
- g. Pada tampilan ini langsung saja klik *next* untuk melanjutkan.



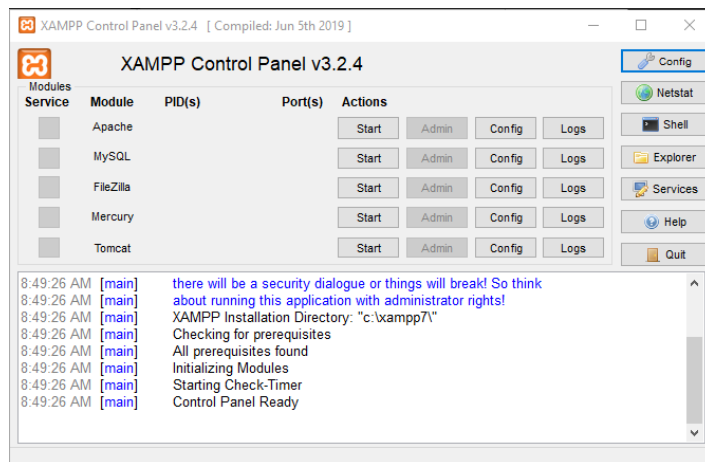
- h. Kemudian tunggu hingga proses instalasi selesai.



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



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



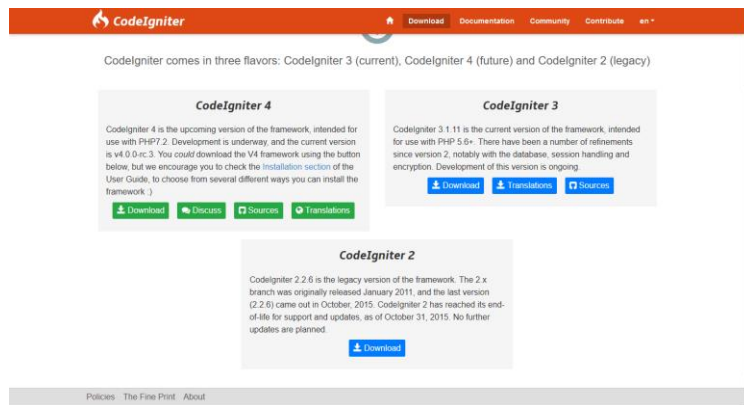
k. Jika berhasil, maka tampilan utama *control panel* dari xampp seperti berikut.

2. Codeigniter

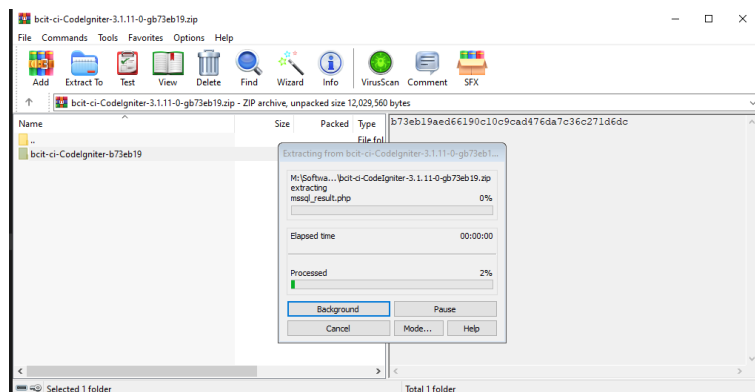
Codeigniter (CI) merupakan sebuah *web application framework* yang bersifat *open source* dimana digunakan untuk membangun aplikasi php dinamis. *Framework* itu sendiri merupakan abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh kode yang dibuat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

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

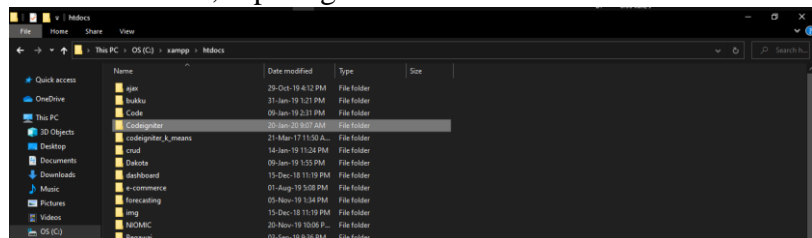
- a. Pertama, *download* terlebih dahulu codeigniter pada *website* resminya pada link berikut. <https://codeigniter.com/download>.



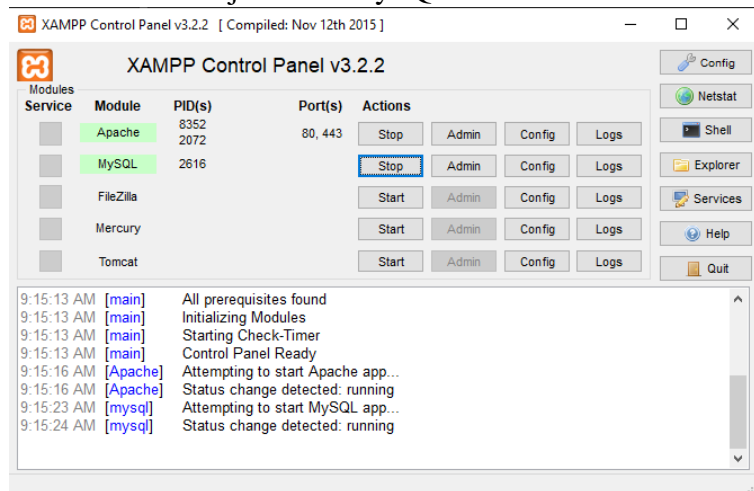
- b. Setelah berhasil di *download*, selanjutnya ekstrak file tersebut.



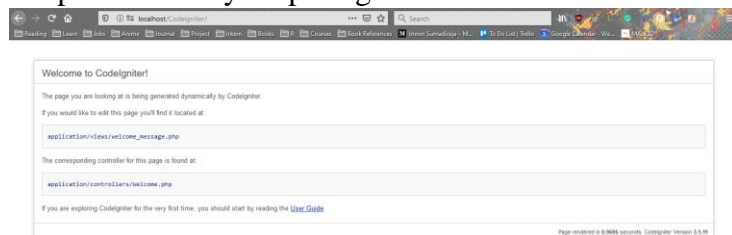
- c. Kemudian, *copy* file yang telah di ekstrak ke dalam direktori htdocs, seperti gambar berikut.



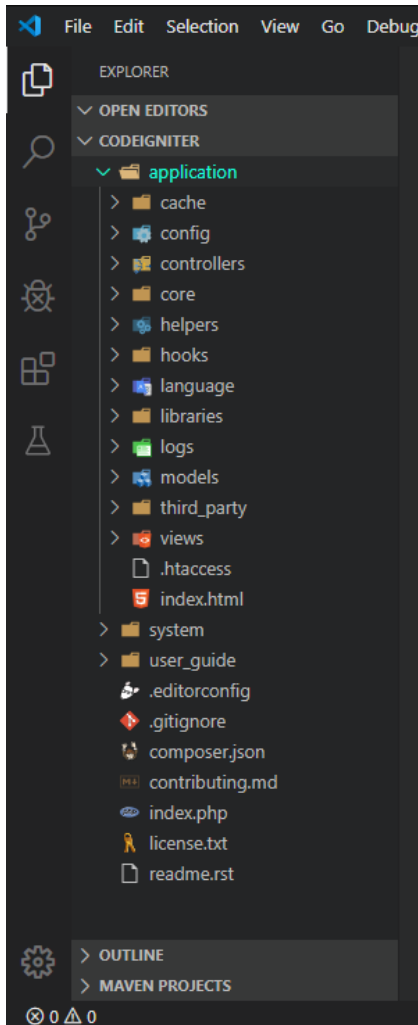
- d. Untuk menjalankannya, kita menggunakan xampp, untuk itu silakan jalankan terlebih dahulu aplikasi xampp-nya seperti berikut.
- e. Buka aplikasi xamppnya, kemudian klik tombol start pada kolom actions baris pertama untuk menjalankan apache, dan klik tombol start pada kolom actions baris kedua untuk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

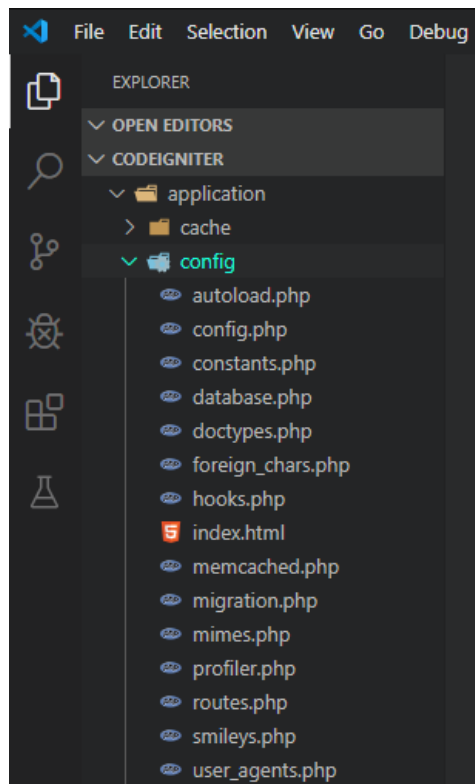


- a) *Application*, merupakan folder yang pada dasarnya menyimpan aplikasi yang sedang kita buat.
- b) *Cache*, merupakan folder yang menyimpan semua cache yang dibuat oleh cache library.

- 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 salah satu dari konsep MVC yang dapat 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.

3. 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-LOADING
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 minimal 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 |
21 | These are the things you can load automatically:
22 |
23 | 1. Packages
24 | 2. Libraries
25 | 3. Drivers
26 | 4. Helper files
27 | 5. Custom config files
```

- b. Kemudian temukan kode berikut.

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

- c. Ubah kode tersebut menjadi seperti berikut.

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

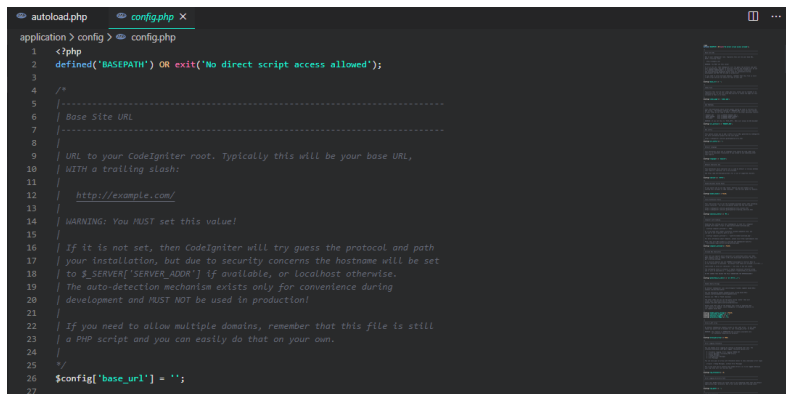
Kode tersebut dapat diartikan kita dapat meload library 'database' secara otomatis.

- 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.



```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 /*
5 |-----
6 | Base Site URL
7 |-----
8 |
9 | URL to your CodeIgniter root. Typically this will be your base URL,
10 | WITH a trailing slash:
11 |
12 | http://example.com/
13 |
14 | WARNING: You MUST set this value!
15 |
16 | If it is not set, then CodeIgniter will try guess the protocol and path
17 | your installation, but due to security concerns the hostname will be set
18 | to $_SERVER['SERVER_ADDR'] 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 easily do that on your own.
24 |
25 | */
26 $config['base_url'] = '';
```

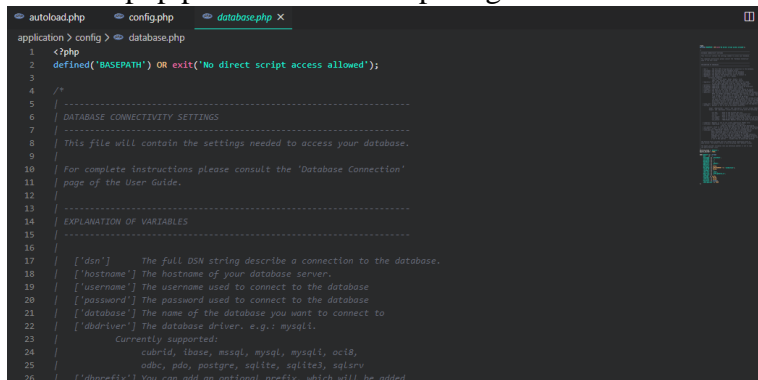
f. Kemudian temukan kode berikut.

```
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, merupakan salah satu file yang berkaitan dengan koneksi ke database. Adapun konfigurasi yang perlu diperhatikan, diantaranya: hostname, username, password dan database. Buka file database.php pada teks editor seperti gambar berikut.



```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 /*
5 |-----
6 | DATABASE CONNECTIVITY SETTINGS
7 |-----
8 | This file will contain the settings needed to access your database.
9 |
10 | For complete instructions please consult the 'Database Connection'
11 | page of the User Guide.
12 |
13 |-----
14 | EXPLANATION OF VARIABLES
15 |-----
16 |
17 | ['dsn'] The full DSN string describe a connection to the database.
18 | ['hostname'] The hostname of your database server.
19 | ['username'] The username used to connect to the database
20 | ['password'] The password used to connect to the database
21 | ['database'] The name of the database you want to connect to
22 | ['driver'] The database driver. e.g.: mysql.
23 |
24 | Currently supported:
25 | cubrid, ibase, mssql, mysql, mysqli, oci8,
26 | odbc, pdo, postgres, sqlite, sqldr, sqlsrv
```

- 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.

```

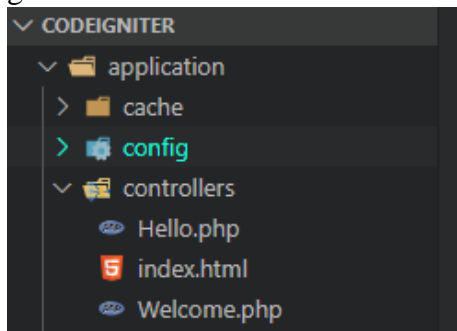
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost', //hostname
79     'username' => 'root', //username
80     'password' => '', //password
81     'database' => 'database_name', //nama database
82     'dbdriver' =>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

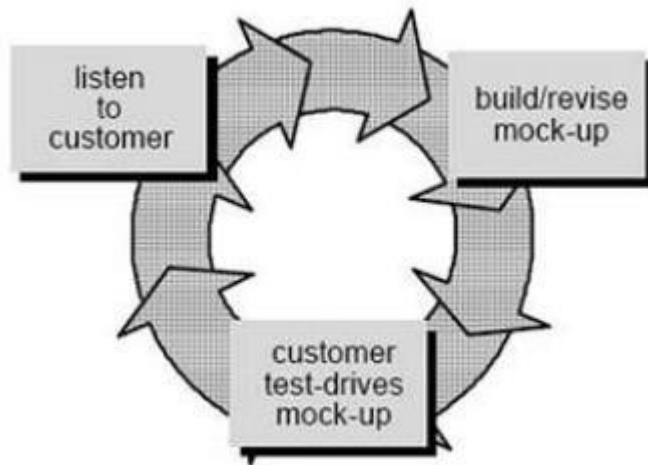
METODELOGI PENELITIAN

4.1 Diagram Alur Metodologi Penelitian

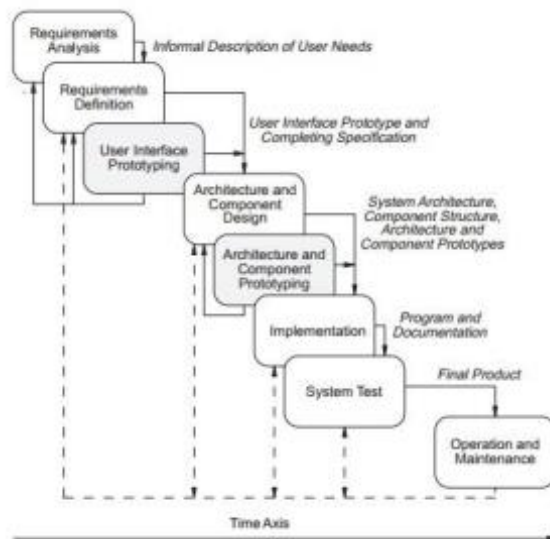
Metodologi penelitian adalah sekumpulan peraturan, kegiatan, dan prosedur yang digunakan oleh pelaku suatu disiplin. Metodologi juga merupakan analisis teoritis mengenai suatu cara atau metode. Penelitian merupakan suatu penyelidikan yang sistematis untuk meningkatkan sejumlah pengetahuan, juga merupakan suatu usaha yang sistematis dan terorganisasi untuk menyelidiki masalah tertentu yang memerlukan jawaban[28]. Hakikat penelitian dapat dipahami dengan mempelajari berbagai aspek yang mendorong penelitian untuk melakukan penelitian. Setiap orang mempunyai motivasi yang berbeda, di antaranya dipengaruhi oleh tujuan dan profesi masing-masing. Motivasi dan tujuan penelitian secara umum pada dasarnya adalah sama, yaitu bahwa penelitian merupakan refleksi dari keinginan manusia yang selalu berusaha untuk mengetahui sesuatu. Keinginan untuk memperoleh dan mengembangkan pengetahuan merupakan kebutuhan dasar manusia yang umumnya menjadi motivasi untuk melakukan penelitian[29]. Dalam penelitian ini

penulis menggunakan metode pengembang sistem yaitu metode *prototype*.

Metode pengembangan sistem yang digunakan adalah dengan menggunakan metode *prototype*. Metode *prototype* digunakan untuk menjelaskan kebutuhan pengguna secara lebih rinci karena pengguna sering mengalami kesulitan dalam penyampaian kebutuhan secara detail tanpa melihat gambaran yang jelas. Untuk mengantisipasi agar proyek aplikasi dapat berjalan sesuai dengan rencana dan tepat waktu, maka sebaiknya spesifikasi kebutuhan sistem harus sudah disepakati terlebih dahulu oleh pengembang dan pengguna. Proses untuk menghasilkan *prototype* disebut *Prototyping*, berikut adalah tahapan metode *prototyping* [17] :



Gambar 4.1 menjelaskan bahwa metode prototyping dimulai dengan mendengarkan kebutuhan dan masukan dari pengguna. Pengembang dan pengguna bertemu dan bersama-sama menentukan tujuan keseluruhan untuk perangkat lunak dan mengidentifikasi apapun persyaratan yang diperlukan. Lalu pengembang membuat sebuah gambaran tentang aplikasi yang selanjutnya dapat dipresentasikan kepada pelanggan. Gambaran tersebut berfokus pada representasi aspek-aspek aplikasi yang akan terlihat oleh pelanggan/pengguna. Berikut adalah alur proses dari Metode Prototyping :



Gambar 4. 1 Alur Proses Metode Prototyping

Gambar 4.2 menjelaskan mengenai alur pengembangan sistem dengan menggunakan metode Prototyping Oriented Software. Berikut tahapan-tahapan dari alur proses metode prototyping :

1. Informal Description of User Needs

Pada tahap pertama, dilakukan analisis kebutuhan dan pendefinisian kebutuhan. Kebutuhan yang dimaksud disini adalah kebutuhan pelanggan/pengguna.

2. User Interface Prototype and Completing Specification

Selanjutnya pada tahap kedua dilakukan pembuatan prototype dari aplikasi yang akan dibangun, mulai dari perancangan user interface prototyping dan dilanjutkan hingga penyusunan arsitektur dan komponen-komponen yang berkaitan dengan aplikasi yang akan dibangun.

3. System Architecture, Component Structure. Architecture and Component Prototypes

Pada tahap ini dilakukan pengembangan sistem, dimana aplikasi akan dibangun sesuai dengan prototype yang telah dibuat sebelumnya.

4. Program and Documentation

Setelah aplikasi dibuat sesuai dengan kebutuhan maka dilakukan proses pengujian aplikasi sebelum aplikasi tersebut digunakan oleh perusahaan, apabila saat pengujian aplikasi terdapat masalah maka tidak bisa melanjutkan ke tahap selanjutnya

sehingga permasalahan tersebut harus diselesaikan terlebih dahulu untuk melanjutkan pada tahap berikutnya.

5. Final Product

Pada tahap final product ini menyatakan bahwa aplikasi sudah sesuai dengan kebutuhan dan siap untuk digunakan.

ANALISIS DAN PERANCANGAN SISTEM

5.1 Analisis dan Perancangan Sistem

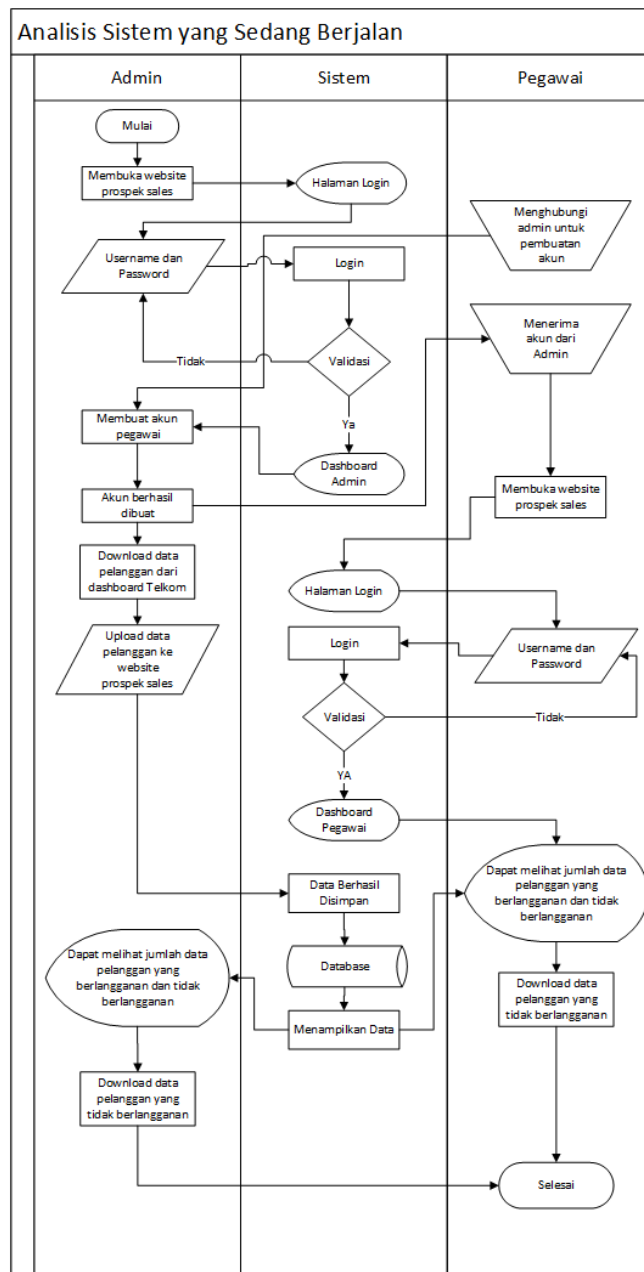
5.1.1 Analisis Sistem Berjalan (*Curent System*)

Analisis sistem merupakan sebuah tahapan paling awal dari pengembangan sistem yang menjadi pondasi menentukan keberhasilan sistem informasi yang dihasilkan nantinya. Tahapan ini sangat penting karena menentukan bentuk sistem yang harus dibangun. Tahapan ini bisa merupakan tahapan yang mudah jika kalian tahu betul dari fungsionalitas dari sistem informasi yang akan dibuat. Tetapi tahap ini bisa menjadi tahap yang paling sulit jika tidak bisa mengidentifikasi kebutuhannya atau tertutup pada pihak luar yang ingin mengetahui *detail* proses-proses bisnisnya.

5.1.1.1 Analisis Prosedur yang berjalan (*Flowmap/Flowchart*)

Hal yang dilakukan dalam pembuatan suatu sistem informasi adalah menganalisa sistem yang sedang di gunakan di lapangan. Dimana analisis sistem merupakan peroses menggali informasi dari suatu sistem, baik itu manual ataupun terkomputerisasi. Kemudian informasi tersebut diidentifikasi dan dievaluasi sehingga dapat ditemukan permasalahan serta

solusi untuk masalah tersebut yang kemudian dapat mengarah pada pembuatan sistem. Berikut Gambar 5.1 merupakan *flowmap Prospek Sales* yang sedang berjalan :



Gambar 5. 1 Flowmap Sistem yang Sedang Berjalan

5.1.1.2 Analisis Dokumen yang Digunakan

Dari hasil analisis yang dilakukan, dokumen yang digunakan diantaranya adalah dokumen *add On* berupa data pelanggan yang berlangganan produk dari *digital service*. Untuk mempermudah pendokumentasian pelanggan yang masih menggunakan produk dari *digital service* maka dibutuhkan suatu sistem yang dapat mempermudah dalam proses *upload* dan *update* data. Adapun dokumen yang dimaksud adalah sebagai berikut:

Tabel 5. 1 Dokumen Add On Pelanggan

Dibuat oleh	Sistem <i>website</i> Telkom
Dibuat untuk	Telkom Regional dan Witel
Isi	Berupa daftar pelanggan
Frekuensi	Dibuat setiap bulannya sesuai jumlah pelanggan
Tujuan	Mengetahui jumlah pelanggan

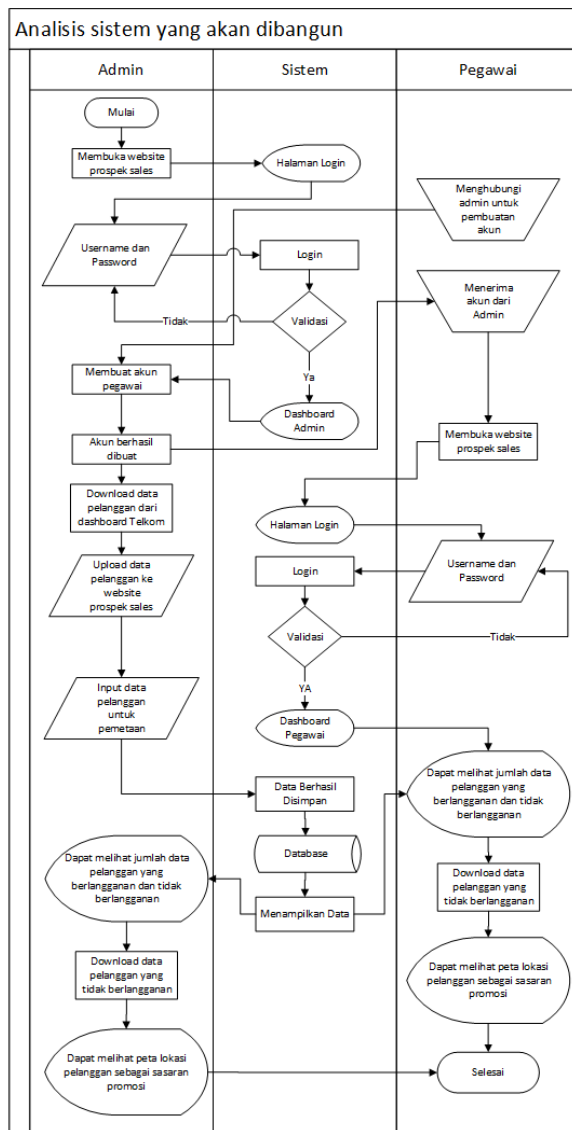
5.1.2 Analisis Sistem yang akan dibangun

Setelah melakukan analisa dan mengetahui kelemahan kelemahan pada sistem yang sedang berjalan, maka dapat dibuat sebuah sistem baru yang dapat memperbaiki kekurangan kekurangan terhadap sistem yang lama sehingga dapat membantu untuk memproses informasi dengan lebih cepat. Kekurangan pada sistem yang lama yaitu belum ada Pemetaan

pelanggan IndiHome sebagai sasaran promosi. Pada sistem yang akan dibangun ini maka akan ditambahkan untuk Pemetaan pelanggan IndiHome menggunakan *Google Maps*.

5.1.2.1 Analisis Prosedur yang akan dibangun

Analisis untuk sistem ini digunakan untuk mengetahui sistem yang akan dibangun pada perancangan sistem informasi *Prospek Sales*. Berikut Gambar 5.2 merupakan *flowmap Prospek Sales* yang akan dibangun:



Gambar 5. 2 Flowmap Proses yang Akan Dibangun

Pada Gambar 5.2 pertama Admin *login* terlebih dahulu untuk masuk ke halaman *dashboard* admin, kemudian admin bisa mengelola data *user/pegawai*, *download* data pelanggan

IndiHome dari dashboard telkom kemudian di *upload* ke *website prospek sales* dan bisa melakukan input data pelanggan untuk pemetaan, data tersebut tersimpan ke *database* dan pada *dashboard* admin dapat melihat jumlah pelanggan indihome yang tidak berlangganan layanan *add on*, *download* data pelanggan yang tidak berlangganan dan dapat melihat lokasi pelanggan pada peta/*maps* sebagai sasaran promosi. Untuk *user/pegawai* yang belum mempunyai akun terlebih dahulu menghubungi admin untuk pembuatan akun, setelah akun dibuat *user/pegawai* membuka *website prospek sales* dan melakukan login apabila data *valid* maka *user/pegawai* bisa mengakses *dashboard user/pegawai*, dari *dashboard* tersebut bisa melihat jumlah data pelanggan indihome yang tidak berlangganan layanan *add on*, *download* data pelanggan yang tidak berlangganan dan dapat melihat lokasi pelanggan pada peta/*maps* sebagai sasaran promosi.

5.1.2.2 Analisis Dokumen yang dibangun

Analisis dokumen aplikasi merupakan suatu kebutuhan dokumen yang berhubungan dengan dokumen sistem yang akan dibuat. Dimana menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sebuah sistem. Adapun kebutuhan aplikasi yang akan dibuat yaitu pengelolaan data proses sesuai dengan urutan sebagai berikut :

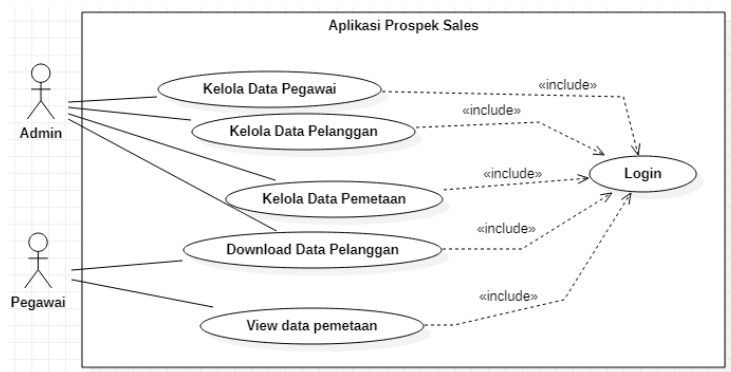
1. *Login* admin (Manager Unit *Digital Service & Wifi* Treg III yang diberi hak akses sebagai admin) dan pegawai;
2. Kelola data pelanggan atau *Add On*;
3. Kelola data pemetaan;
4. Kelola data akun pegawai.

Setiap proses memiliki representasi masing-masing pada sebuah tabel atau data yang terdapat pada *database* yang telah dirancang sebelumnya, dan setiap proses berhubungan langsung dengan entitas atau aktor.

5.1.2.3 UML (*Unified Modeling Language*)

Proses analisis menggunakan UML harus diberikan spesifikasi prosesnya dan dijelaskan dengan tulisan secara lengkap. Contoh Tahapan proses pada UML :

1. *Use Case Diagram*



Gambar 5. 3 Use Case Diagram

a. Definisi Aktor

Pada definisi aktor akan menjelaskan aktor-aktor yang terlibat dalam aplikasi adalah sebagai berikut :

Tabel 5. 2 Definisi Aktor

No.	Aktor	Deskripsi
1	Admin	a. <i>Login</i> b. Kelola Data Pegawai c. Kelola Data Pelanggan d. Kelola Data Pemetaan e. Download Data Pelanggan
2	Pegawai	a. <i>Login</i> b. Download Data Pelanggan c. View Data Pemetaan

b. Definisi Use Case

Use case merupakan teknik untuk merekam persyaratan fungsional sebuah sisistem. *Use cas* mendeskripsikan interaksi antara para pengguna sistem dengan sistem itu sendiri, dengan memberikan sebuah narasi tentang bagaimana sistem tersebut digunakan.

Tabel 5. 3 Definisi Use Case

No.	Use Case	Deskripsi
1	<i>Login</i>	a. Menampilkan <i>form login</i>

		b. Menginputkan <i>username</i> dan <i>password</i>
2	Kelola Data Pegawai	a. Menampilkan Data <i>User</i> b. Mengelola Data <i>User</i>
3	Kelola Data Pelanggan	a. Download Data Pelanggan dari dashboard telkom b. Upload Data Pelanggan
4	Kelola Data Pemetaan	a. Input Nomor internet pelanggan untuk pemetaan b. Menampilkan Data Pemetaan
5	Download Data Pelanggan	a. Download data yang tidak berlangganan
6	View Pemetaan	a. Melihat data pelanggan dari hasil pemetaan di maps sebagai sasaran promosi

c. Skenario *Use Case*

Skenario *Use case* mendeskripsi urutan langkah – langkah dalam proses bisnis, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berikut ini penjelasan dari beberapa masing – masing skenario tersebut:Skenario *Use case* sebagai berikut :

Tabel 5. 4 Skenario Use Case Login

Identifikasi	
No.	UC1
Nama	<i>Login</i>
Tujuan	Memulai proses pada sistem
Deskripsi	Melakukan <i>login</i> untuk memulai proses-proses pada sistem
Aktor	Admin
Skenario	
Kondisi Awal	<i>Display Login</i>
Aksi Aktor	Reaksi Sistem
1. Memulai Aplikasi	a. <i>Display login</i>
2. <i>Input username dan password</i>	b. Melakukan validasi

Tabel 5. 5 Skenario Use Case Kelola Data Pegawai

Identifikasi	
No.	UC2
Nama	Kelola Data Pegawai

Tujuan	Mengelola Data Pegawai
Deskripsi	Mengelola data pegawai untuk pembuatan akun dan reset password
Aktor	<i>Admin</i>
Skenario	
Kondisi Awal	Halaman Kelola Pegawai
Aksi Aktor	Reaksi Sistem
1. Membuat akun pegawai	a. Data disimpan ke Database
2. Melihat data pegawai	b. Menampilkan data pegawai

Tabel 5. 6 Skenario Use Case Kelola Data Pelanggan

Identifikasi	
No.	UC3
Nama	Kelola Data Pelanggan
Tujuan	Mengelola Data Pelanggan
Deskripsi	Mengelola data pelanggan untuk mengetahui jumlah pelanggan yang tidak berlangganan dan pemetaan
Aktor	<i>Admin</i>
Skenario	
Kondisi Awal	Dashboard
Aksi Aktor	Reaksi Sistem
1. Melakukan login	a. Halaman dashboard

2. Pilih more info pada salah satu paket add on	b. Halaman paket add on
3. Upload data pelanggan	c. Check data duplikat
4. View data yang berlangganan dan tidak berlangganan	d. Menampilkan data yang berlangganan dan tidak berlangganan

Tabel 5. 7 Skenario Use Case Kelola Data Pemetaan

Identifikasi	
No.	UC4
Nama	Kelola Data Pemetaan
Tujuan	Mengelola Data Pemetaan
Deskripsi	Menginputkan data pelanggan berdasarkan no_internet untuk pemetaan
Aktor	<i>Admin</i>
Skenario	
Kondisi Awal	Halaman paket add on
Aksi Aktor	Reaksi Sistem
1. Mencari no_intenet pelanggan yang belum di inputkan	
2. Inputkan no_internet pada bagian form pemetaan	a. Get data pelanggan berdasarkan no_internet
3. Marker posisi pelanggan pada peta	b. Mendapatkan Latitude dan Longitude

untuk mengetahui titik koordinat	
4. Pilih button simpan	c. Data disimpan ke database
5. View peta pelanggan	d. Menampilkan peta lokasi pelanggan sebagai sasaran promosi

Tabel 5. 8 Skenario Use Case Download Data Pelanggan

Identifikasi	
No.	UC5
Nama	Download Data Pelanggan
Tujuan	Download Data Pelanggan
Deskripsi	Mendownload data pelanggan dari dashboard telkom kemudian di upload ke aplikasi prospek sales untuk mengetahui pelanggan yang tidak berlangganan add on dan melakukan pemetaan.
Aktor	Admin dan Pegawai
Skenario	
Kondisi Awal	Halaman paket add on
Aksi Aktor	Reaksi Sistem

1. Admin Download Data Pelanggan dari dashboard telkom	a. <i>Get data</i> pelanggan dari dashboard telkom
2. Pegawai Download data pelanggan dari aplikasi prospek sales	b. <i>Get data</i> pelanggan yang tidak berlangganan

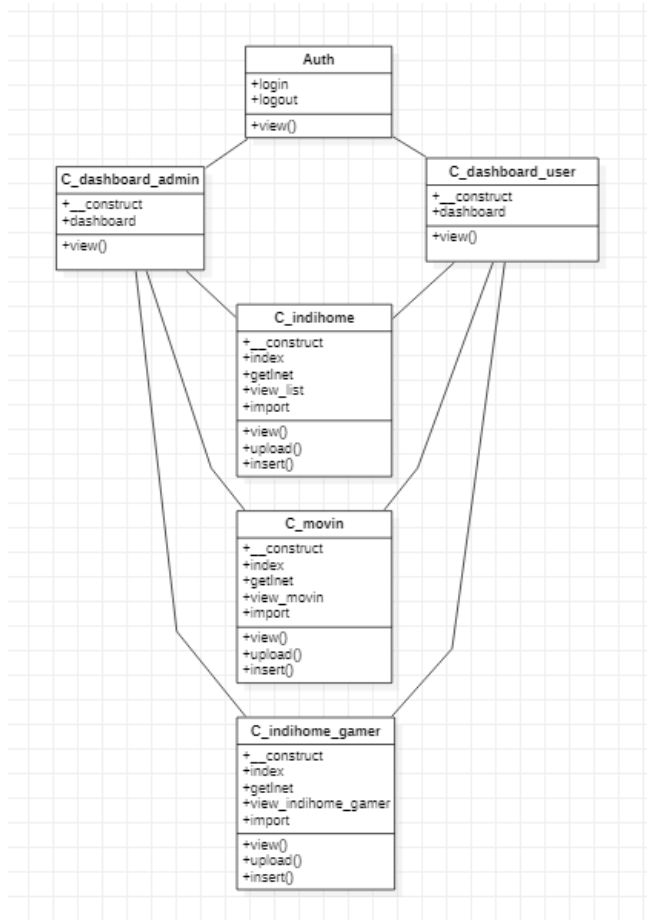
Tabel 5. 9 Skenario Use Case View Pemetaan

Identifikasi	
No.	UC6
Nama	View Pemetaan
Tujuan	Melihat lokasi pelanggan
Deskripsi	Dapat melihat lokasi pelanggan sebagai sasaran promosi
Aktor	Admin dan Pegawai
Skenario	
Kondisi Awal	Halaman peta pelanggan
Aksi Aktor	Reaksi Sistem
1. Memilih peta pelanggan pada menu sidebar	a. menampilkan data pelanggan pada ggoogle maps
2. Dapat melihat lokasi pelanggan	

2. Class Diagram

Class Diagram menggambarkan struktur dan hubungan antar objek – objek yang ada pada sistem. Struktur itu

meliputi atribut – atribut dan method – method yang ada pada masing – masing kelas.



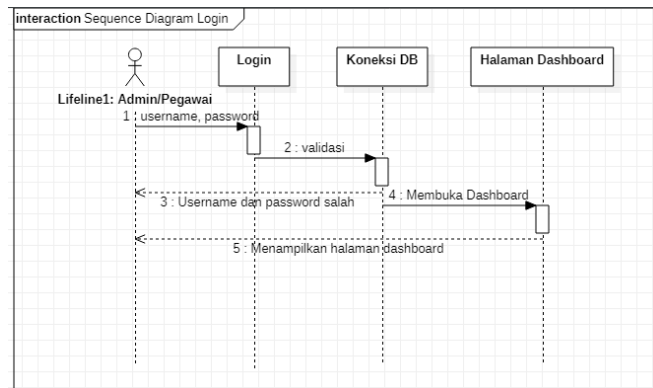
Gambar 5. 4 Class Diagram

3. Sequence Diagram

Sequence diagram menggambarkan interaksi antar masing – masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data

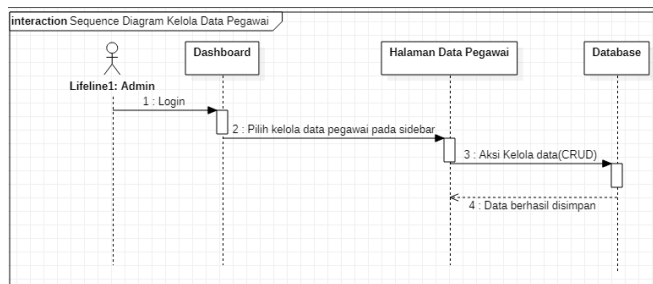
antar objek – objek yang saling berinteraksi. Berikut ini penjelasan dari masing – masing *sequence diagram* :

a. *Sequence Diagram Login*



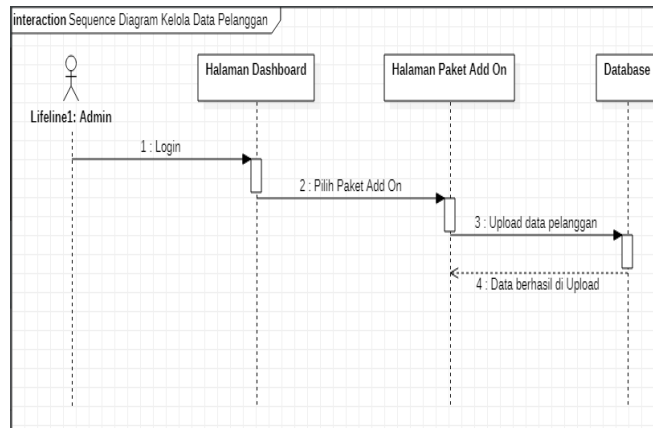
Gambar 5. 5 Sequence Diagram Login

b. *Sequence Diagram Kelola Data Pegawai*



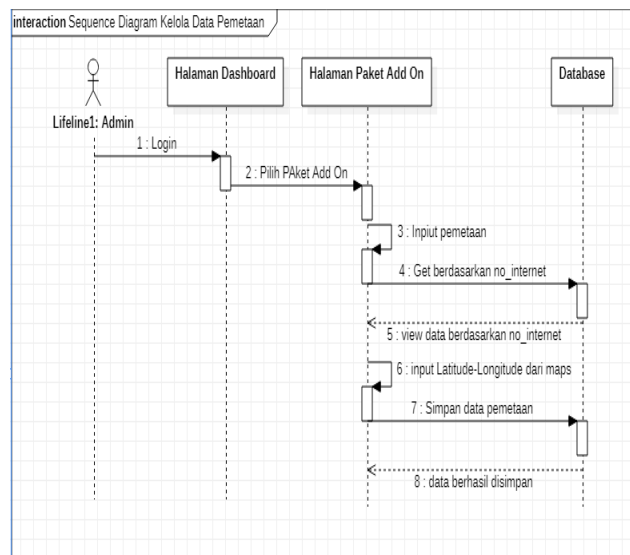
Gambar 5. 6 Sequence Diagram Kelola Data Pegawai

c. *Sequence Diagram Kelola Data Pelanggan*



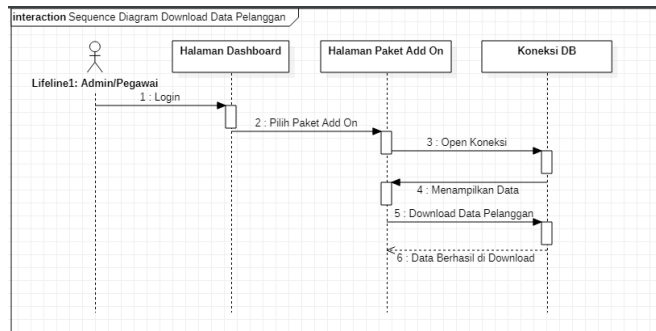
Gambar 5. 7 Sequence Diagram Kelola Data Pelanggan

d. *Sequence Diagram Kelola Data Pemetaan*



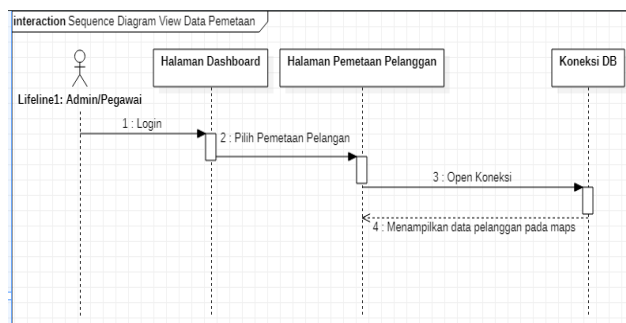
Gambar 5. 8 Sequence Diagram Kelola Data Pemetaan

e. *Sequence Diagram Download Data Pelanggan*



Gambar 5. 9 Sequence Diagram Download Data Pelanggan

f. *Sequence Diagram View Data Pemetaan*

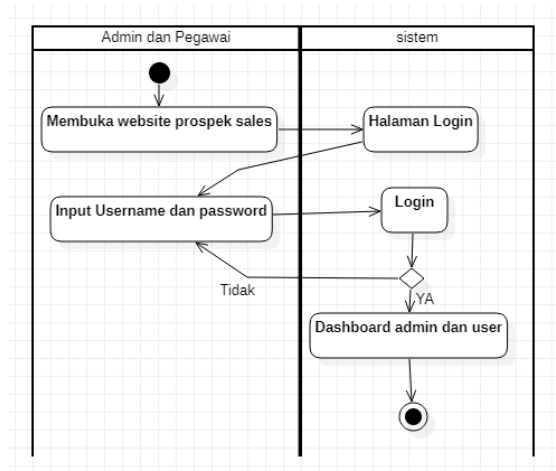


Gambar 5. 10 Sequence Diagram View Data Pemetaan

4. Activity Diagram

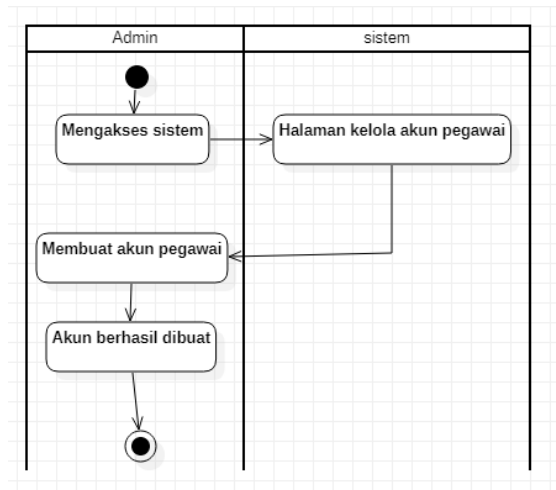
Activity diagram memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses yang mengacu pada *use case diagram* yang ada. Berikut ini alur proses dari masing – masing *activity diagram*:

a. Activity Diagram Login



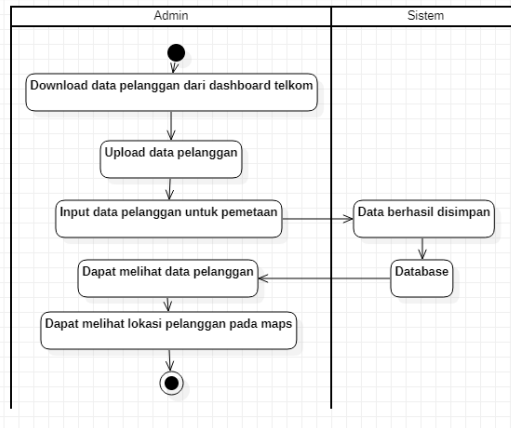
Gambar 5. 11 Activity Diagram Login

b. Activity Diagram Kelola Data Pegawai



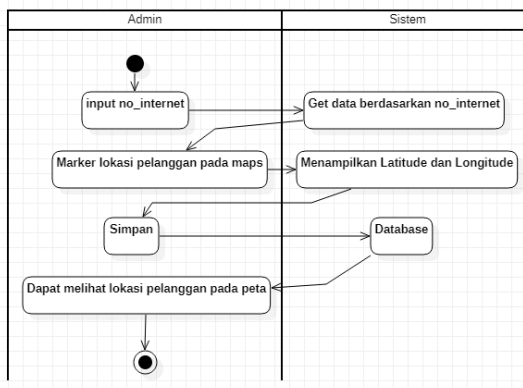
Gambar 5. 12 Activity Diagram Kelola Data Pegawai

c. Activity Diagram Kelola Data Pelanggan



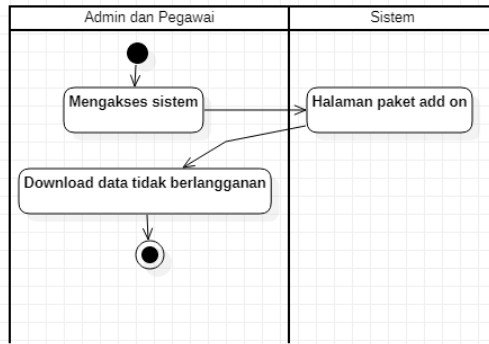
Gambar 5. 13 Activity Diagram Kelola Data Pelanggan

d. Activity Diagram Kelola Data Pemetaan



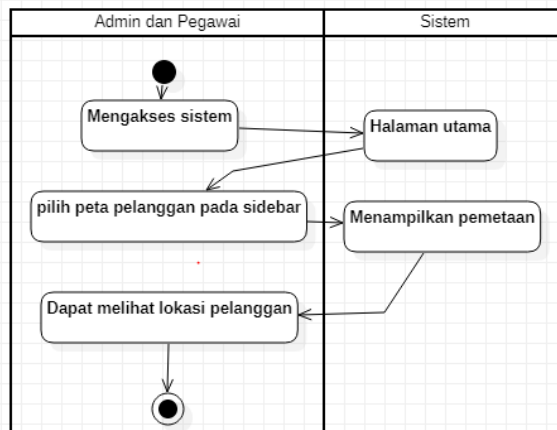
Gambar 5. 14 Activity Diagram Kelola Data Pemetaan

e. Activity Diagram Download Data Pelanggan



Gambar 5. 15 Activity Diagram Download Data Pelanggan

f. Activity Diagram View Pemetaan

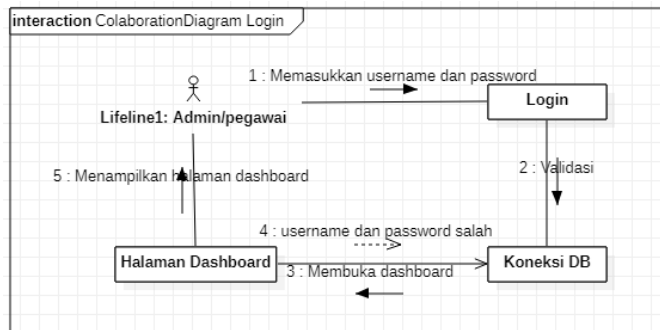


Gambar 5. 16 Activity Diagram View Pemetaan

5. Collaboration Diagram

a. Collaboration Diagram Login

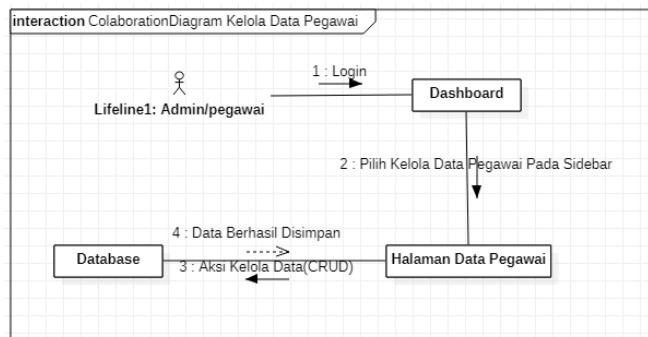
Berikut adalah *collaboration diagram* login dapat di lihat pada Gambar 5.17



Gambar 5. 17 Collaboration Diagram Login

b. Collaboration Diagram Kelola Data Pegawai

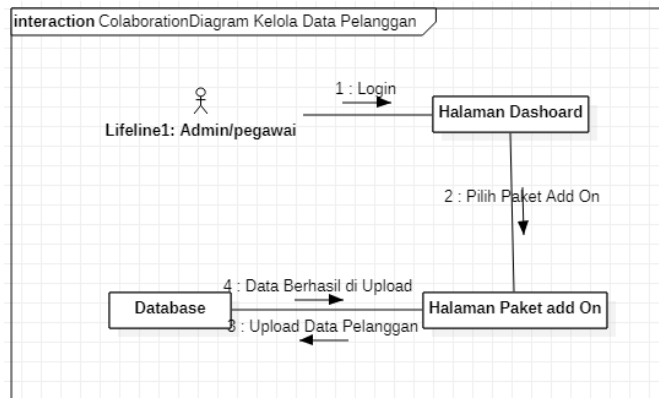
Berikut adalah *collaboration diagram* kelola data pegawai dapat di lihat pada Gambar 5.18



Gambar 5. 18 Collaboration Diagram Kelola Data Pegawai

c. Collaboration Diagram Kelola Data Pelanggan

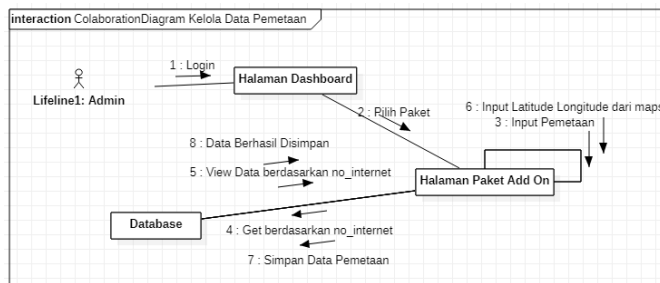
Berikut adalah *collaboration diagram* kelola data pelanggan dapat di lihat pada Gambar 5.19



Gambar 5. 19 Collaboration Diagram Kelola Data Pelanggan

d. Collaboration Diagram Kelola Data Pemetaan

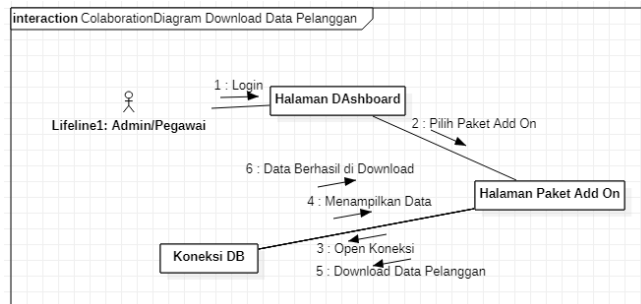
Berikut adalah *collaboration diagram* kelola data pemetaan dapat dilihat pada Gambar 5.20



Gambar 5. 20 Collaboration Diagram Kelola Data Pemetaan

e. Collaboration Diagram Download Data Pelanggan

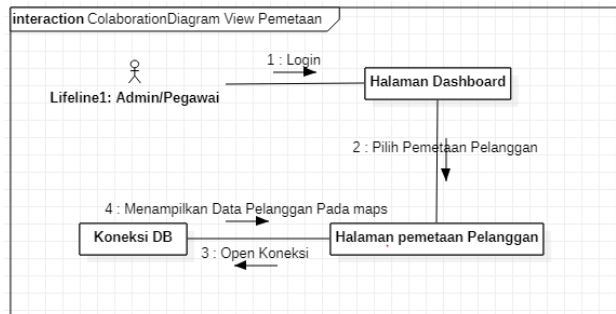
Berikut adalah *collaboration diagram* download data pelanggan dapat dilihat pada Gambar 5.21



Gambar 5. 21 Collaboration Diagram Download Data Pelanggan

f. Collaboration Diagram View Data Pemetaan

Berikut adalah *collaboration diagram view data pemetaan* dapat di lihat pada Gambar 5.22



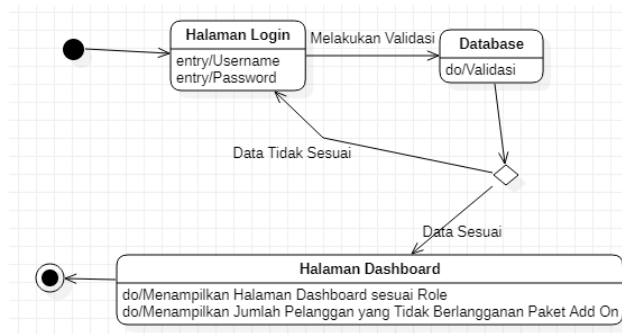
Gambar 5. 22 Collaboration Diagram View Data Pemetaan

6. Statechart Diagram

Diagram status digunakan untuk menyatakan kondisi (status) sebuah objek pada saat sistem informasi berjalan. Diagram ini diadopsi dari penggambaran kondisi mesin

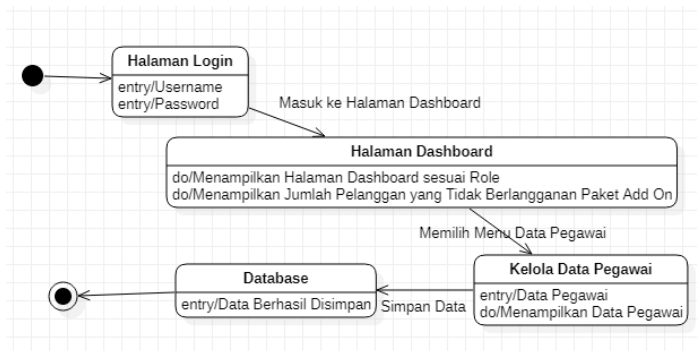
status (*state machine*) yang menggambarkan status apa saja yang dialami oleh mesin.

a. Statechart Diagram Login



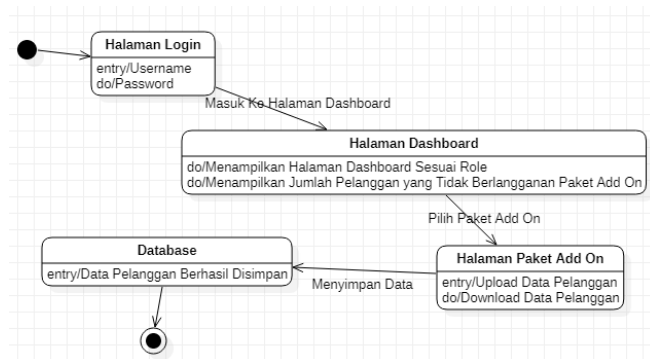
Gambar 5. 23 Statechart Diagram Login

b. Statechart Diagram Kelola Data Pegawai



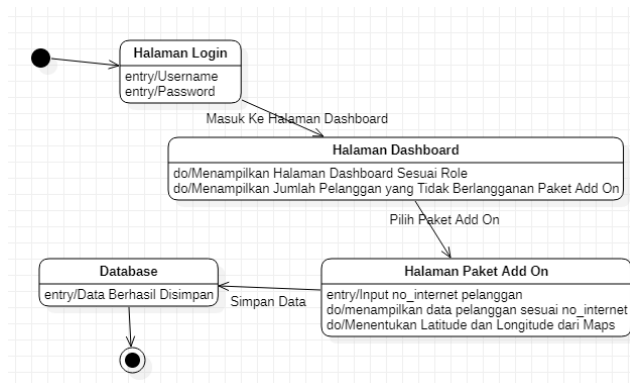
Gambar 5. 24 Statechart Diagram Kelola Data Pegawai

c. Statechart Diagram Kelola Data Pelanggan



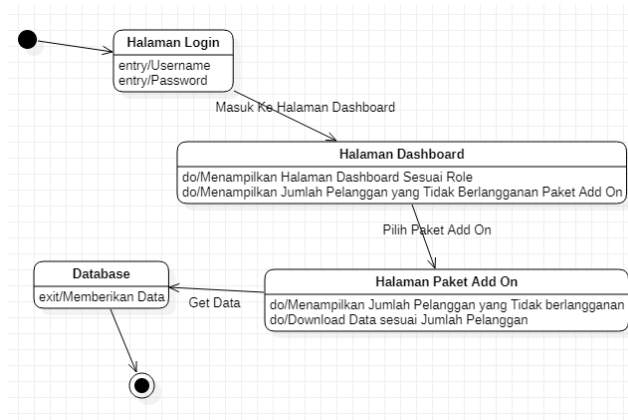
Gambar 5. 25 Statechart Diagram Kelola Data Pelanggan

d. Statechart Diagram Kelola Data Pemetaan



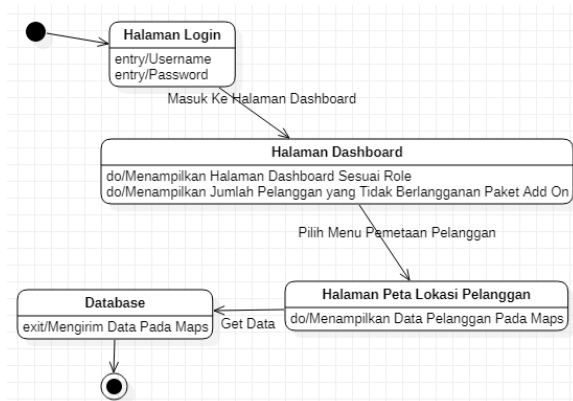
Gambar 5. 26 Statechart Diagram Kelola Data Pemetaan

e. Statechart Diagram Download Data Pelanggan



Gambar 5. 27 Statechart Diagram Download Data Pelanggan

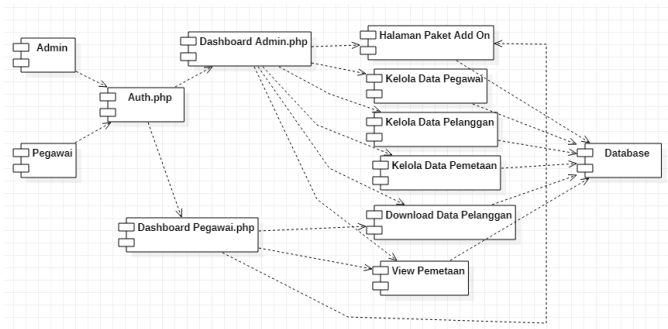
f. Statechart Diagram View Data Pemetaan



Gambar 5. 28 Statechart Diagram View Data Pemetaan

7. Component Diagram

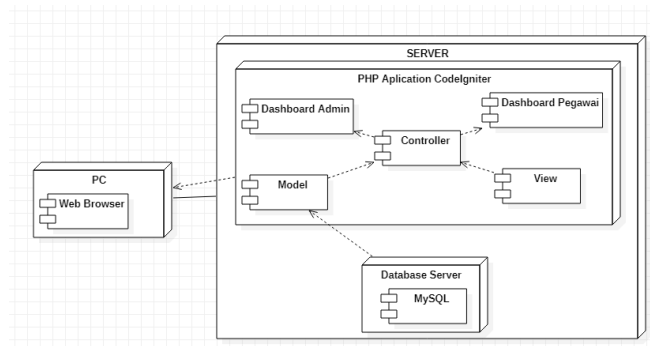
Component diagram digunakan untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.



Gambar 5. 29 Componentd Diagram

8. 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.



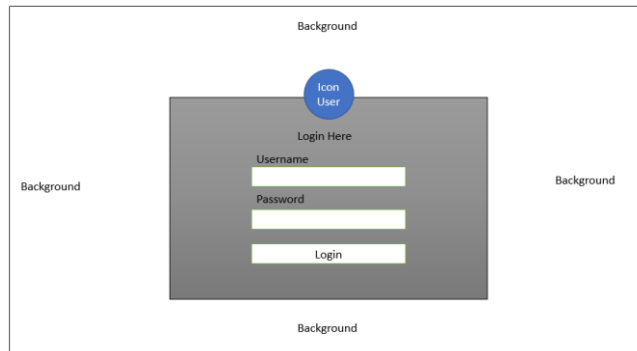
Gambar 5. 30 Deployment Diagram

5.2 Perancangan *User Interface* Sistem

Setelah melakukan analisis masuk pada tahap kedua dalam penerapan metode *incremental*, yaitu tahap desain. Pada tahap ini penulis masih melakukan strategi wawancara dengan pembimbing untuk mengetahui bagaimana bentuk perancangan antarmuka yang digunakan pada sistem informasi Prospek Sales.

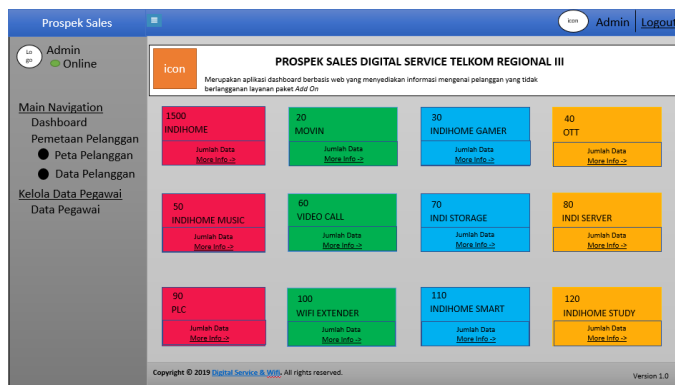
Rancangan *Interface* adalah rancangan pembangunan dari komunikasi antar pemakai (*administrator*) dengan komputer. Antar muka (*interface*) ini terdiri dari proses pemasukan data ke sistem dan menampilkan *output* informasi kepada *administrator*. Berikut beberapa gambar bentuk rancangan *user interface* yang sesuai dengan *scoop* penulis:

1. User Interface Login Admin/Pegawai



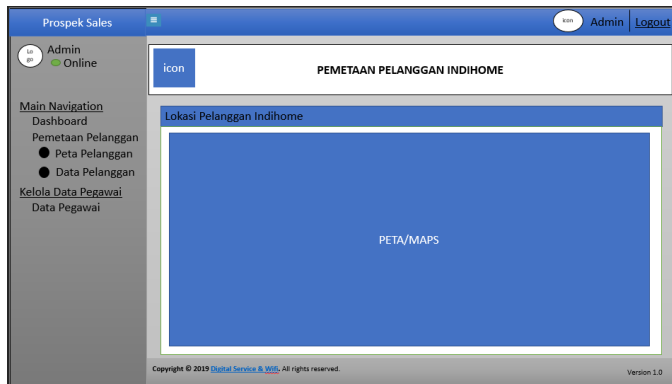
Gambar 5. 31 User Interface Login Admin/Pegawai

2. User Interface Dashboard Admin



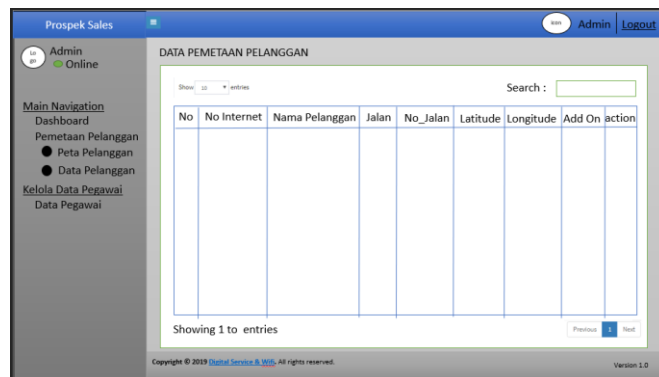
Gambar 5. 32 User Interface Dashboard Admin

3. *User Interface* Halaman Peta Pelanggan



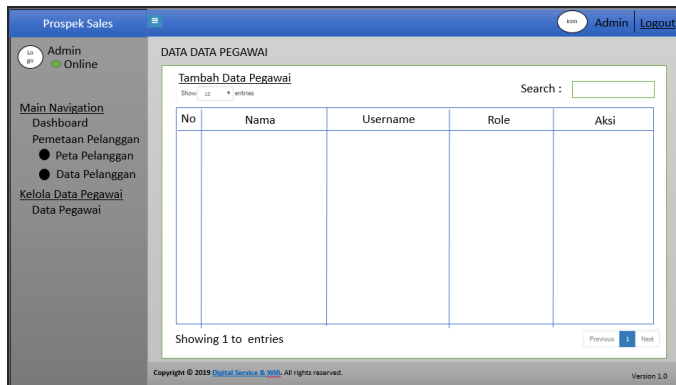
Gambar 5. 33 User Interface Halaman Peta Pelanggan

4. *User Interface* Halaman Data Pelanggan



Gambar 5. 34 User Interface Halaman Data Pelanggan

5. User Interface Halaman Data Pegawai



Gambar 5. 35 User Interface Halaman Data Pegawai

6. User Interface Tambah Data Pegawai

Form Input Data Pegawai

Nama Pegawai

Username

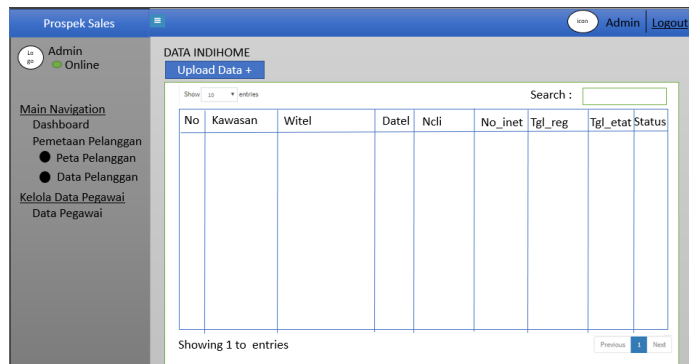
Password

Role

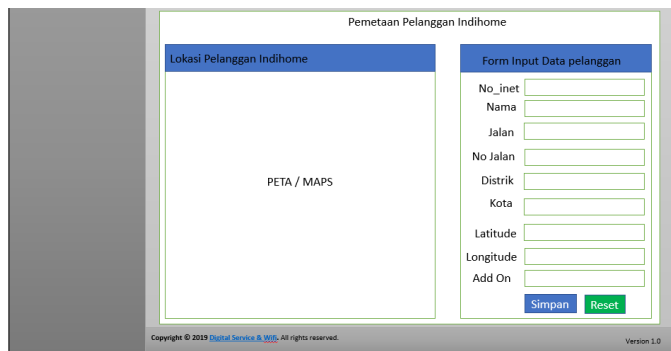
--Pilih Role--

Gambar 5. 36 User Interface Tambah Data Pegawai

7. *User Interface* Halaman Indihome Admin



Gambar 5. 37 User Interface Halaman Indihome Admin



Gambar 5. 38 User Interface Halaman Indihome Admin Bagian Pemetaan

8. User Interface Halaman Paket Add On Admin

Prospek Sales

Admin | Logout

Admin Online

Main Navigation

- Dashboard
- Pemetaan Pelanggan
 - Peta Pelanggan
 - Data Pelanggan
- Kelola Data Pegawai
 - Data Pegawai

DATA PROSPEK

Periode: TO

No	Witel	Jumlah Tidak Berlangganan
1.	Bandung	xxxxxx
Total		xxxxxx

Data Berlangganan

Show: 10 entries Search:

Menampilkan Data Dari Tiap Tabel Paket Add On

Gambar 5. 39 User Interface Halaman Paket Add On Admin

Prospek Sales

Admin | Logout

Pemetaan Pelanggan

Lokasi Pelanggan

PETA / MAPS

Form Input Data pelanggan

No_inet

Nama

Jalan

No Jalan

Distrik

Kota

Latitude

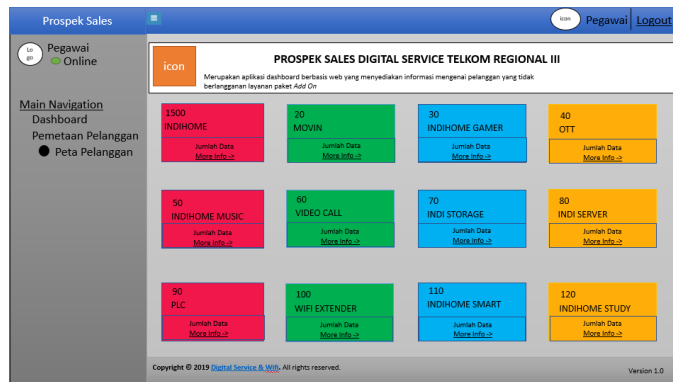
Longitude

Add On

Copyright © 2019 Digital Service & You. All rights reserved. Version 1.0

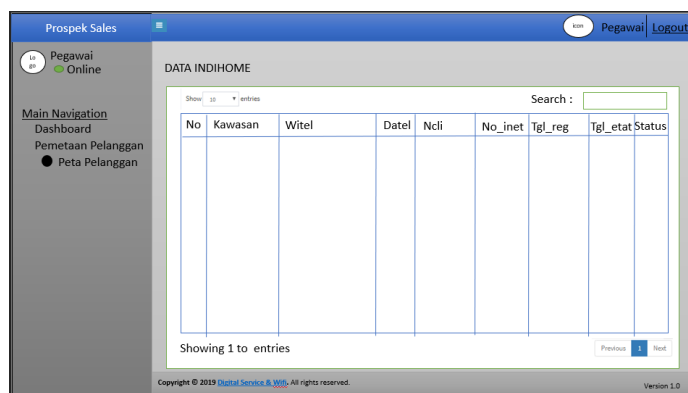
Gambar 5. 40 User Interface Halaman Paket Add On Admin
Bagian Pemetaan

9. User Interface Dashboard Pegawai



Gambar 5. 41 User Interface Dashboard Pegawai

10. User Interface Halaman Indihome Pegawai



Gambar 5. 42 User Interface Halaman Login Pegawai

11. User Interface Halaman Paket Add On Pegawai

The screenshot displays the 'Prospek Sales' application interface. The top header bar is blue and contains the title 'Prospek Sales', a user profile icon, the name 'Pegawai', and a 'Logout' button. The left sidebar is dark gray and lists navigation options: 'Main Navigation', 'Dashboard', 'Pemetaan Pelanggan', and 'Peta Pelanggan'. The main content area is light gray and features two data sections. The first section, 'DATA PROSPEK', includes a date range filter with 'Periode' and 'TO' input fields and a 'Submit' button. Below this is a table titled 'Data Tidak Berlangganan' with columns 'No', 'Witel', and 'Jumlah Tidak Berlangganan'. The table contains one data row for 'Bandung' and a 'Total' row. The second section, 'Data Berlangganan', includes a 'Search' input field and a large empty box with the text 'Menampilkan Data Dari Tiap Tabel Paket Add On'. At the bottom right of this section are 'Previous' and 'Next' pagination buttons.

No	Witel	Jumlah Tidak Berlangganan
1.	Bandung
Total	

Data Berlangganan		
Menampilkan Data Dari Tiap Tabel Paket Add On		

Gambar 5. 43 User Interface Halaman Paket Add On Pegawai

PENGAJIAN DAN EVALUASI

6.1 Implementasi

Implementasi merupakan sistem/aplikasi yang dibuat dengan merinci komponen – komponen pendukung berupa program, lingkungan implementasi, tampilan antarmuka, dan petunjuk penggunaan.

6.1.1 Lingkungan Implementasi

Perancang aplikasi ini dapat dilaksanakan dengan baik karena di dukung oleh perangkat pendukung yaitu perangkat lunak dan perangkat keras. Berikut kebutuhan perangkat keras dan perangkat lunak :

a. Kebutuhan Perangkat Keras\

Pada pembuatan aplikasi Prospek Sales ini terdapat perangkat keras pendukung yang digunakan adalah sebagai berikut :

Tabel 6. 1 Kebutuhan Perangkat Keras

No	Nama Perangkat	Spesifikasi
1	<i>Processor</i>	<i>Intel Core i5-8250U</i> <i>1.6GHz</i>
2	<i>RAM</i>	4GB
3	<i>Harddisk</i>	1TB

b. Kebutuhan Perangkat Lunak

Pada pembuatan aplikasi Prospek Sales ini terdapat perangkat keras pendukung yang digunakan adalah sebagai berikut :

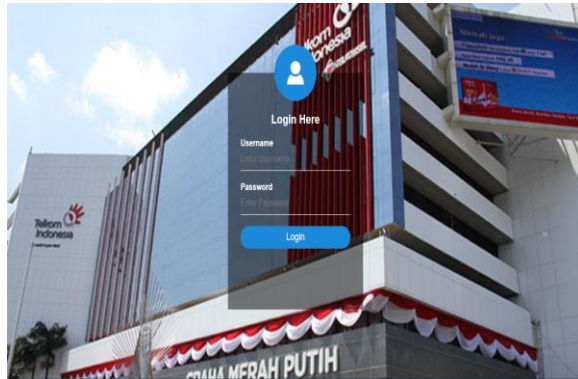
Tabel 6. 2 Kebutuhan Perangkat Keras

No	Nama Perangkat	Spesifikasi
1	Sistem Operasi	Windows 10
2	Database	<i>MySql</i>
3	Bahasa Pemrograman	<i>PHP</i>
4	Framework	<i>CodeIgniter</i>

6.2 Pembahasan Hasil Implementasi

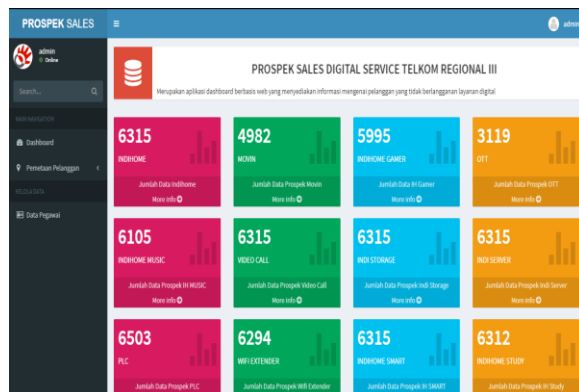
Berdasarkan perancangan yang telah dibuat, didapat hasil dari implementasi yang menjadi tujuan pembuatan perangkat lunak ini yaitu dapat melihat jumlah pelanggan yang tidak berlangganan paket *Add On* IndiHome dan dapat melihat lokasi pelanggan IndiHome pada peta/*maps* sebagai sasaran promosi, berikut adalah tampilan dari aplikasi Prospek Sales :

a. Halaman *Login* Admin/Pegawai



Gambar 6. 1 Halaman Login Admin/Pegawai

b. Halaman *Dashboard* Admin



Gambar 6. 2 Halaman Dashboard Admin

c. Halaman Peta Pelanggan



Gambar 6. 3 Halaman Peta Pelanggan

d. Halaman Data Pemetaan Pelanggan

PROSPEK SALES

DATA PELANGGAN

Tampilkan Data Pelanggan

Search

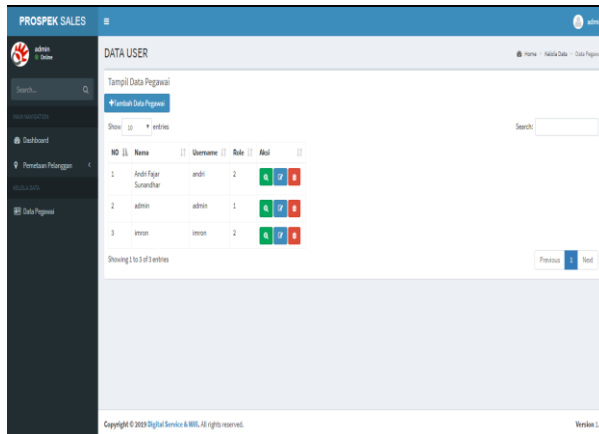
NO	INTERNET	NAMA PELANGGAN	JALAN	NO JALAN	DISTRIK	KOTA	LATITUDE	LONGITUDE	ADD ON	Action
1.	131188118651	MIRA HASAN BAKIR	WISSEN RI		CIPEDES	KOTA BANDUNG	-6.80221243898586	108.21801030122582	NOVON	
2.	131188118652	DIKE LUKMAN	Gg H RDO		GEDERWALING	KOTA BANDUNG	-6.80861445135054	108.21021026446539	RIZKHIS	
3.	131188118653	RINA OKTAVIANA	KURANG TINEUNG DALAM		CIPEDES	KOTA BANDUNG	-6.807181450796232	108.22239888368602	NOVON	
4.	131188118653	RINA OKTAVIANA	KURANG TINEUNG DALAM	70	CIPEDES	KOTA BANDUNG	-6.8022047481241565	108.22088105888684	RIZKHIS	

Showing 1 to 4 of 4 entries

Previous Next

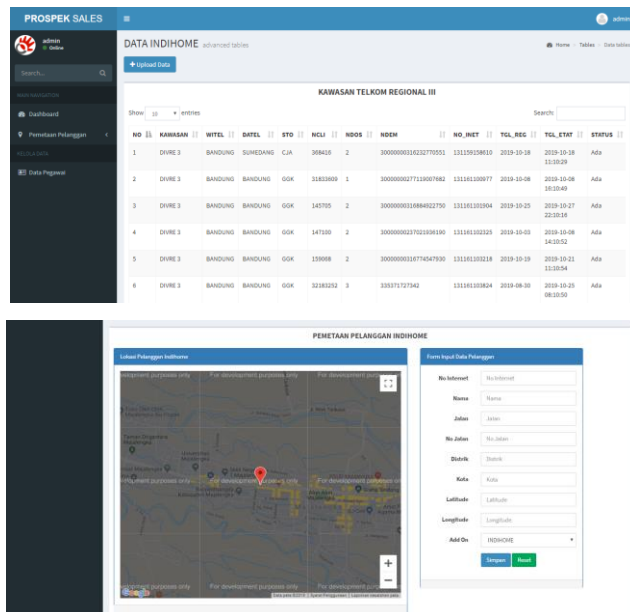
Gambar 6. 4 Halaman Data Pemetaan Pelanggan

e. Halaman Data Pegawai



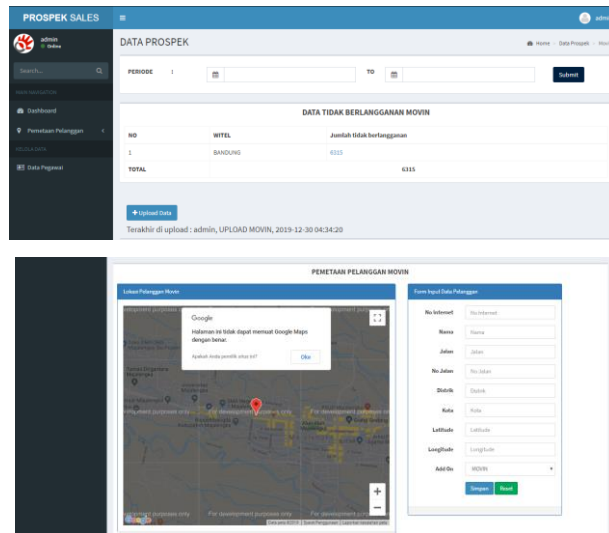
Gambar 6. 5 Halaman Data Pegawai

f. Halaman Indihome Admin



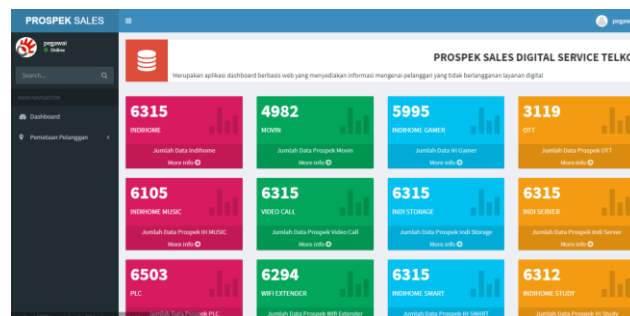
Gambar 6. 6 Halaman Indihome Admin

g. Halaman Paket *Add On* Admin



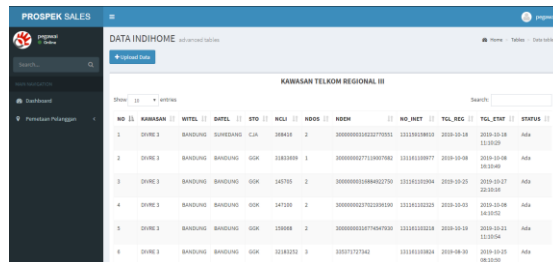
Gambar 6. 7 Halaman Paket *Add On* Admin

h. Halaman *Dashboard* Pegawai



Gambar 6. 8 Halaman *Dashboard* Pegawai

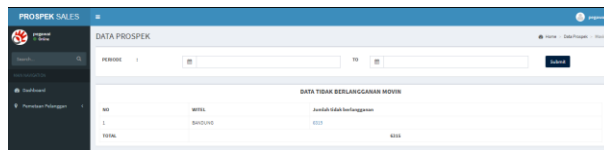
i. Halaman Indihome Pegawai



NO	KAWASAN	WITEL	DAPEL	STO	NGU	HDS	HDH	NO_NET	TOL_REG	TOL_ETAP	STATUS
1	DIHRE 1	BANDUNG	SUREBANG	CJA	38438	2	30000000000000000000	11110100000	2019-01-18	2019-01-18	Ada
2	DIHRE 1	BANDUNG	BANDUNG	GGK	1000000	1	30000000000000000000	11110100000	2019-01-18	2019-01-18	Ada
3	DIHRE 1	BANDUNG	BANDUNG	GGK	145700	2	30000000000000000000	11110100000	2019-01-25	2019-01-25	Ada
4	DIHRE 1	BANDUNG	BANDUNG	GGK	145700	2	30000000000000000000	11110100000	2019-01-25	2019-01-25	Ada
5	DIHRE 1	BANDUNG	BANDUNG	GGK	100000	2	30000000000000000000	11110100000	2019-01-18	2019-01-18	Ada
6	DIHRE 1	BANDUNG	BANDUNG	GGK	100000	3	30000000000000000000	11110100000	2019-01-18	2019-01-18	Ada

Gambar 6. 9 Halaman IndiHome Pegawai

j. Halaman Paket Add On Pegawai



DATA PROSPEK	
NO	WITEL
1	BANDUNG
Jumlah tidak terbagikan	
TOTAL	

Gambar 6. 10 Halaman Paket Add