

**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

BAHASA PEMROGRAMAN WEBSITE

Bahasa pemrograman digunakan untuk membuat *website* seperti blog, sistem informasi, *dashboard* dan lain sebagainya. Dalam istilah mendasar, Anda harus mengetahui tentang *client side scripting* dan *server side scripting*. Berikut akan dijelaskan maksud dari kedua istilah tersebut

Client Side Scripting

Merupakan kode-kode program yang dieksekusi pada sisi klien atau di komputer lain. *Browser web* mengambil sebuah halaman dari server, kemudian memproses kode yang disematkan di halaman dan menampilkan konten halaman yang diambil ke pengguna.

Server Side Scripting

Merupakan teknik yang digunakan dalam pengembangan web dengan melibatkan pengguna skrip pada server web yang menghasilkan respons yang disesuaikan untuk permintaan setiap pengguna (klien) ke situs web. Alternatifnya untuk server web itu sendiri ialah mengantarkan halaman web statis, Skrip dapat ditulis dalam salah satu dari sejumlah bahasa *script* pada sisi server yang tersedia. Ketika server menyajikan data secara umum untuk digunakan, misalnya menurut HTTP atau

FTP protokol, pengguna dapat memiliki pilihan dari sejumlah program klien (web browser yang paling modern dapat meminta dan menerima data menggunakan kedua dari mereka protokol). Dalam kasus aplikasi yang lebih khusus, programmer dapat menulis sendiri protokol server, client, dan komunikasi mereka, yang hanya dapat digunakan dengan satu sama lain.

1.1 Apa itu HTML ?

HTML (*HyperText Markup Language*) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi seperti gambar, teks, video dan suara pada penjelajah web Internet, yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.

Pada tahun 1980, fisikawan Tim Berners Lee, seorang kontraktor di CERN, mengajukan dan merancang INQUIRE, sebuah sistem bagi para periset CERN untuk menggunakan dan berbagi dokumen. Pada tahun 1989, Berners Lee menulis sebuah memo yang mengusulkan sistem hypertext berbasis internet.

1.2 Apa itu CSS ?

CSS (*Cascading Style Sheet*) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web

sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. Sama halnya seperti styles dalam aplikasi pengolahan kata, Microsoft Word, yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

CSS pertama kali diusulkan oleh Hakom Wium Lie pada tanggal 10 Oktober 1994. Pada saat itu, Lie bekerja dengan Tim Berners-Lee di CERN. Beberapa bahasa ala style lainnya untuk web diusulkan sekitar waktu yang sama, dan diskusi di milis publik dan di dalam World Wide Web Consortium yang akhirnya menghasilkan rekomendasi CSS W3C pertama (CSS1) yang dirilis pada tahun 1996.

Nama CSS didapat dari fakta bahwa setiap deklarasi style yang berbeda dapat diletakan secara berurutan, yang kemudian membentuk hubungan ayah-anak (*parent-child*) pada setiap style. CSS sendiri merupakan sebuah teknologi internet yang direkomendasikan oleh World Wide Web Consortium atau W3C pada tahun 1996. Setelah CSS distandarisasikan, Internet Explorer dan Netscape melepas browser terbaru mereka yang telah sesuai atau paling tidak hampir mendekati dengan standar CSS.

1.3 Apa itu JavaScript ?

Javascript adalah bahasa pemrograman tingkat tinggi untuk membuat website dinamis, dan jquery adalah *library* atau pustaka dari javascript yang dirancang untuk memudahkan penerapan *client side scripting* dan menyajikan sebuah paradigma baru pada penanganan event pada Javascript.

Pada tahun 1993, *National Center for Supercomputing Applications* (NCSA), merupakan sebuah unit dari Universitas Illinois di Urbana Champaign, merilis NCSA Mosaic, browser Web grafis pertama yang populer, yang memainkan peran penting dalam memperluas pertumbuhan World Wide Web. Pada tahun 1994, Sebuah perusahaan bernama Mosaic Communications didirikan di Mountain View, California dan mempekerjakan banyak penulis asli NCSA Mosaic untuk membuat Mosaic Netscape.

Pada tahun 1995, Netscape Communicatons merekrut Brendan Eich dengan tujuan menanamkan bahasa pemrograman skema ke dalam Netscape Navigatornya. Netscape Communications berkolaborasi dengan Sun Microsystems untuk memasukan dalam Netscape Navigator Sun, untuk bersaing dengan Microsoft untuk diadopsi pengguna teknologi Web dan Platform.

1.4 Apa itu JQuery ?

JQuery adalah sebuah framework berbasiskan Javascript. JQuery sama dengan Javascript Library, yaitu kumpulan kode atau fungsi Javascript siap pakai sehingga mempermudah dan mempercepat kita dalam membuat kode Javascript. JQuery adalah library Javascript yang dibuat untuk memudahkan pembuatan website dengan HTML yang berjalan di sisi Client.

Jquery pertama kali dikembangkan oleh Jhon Resig pada tahun 2005. Saat itu ia terinspirasi dari kode Behavior. Jhon merasa kode-kode behavior tidak elegan dan bahkan sangat jelek, maka dia memikirkan cara untuk membuat Library yang handal dan ringan untuk Javascript. Lalu lahirlah JQuery pada 14 Januari 2006 dan versi pertama dari JQuery sampai sekarang masih terus dikembangkan dan disempurnakan.

Semenjak dirilis pertama kali pada tahun 2006 oleh John Resig, jQuery telah mencuri perhatian para developer web. Buktinya,

pada tahun-tahun berikutnya jQuery telah banyak digunakan oleh website-website terkemuka di dunia seperti Google, Microsoft, intel, Nokia, Oracle, IBM, DELL, BBC, NBC, ESPN, EAPN, EA Sport, Twitter, Facebook, Amazon, Techorati, Youtube, Apple, Wordpress, Mozila, Netflix, Digg, Time, dan masih banyak lagi. Bahkan website lokalpun tidak mau ketinggalan, Lihat saja Detik, Studio 21 (21cineplex.com), Vivanews, Kompas, dan lain-lain.

1.5 Apa itu PHP ?

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan untuk membuat website atau situs dinamis dan menangani rangkaian bahasa pemrograman antara *client side scripting* dan *server side scripting*.

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Ramus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Ramus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI.

1.6 Apa itu CodeIgniter ?

CodeIgniter adalah sebuah aplikasi open source yang berupa kerangka kerja atau framework untuk membangun website menggunakan bahasa pemrograman PHP. Tujuannya memungkinkan pengembangan proyek yang lebih cepat daripada penulisan kode dasar atau kode terstruktur, dengan menyediakan banyak library yang biasanya digunakan dalam pengerjaan. Antarmuka yang sederhana dan struktur logika untuk mengakses library ini membuat CodeIgniter mudah digunakan dan dipelajari. CodeIgniter ditulis atau dibuat oleh Ellis Lab dan dirilis pertama kali pada 28 Februari 2006.

BAB II

LANDASAN TEORI

2.1 Codeigniter

2.1.1 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.



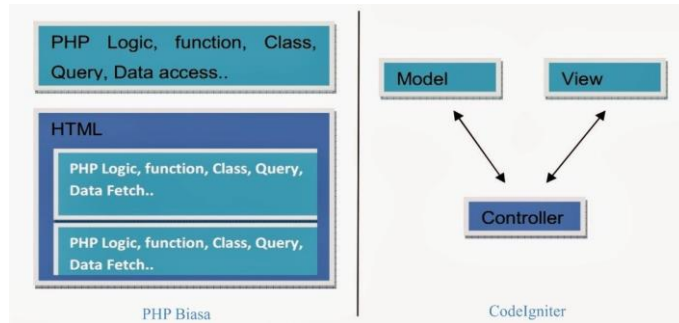
Gambar 1. Logo Codeigniter

2.1.2 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.



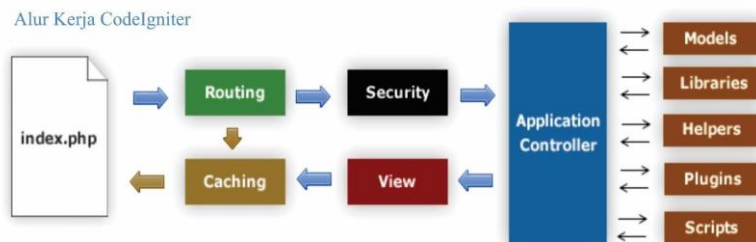
Gambar 2. Perbandingan PHP dengan Codeigniter

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.

2.1.3 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.

2.1.4 Keunggulan dan kekurangan Codeigniter

2.1.4.1 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.

2.1.4.2 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.

2.2 Framework

2.2.1 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.

2.2.2 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.

2.2.3 Kelebihan dan Kelemahan *Framework*

2.2.3.1 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.

2.2.3.2 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.

2.3 *Object Oriented Programming* (OOP)

2.3.1 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.

2.3.2 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.

2.3.3 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 – 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*.

2.3.4 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.

2.3.5 Keunggulan dan kekurangan OOP

2.3.5.1 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).

2.3.5.2 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.

2.4 Hypertext Preprocessor (PHP)

2.4.1 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 perhatian *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 dihunikan 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.

2.4.2 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.

2.4.3 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.

2.4.4 Keunggulan dan kekurangan PHP

2.4.4.1 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.

2.4.4.2 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.

2.5 Google Maps API

2.5.1 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*.

2.5.2 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 Programer 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.

2.5.3 Kelebihan dan kekurangan Google Maps API

2.5.3.1 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

2.5.3.2 Kekurangan Google Maps API

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

2.6 XAMPP

2.6.1 Sejarah XAMPP

XAMPP adalah pengembangan dari LAMP (*Linux Apache, MySQL, PHP and PERL*). XAMPP didirikan oleh Kai ‘Oswalad’ 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.

2.6.2 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 menginstalasi dan mengkonfigurasiknya 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.

2.6.3 Penjelasan singkatan XAMPP

- **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.

2.6.4 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*

Basaha pemrograman *script* sisi server yang didesain untuk pengembangan *web*

2.6.5 Bagian-bagian Xampp

Berikut ini adalah merupakan bagian-bagian dari XAMPP :

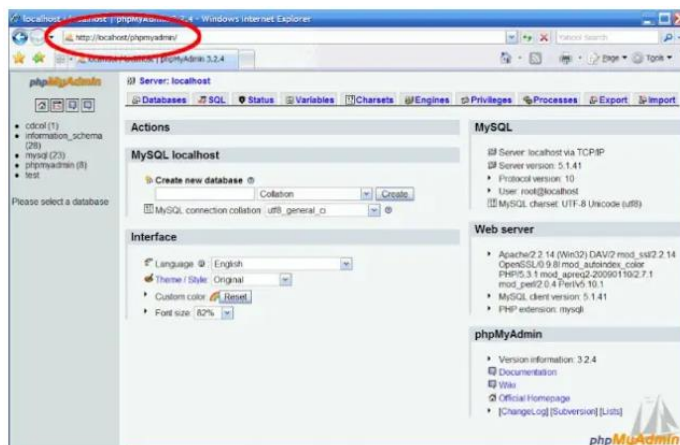
1. Htdocs

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

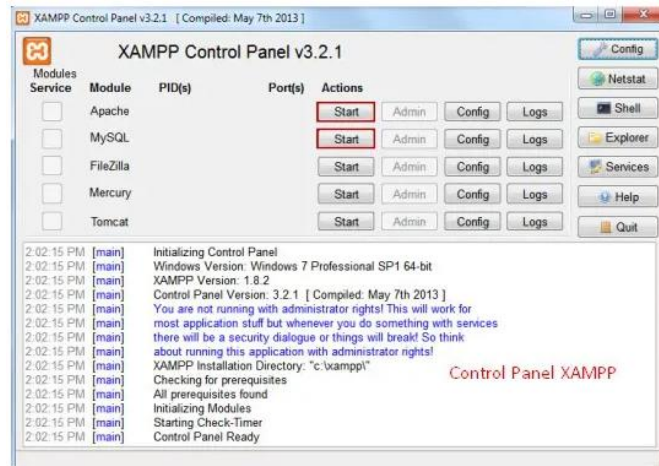


Gambar x. phpMyadmin

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.

2.6.6 Kelebihan dan Kekurangan XAMPP

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

2.6.6.1 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.

2.6.6.2 Kekurangan Xampp

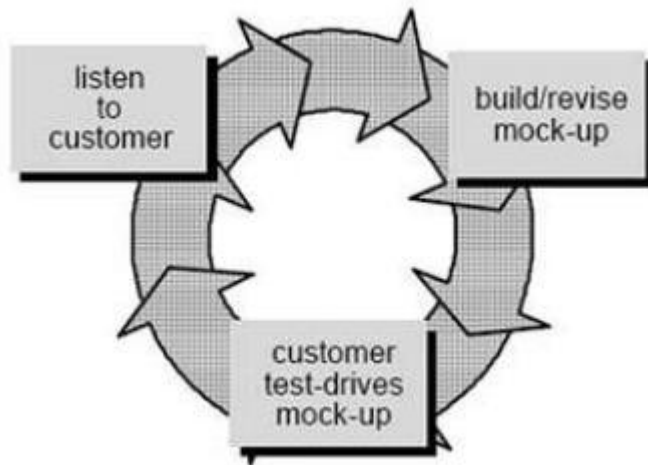
1. Tidak cocok untuk menangani data dalam jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data.
2. Memiliki keterbatasan dalam kemampuan kinerja pada server ketika data yang disimpan telah melebihi batas maksimal kemampuan.

2.7 Metodologi Penelitian

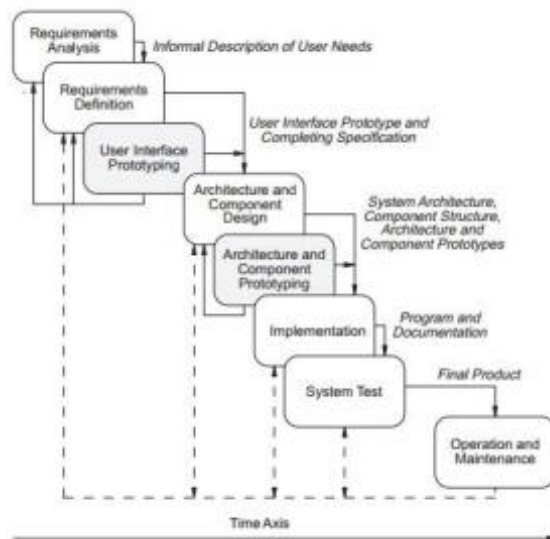
Metodologi penelitian adalah sekumpulan peraturan, kegiatan, dan prosedur yang digunakan oleh pelaku suatu disiplin. Metodologi juga merupakan analisis teoritis mengenai suatu cara atau metode. Penelitian merupakan suatu penyelidikan yang sistematis untuk meningkatkan sejumlah pengetahuan, juga merupakan suatu usaha yang sistematis dan terorganisasi untuk menyelidiki masalah tertentu yang memerlukan jawaban. Hakekat 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. 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* :



Pada gambar diatas 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* :



Pada gambar diatas 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.

2.8 Metode *Prototype*

Prototype adalah versi awal dari sistem perangkat lunak yang digunakan untuk mendemonstrasikan konsep-konsep, percobaan rancangan, dan menemukan lebih banyak masalah dan solusi yang memungkinkan. Sistem prototype

memperbolehkan pengguna untuk mengetahui bagaimana sistem berjalan dengan baik. Penggunaan metode prototyping di dalam penelitian ini bertujuan agar peneliti mendapatkan gambaran aplikasi yang akan dibangun melalui tahap pembangunan aplikasi prototype terlebih dahulu yang akan dievaluasi oleh user. Aplikasi prototype yang telah dievaluasi oleh user selanjutnya akan dijadikan acuan untuk membuat aplikasi yang dijadikan produk akhir sebagai output dari penelitian ini. Beberapa keunggulan dalam menggunakan metode prototyping :

1. Pengembang sistem dan pengguna saling berkomunikasi khususnya dalam hal penyamaan persepsi terhadap pemodelan sistem yang akan menjadi dasar pengembangan sistem operasionalnya,
2. Pelanggan/pengguna ikut terlibat secara aktif dan berpartisipasi dalam menentukan model sistem dan sistem operasionalnya sehingga pelanggan/pengguna akan puas karena sistem yang dibuat sesuai dengan keinginan dan harapannya,
3. Sistem yang dibangun memiliki kualitas yang diinginkan karena sesuai dengan kebutuhan yang ada.

2.9 *Unified Modelling Language (UML)*

Dalam perancangan Aplikasi tersebut, penulis akan menggunakan bahasa pemodelan *Unified Modeling Language* (UML) dalam membuat perancangannya. UML adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. UML juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek

BAB III

INSTALLASI APLIKASI YANG AKAN DIGUNAKAN

Alat yang digunakan untuk membangun Aplikasi Prospek Sales ini diantaranya :

- *Xampp*
- *Sublime text 3*
- *Codeigniter*

3.1 Instalasi Xampp

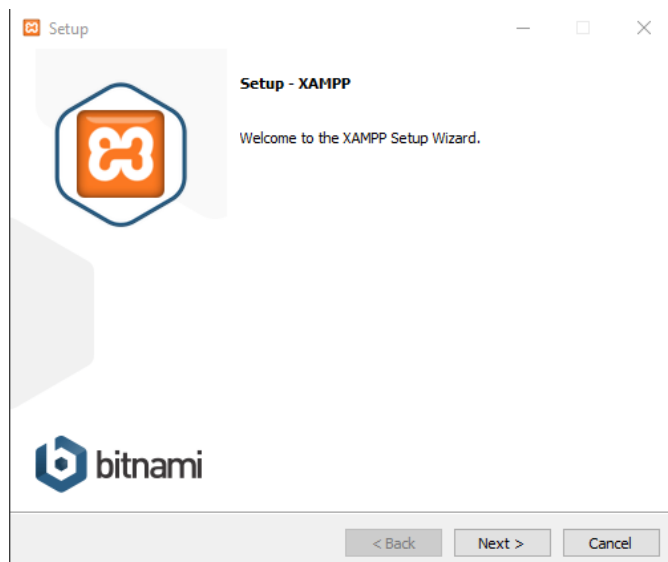
Xampp merupakan sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL / MariaDB, PHP, dan PERL. Sementara huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat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

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

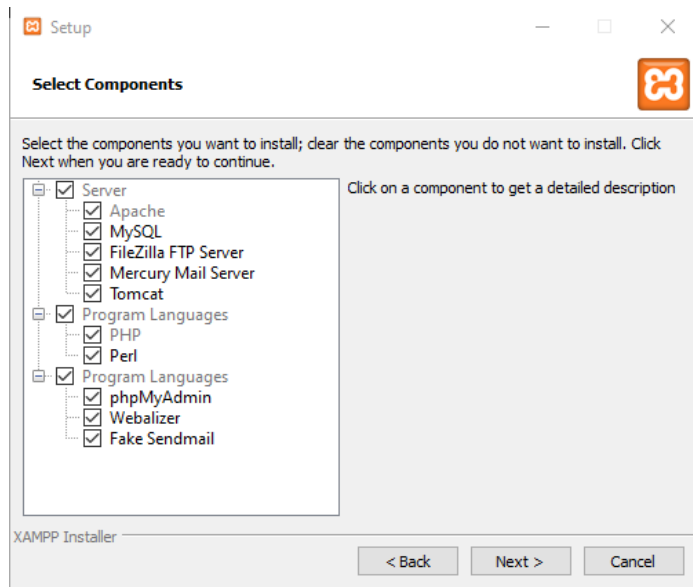
- a. Pertama, *download* terlebih dahulu file installer xampp pada link berikut ini. [https://apachefriends\(dot\)org](https://apachefriends(dot)org)



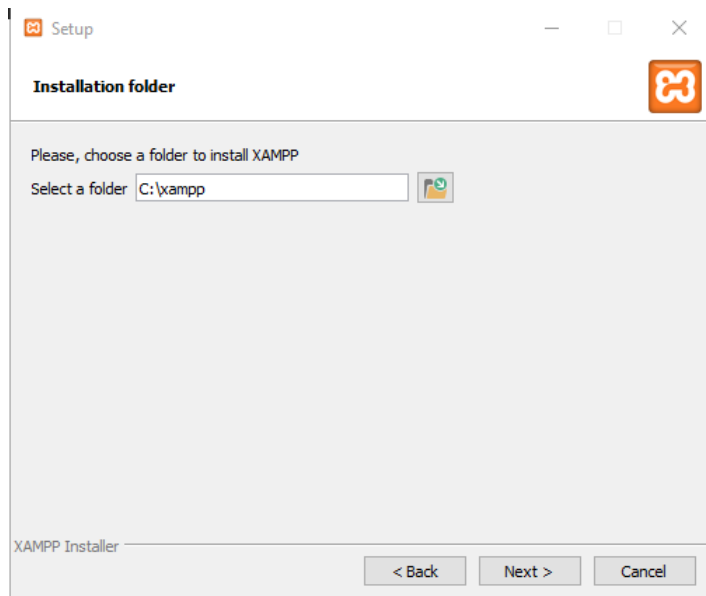
- b. Pilih file installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Klik file xampp yang sudah berhasil di *download*, maka akan muncul tampilan seperti ini.



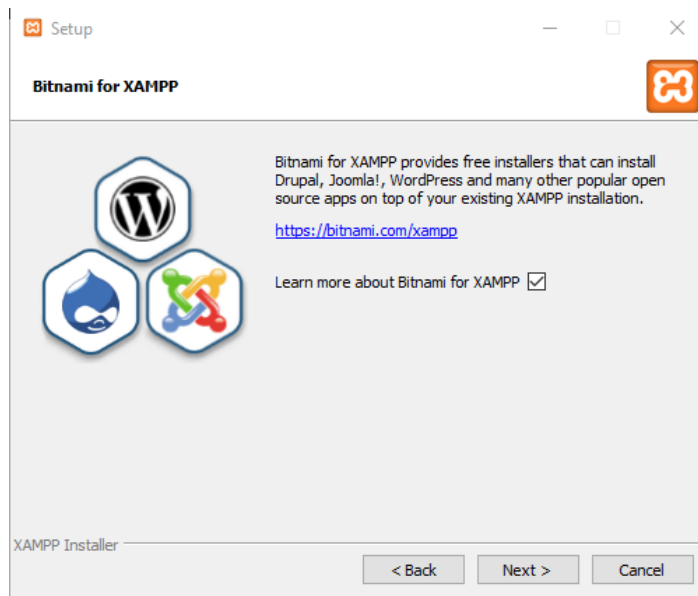
- d. Kemudian, pilih next untuk melanjutkan pada proses instalasi.



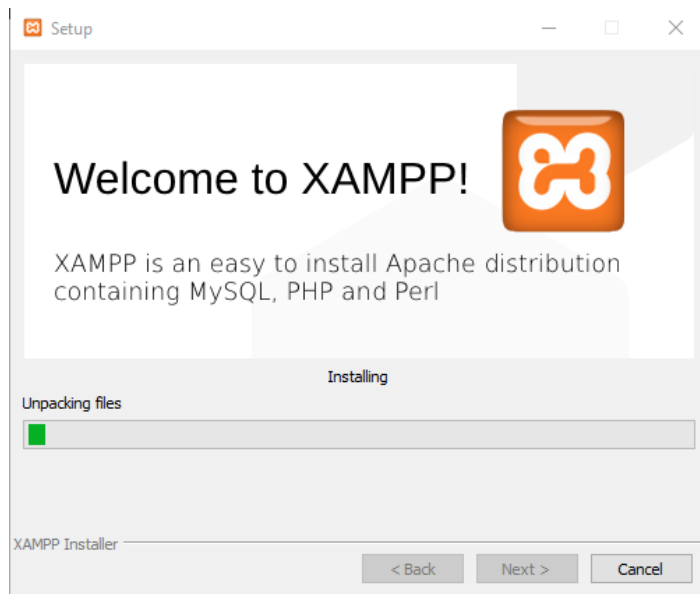
- e. Pada tampilan *select component*, centang semua pilihannya agar dapat menjalankan semua program yang dipilih. Kemudian pilih *next*.



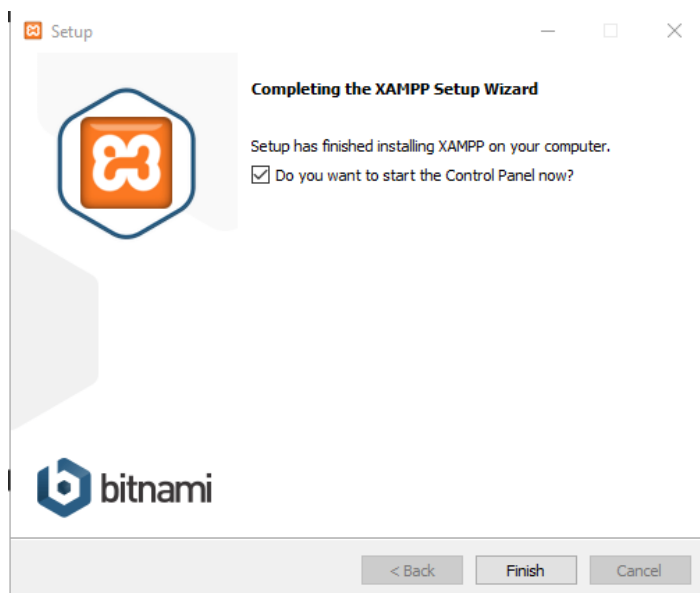
- f. Pada, *installation folder* secara *default* sistem akan menyimpannya pada direktori C. Jika sudah klik *next* untuk melanjutkan proses instalasi.



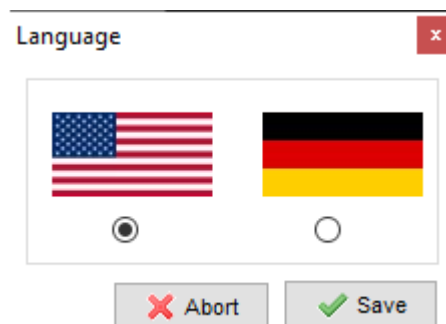
- g. Pada tampilan ini langsung saja klik *next* untuk melanjutkan.



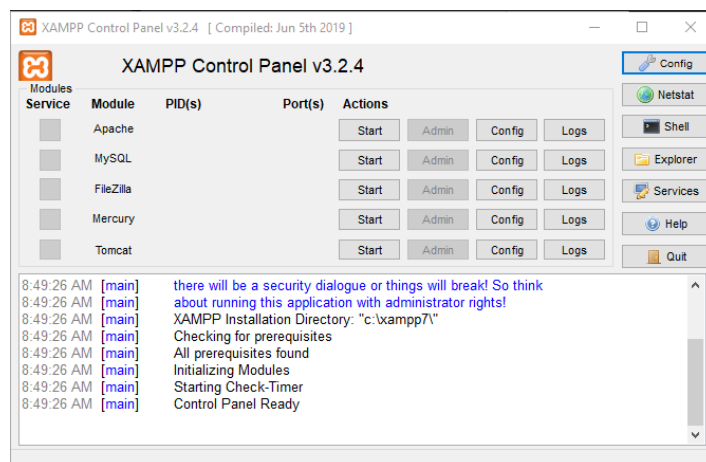
h. Kemudian tunggu hingga proses instalasi selesai.



- 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.

3.2 Instalasi Sublime Text 3

Sublime Text merupakan sebuah software text editor yang digunakan oleh programmer untuk membuat aplikasi berbasis web. Sublime biasanya digunakan oleh programmer untuk mengedit kode atau script dalam proses pembuatan aplikasi berbasis web. Berikut merupakan langkah-langkah untuk melakukan instalasi Sublime Text 3 pada sistem operasi windows.

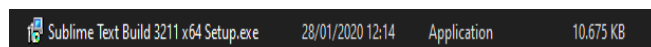
- a. Langkah pertama yang harus dilakukan yaitu download terlebih dahulu Sublime Text 3 di web resmi Sublime Text nya. Atau bisa akses dengan link berikut <https://www.sublimetext.com/3>
- b. Setelah masuk ke web resmi sublime nya maka terdapat beberapa pilihan diantaranya untuk sistem operasi OS X, Windows 32bit, Windows 64bit, dan linux.



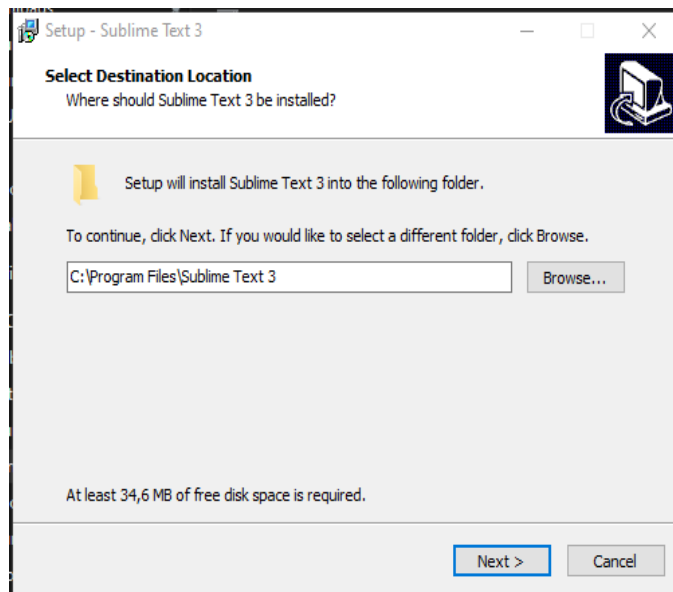
- c. Untuk pembuatan aplikasi prospek sales ini menggunakan sublime dengan sistem operasi Windows 64 bit.
- d. Klik pada Windows 64 bit maka akan otomatis terdownload.

- [Windows 64 bit - also available as a portable version](#)

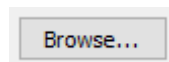
- e. Berikut file yang berhasil di download dalam bentuk .exe



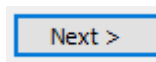
- f. Klik dua kali pada file sublime.exe tersebut.
- g. Maka akan muncul tampilan proses instalasinya, secara default sublime akan diarahkan ke direktori tempat penyimpanan hasil instalasinya di C:\Program Files\Sublime Text 3.



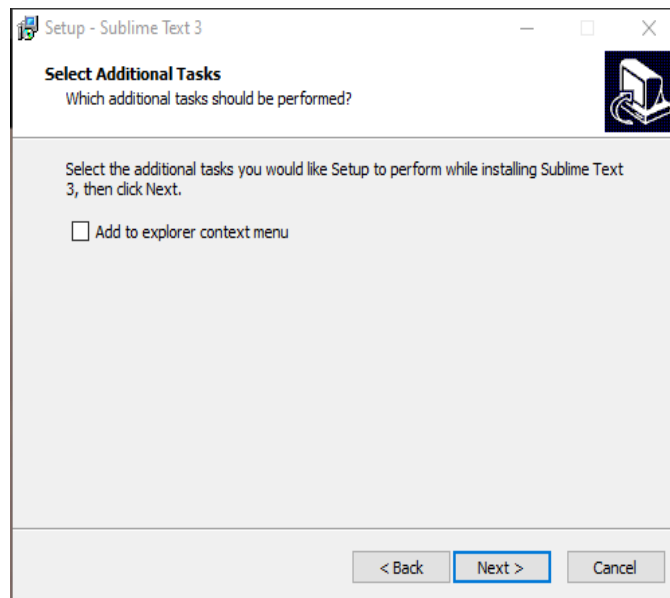
- h. Akan tetapi direktori tersebut bisa kita rubah tempat penyimpanan nya dengan mengklik bagian browse.



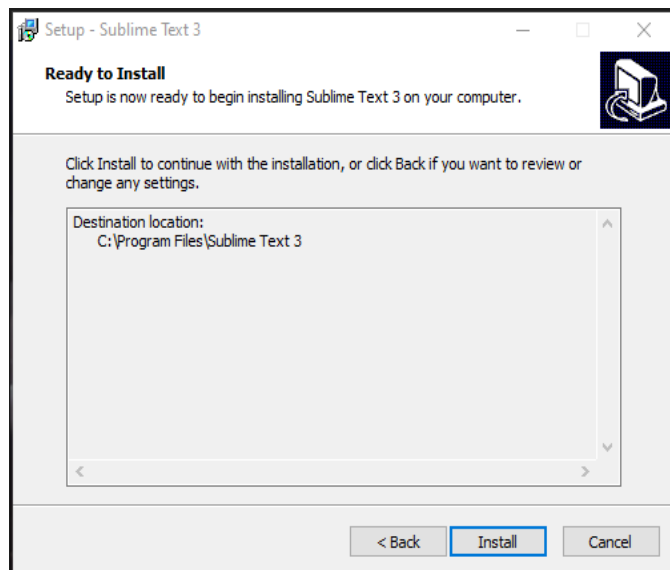
- i. Setelah direktori penyimpanan telah diatur kemudian klik next untuk melanjutkan proses instalasi.



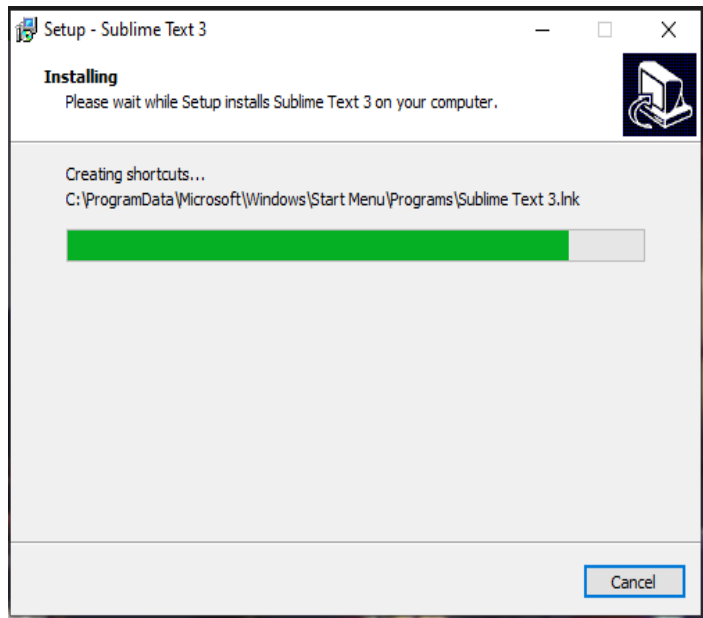
- j. Pada bagian ini terdapat checkbox untuk nge add ke explorer atau bisa langsung next.



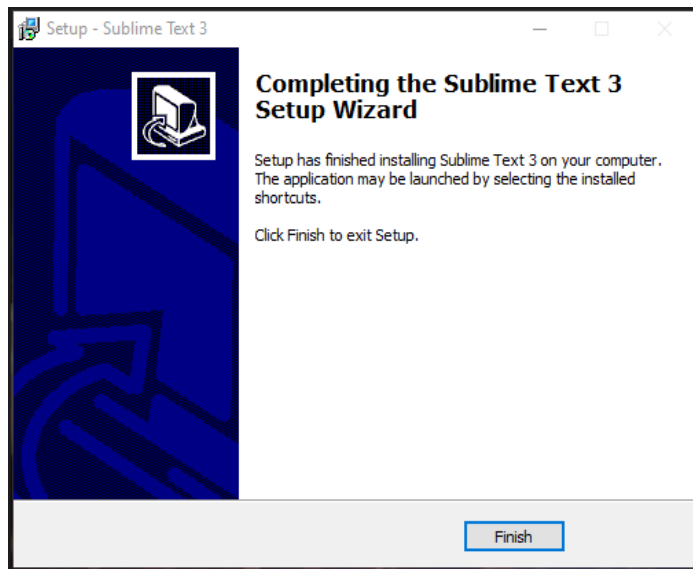
- k. Pada tahap ini bisa langsung klik install untuk melakukan proses instalasinya.



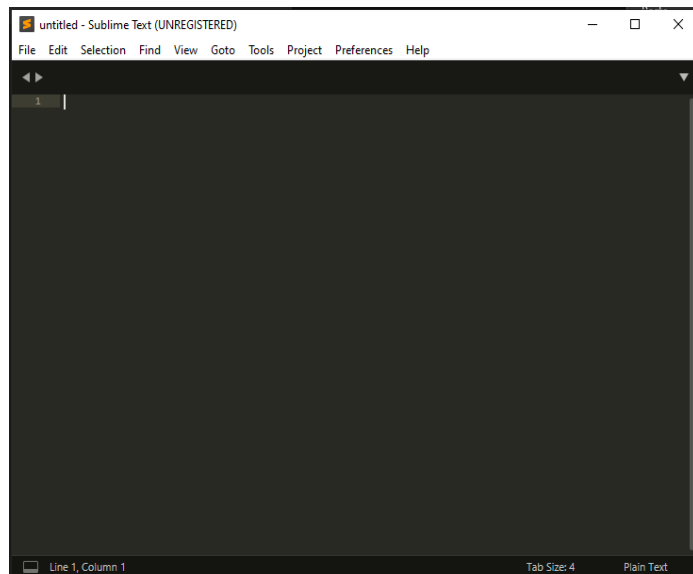
1. Tunggu proses instalasi tersebut sampai selesai



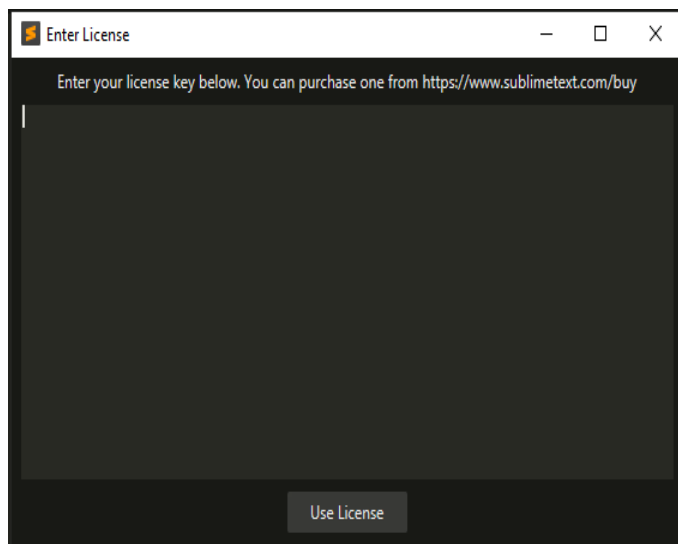
- m. Apabila muncul tampilan seperti dibawah, maka proses instalasi sublime text 3 telah berhasil. Klik finish.



- n. Untuk melihat hasilnya, pada bagian pencarian ketik sublime kemudian buka. Terlihat seperti gambar dibawah merupakan tampilan dari software sublime text 3.



- o. Sublime tersebut sudah bisa digunakan akan tetapi ber status trial, apabila mempunyai serial number bisa di masukan serial number tersebut di bagian *help* kemudian *enter license*.



- p. Jadi apabila mempunyai lisensi sublime, maka bisa dimasukan serial numbernya pada bagian di atas.
- q. Dengan menggunakan sublime dalam status Trial sudah bisa digunakan untuk membuat web, namun fitur-fiturnya terbatas. Akan lebih baik jika mempunyai lisensi sublime nya.

3.3 Instalasi CodeIgniter

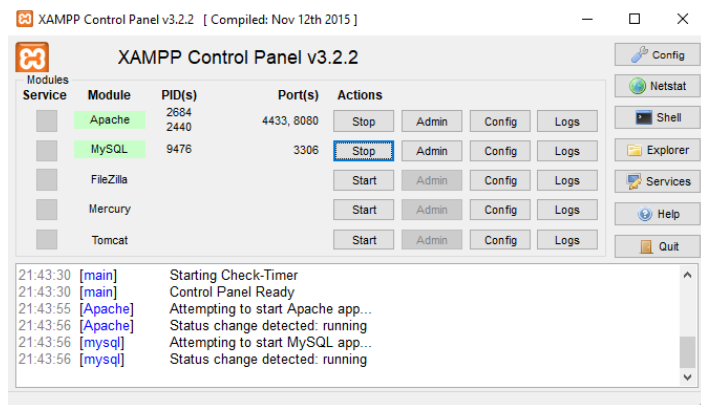
Codeigniter merupakan sebuah *framework* PHP yang menggunakan konsep MVC (*Model, View, Controller*). Adapun beberapa kelebihan dari codeigniter dibandingkan *framework* lain diantaranya yaitu :

- Performa cepat
- Konfigurasi yang minim
- Memiliki banyak komunitas
- Dokumentasi yang lengkap
- Mudah dipelajari bagi pemula

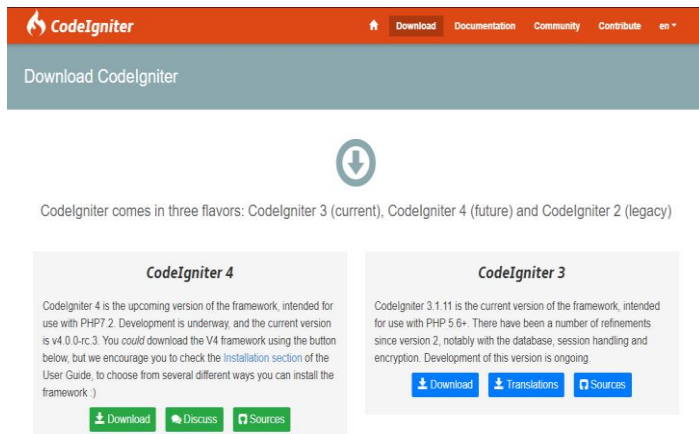
Berikut merupakan tahap-tahap proses instalasi CodeIgniter :

- a. Tahap pertama dalam instalasi codeigniter yaitu jalankan terlebih dahulu aplikasi XAMPP, karena XAMPP tersebut berfungsi sebagai server lokal.

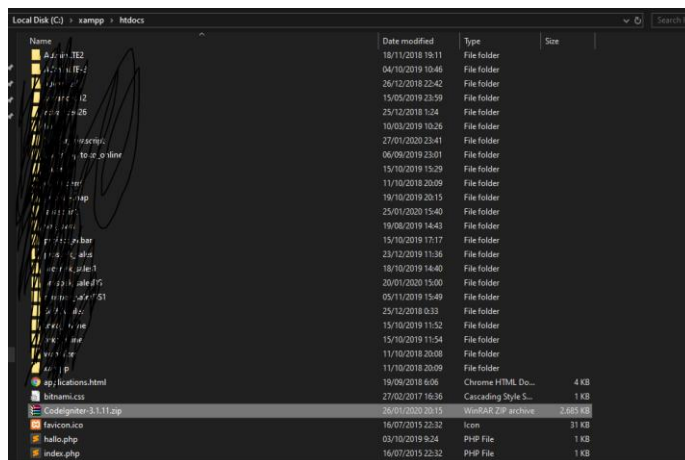
- b. Pada bagian XAMPP kita hanya membutuhkan bagian Apache dan MySQL, maka dari itu jalankan kedua tersebut dengan klik start seperti gambar dibawah ini.



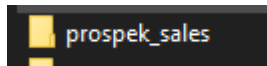
- c. Selanjutnya setelah server lokal dijalankan download terlebih dahulu Codeigniter di web resmi nya atau bisa akses di link <https://codeigniter.com/download> .
- d. Pada halaman download tersebut terdapat dua pilihan yaitu codeigniter 4 dan codeigniter 3. Disini saya menggunakan codeigniter 3, langsung download file nya.



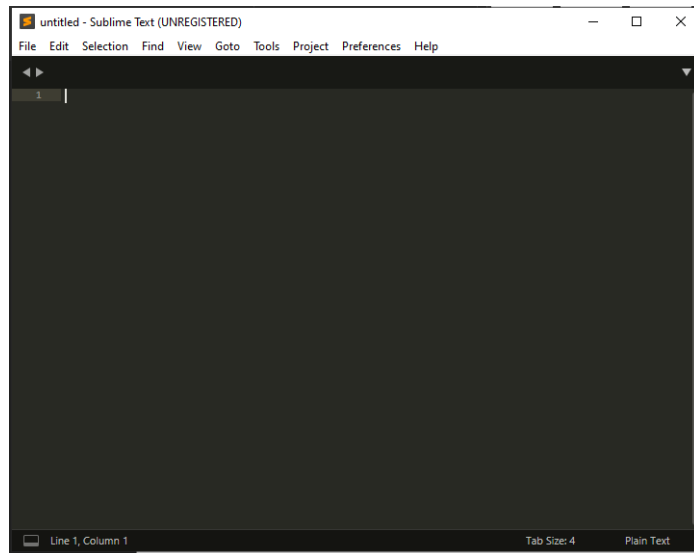
- e. Codeigniter yang berhasil di download pindahkan ke C:\xampp\htdocs supaya aplikasi codeigniter bisa dijalankan di browser.



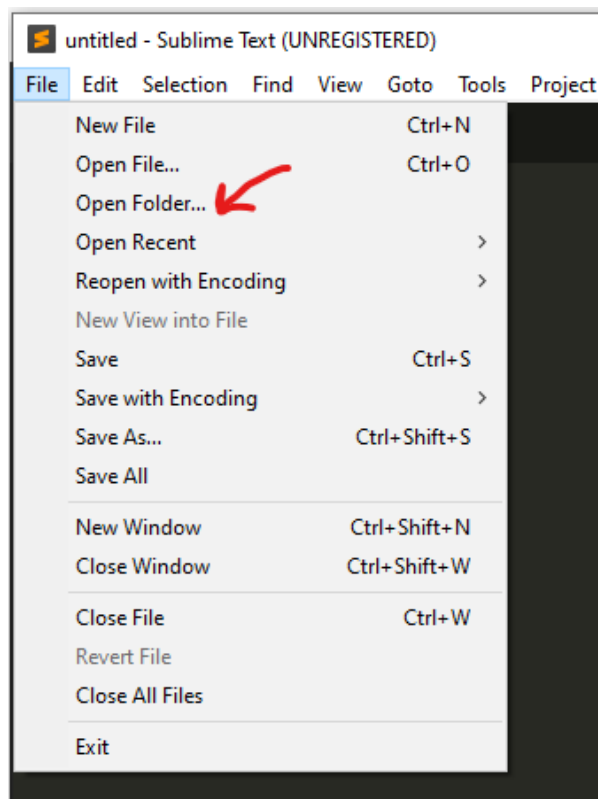
- f. Kemudian extract file tersebut dan rename supaya mudah untuk diakses. sebagai contoh saya merubah nama folder menjadi prospek_sales, sesuai nama aplikasi yang akan dibuat.



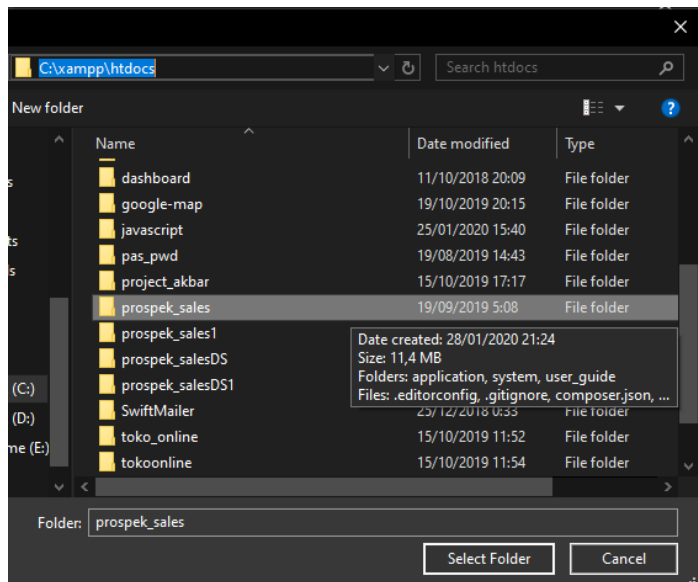
- g. Setelah file codeigniter disiapkan kemudian buka sublime text editor.



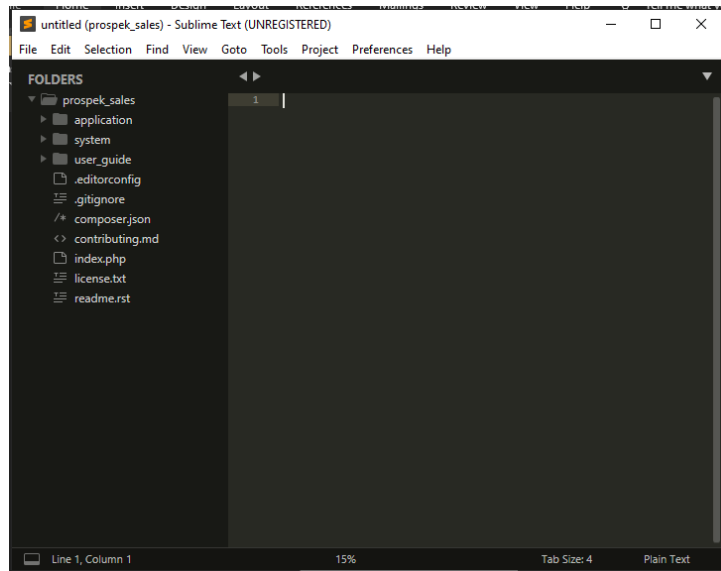
- h. Import file codeigniter tadi di sublime text dengan cara pilih file – kemudian open folder.



- i. Pilih folder `prospek_sales` yang telah tadi siapkan di `C:\xampp\htdocs` , kemudian open folder `prospek_sales` tersebut.

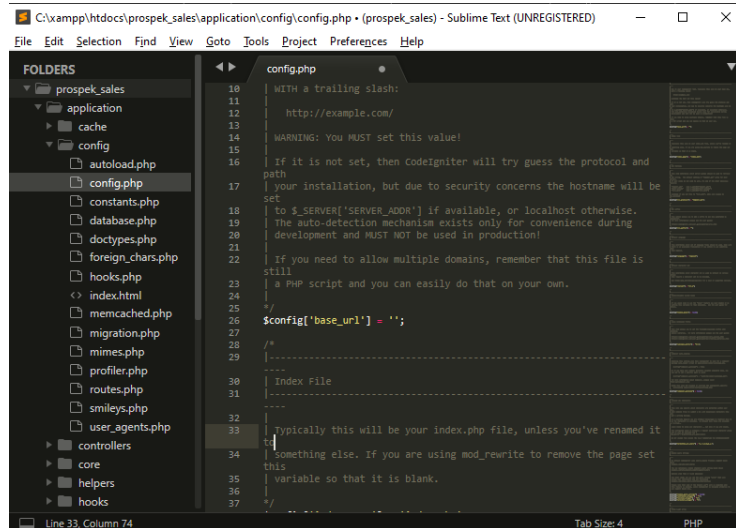


j. Apabila berhasil di import di sublime maka tampilannya seperti berikut :



k. Ada beberapa tahap konfigurasi pada folder **config** yang harus di konfigurasi terlebih dahulu supaya bisa dijalankan pada browser. Diantaranya :

- Perhatikan untuk konfigurasi file **autoload.php**
- Perhatikan untuk konfigurasi file **config.php**



```
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
17 | path
18 | your installation, but due to security concerns the hostname will be
19 | set
20 | to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
21 | The auto-detection mechanism exists only for convenience during
22 | development and MUST NOT be used in production!
23 |
24 | If you need to allow multiple domains, remember that this file is
25 | still
26 | a PHP script and you can easily do that on your own.
27 |
28 | -----
29 | Index File
30 | -----
31 |
32 | Typically this will be your index.php file, unless you've renamed it
33 |
34 | something else. If you are using mod_rewrite to remove the page set
35 | this
36 | variable so that it is blank.
37 |
38 | -----
```

Di file config.php tersebut rubah kode `$config['base_url'] = '';`

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

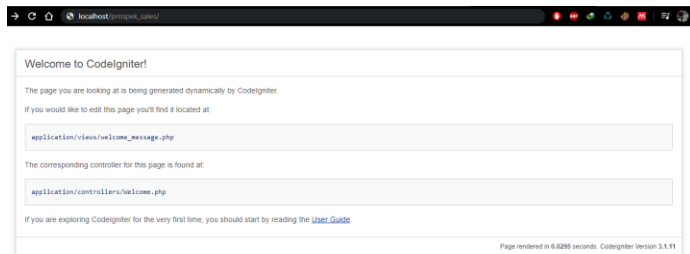
menjadi

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

```
25 */
26 $config['base_url'] = 'http://localhost/prospek_sales/';
27
28 /*
```

- Perhatikan untuk konfigurasi file **database.php**
- Perhatikan untuk konfigurasi file **routes.php**

1. Selanjutnya apabila telah mengkonfigurasinya, buka browser kemudian masukan link http://localhost/prospek_sales/ yg telah di konfigurasi pada bagian config.php tadi.



- m. Jika tampilannya seperti gambar di atas, maka proses instalasi Codeigniter telah berhasil dan siap digunakan untuk pembuatan aplikasi.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis dan Perancangan Sistem

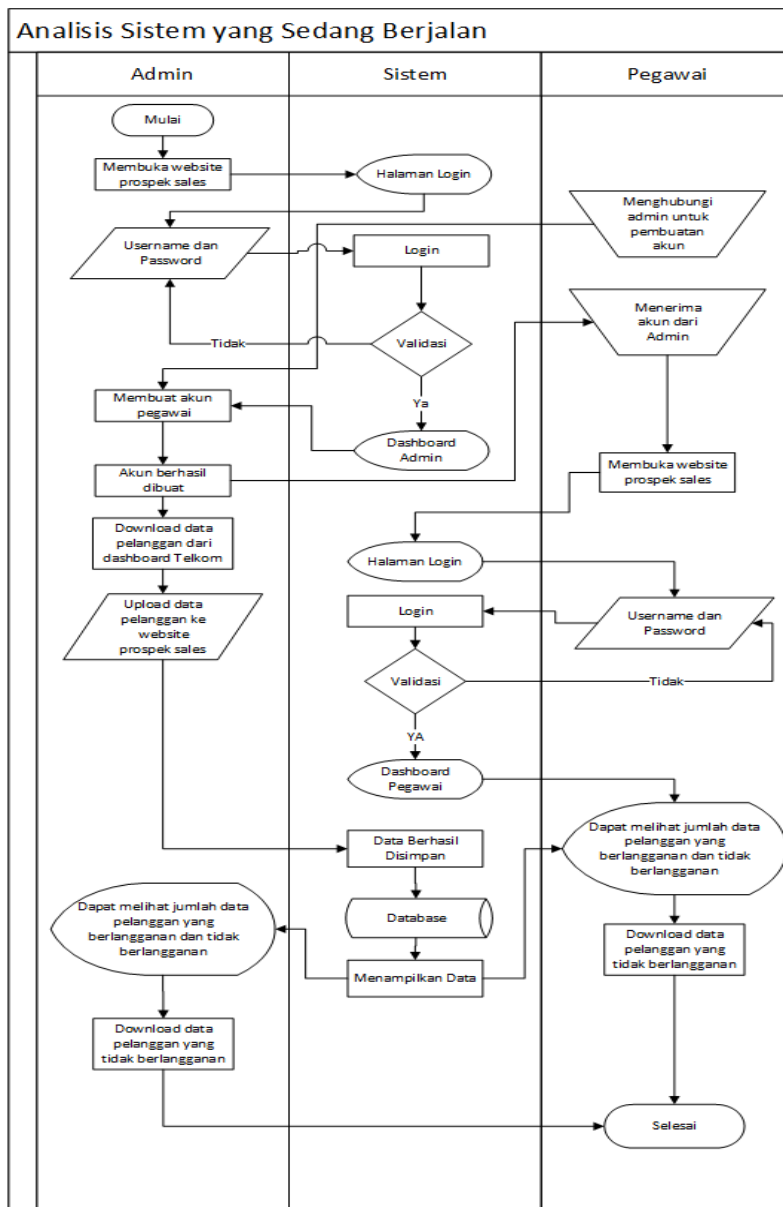
4.1.1 Analisis Sistem Berjalan (*Curent System*)

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.

4.1.1.1 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 merupakan *flowmap Prospek Sales* yang sedang berjalan :



Pada Gambar diatas 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*, 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. 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

4.1.1.2 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:

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

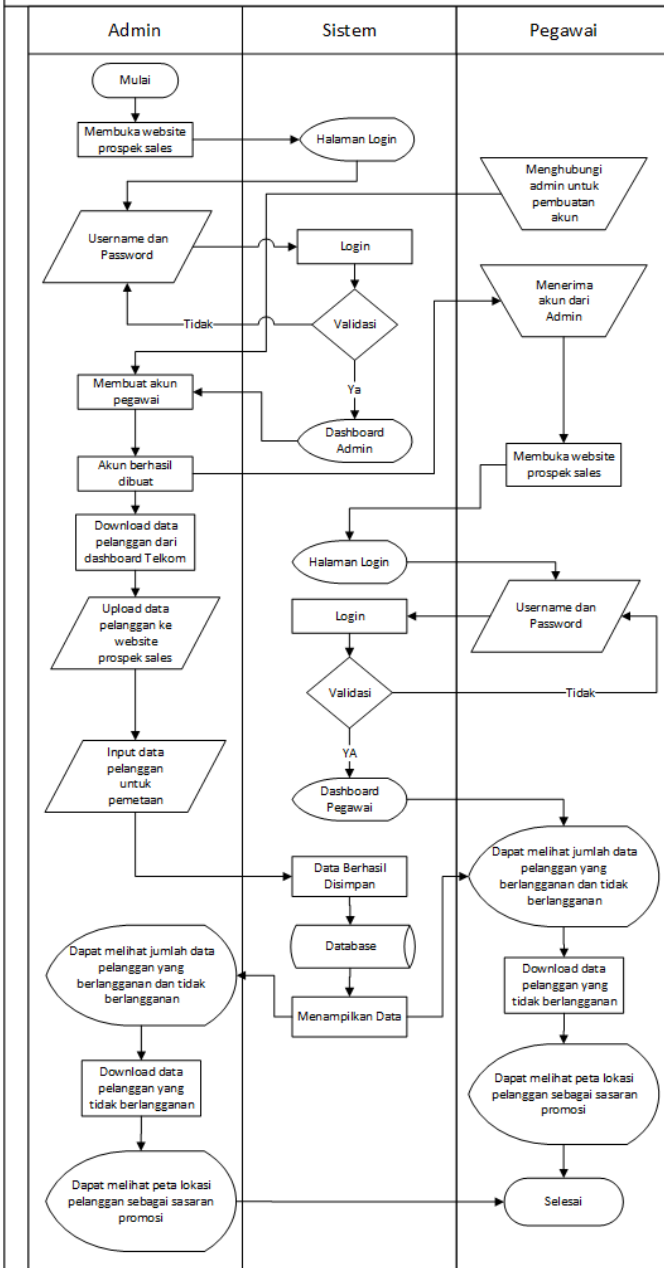
4.1.2 Analisis Sistem yang Akan Dibangun

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*.

4.1.2.1 Analisis Prosedur yang akan Dibangun (*Flowmap/Flowchart*)

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

Analisis sistem yang akan dibangun



Pada Gambar diatas 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.

4.1.2.2 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.

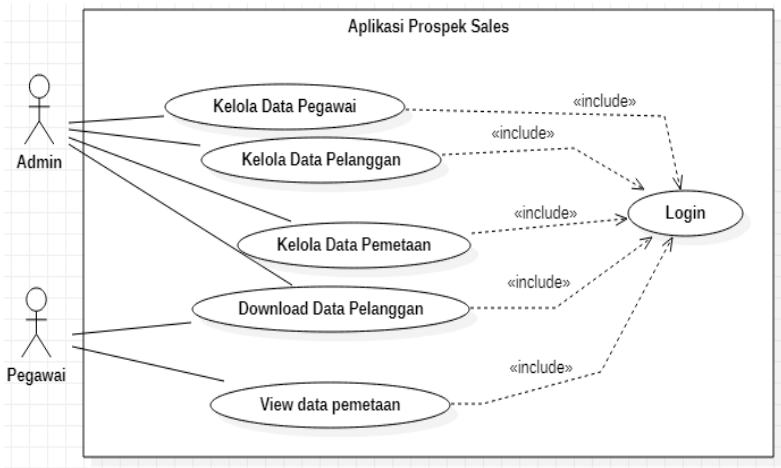
4.1.2.3 UML (*Unified Modeling Language*)

Proses analisis menggunakan UML harus diberikan spesifikasi prosesnya dan dijelaskan dengan tulisan secara lengkap. Ada beberapa tahap proses pada UML diantaranya :

1. *Use Case Diagram*
2. *Class Diagram*
3. *Sequence Diagram*
4. *Activity Diagram*
5. *Collaboration Diagram*
6. *Statechart Diagram*
7. *Component Diagram*
8. *Deployment Diagram*

Berikut contoh tahapan proses dalam pembangunan aplikasi Prospek Sales pada UML :

1. Use Case Diagram



a. Definisi Aktor

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

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

b. Definisi *Use Case*

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

No.	<i>Use Case</i>	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 di aplikasi prospek sales
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 :

- 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

- 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

- 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

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

- 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

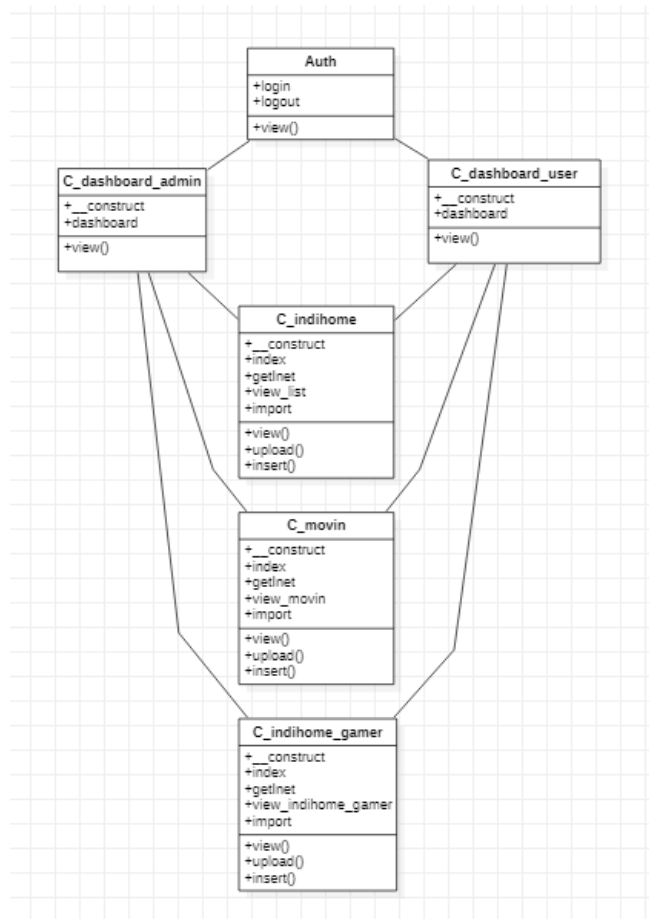
- Skenario *Use Case View* Pemetaan

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

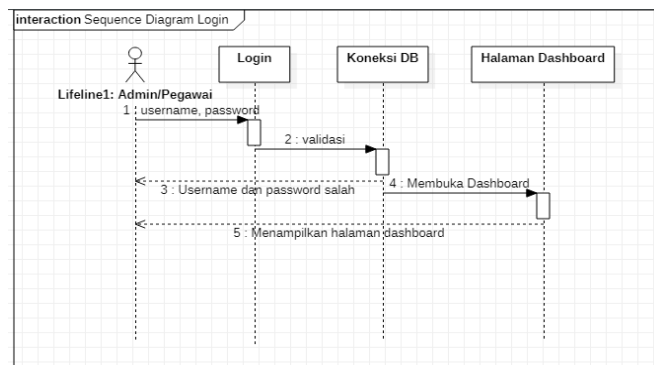
Class Diagram menggambarkan struktur dan hubungan antar objek – objek yang ada pada sistem. Struktur itu meliputi atribut – atribut dan method – method yang ada pada masing – masing kelas.



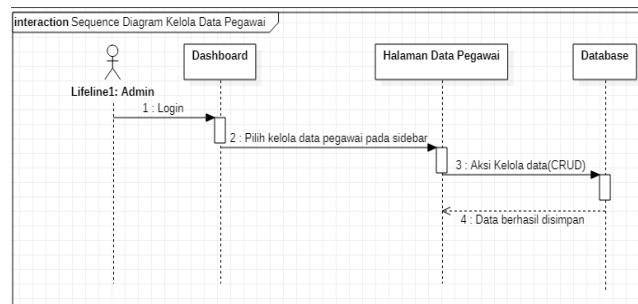
3. Sequence Diagram

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* :

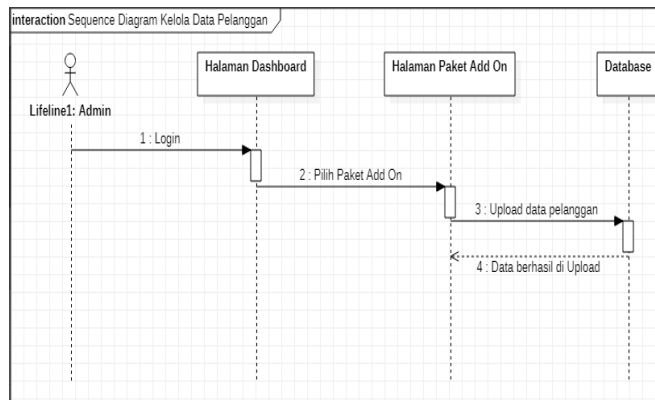
a. Sequence Diagram Login



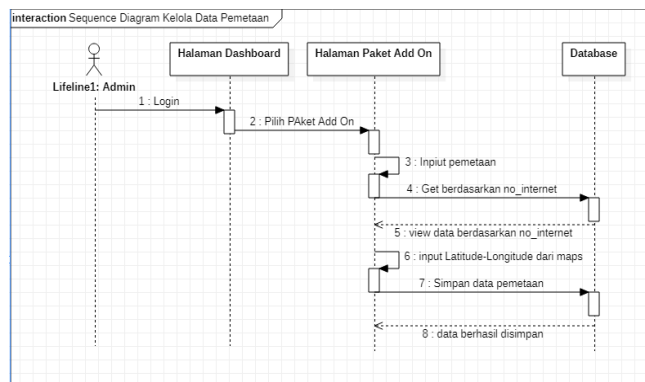
b. Sequence Diagram Kelola Data Pegawai



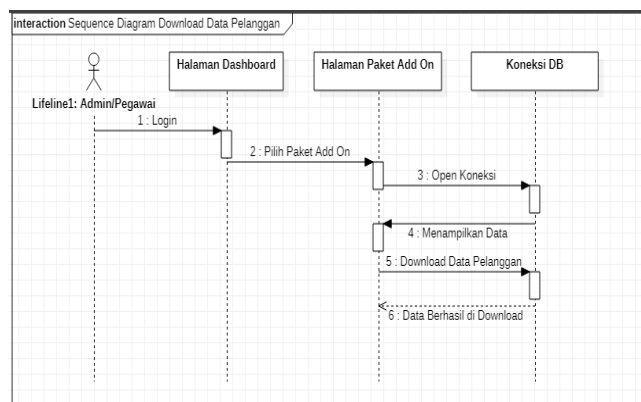
c. Sequence Diagram Kelola Data Pelanggan



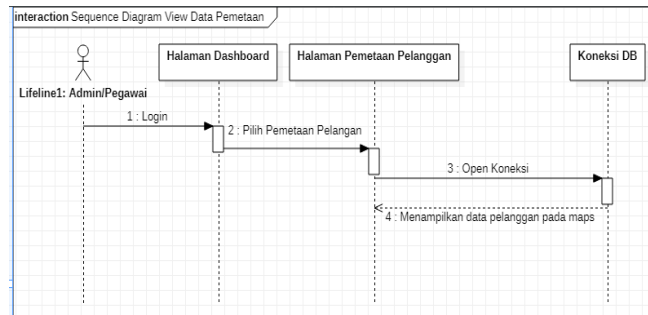
d. *Sequence Diagram Kelola Data Pemetaan*



e. *Sequence Diagram Download Data Pelanggan*



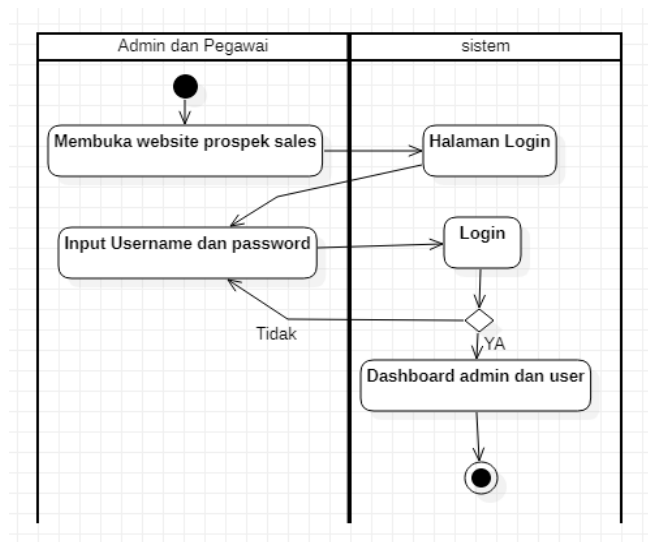
f. *Sequence Diagram view Data Pemetaan*



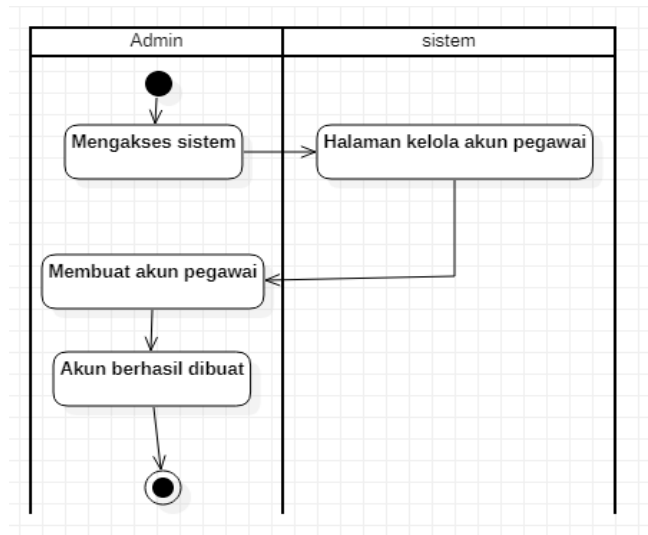
4. *Activity Diagram*

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*:

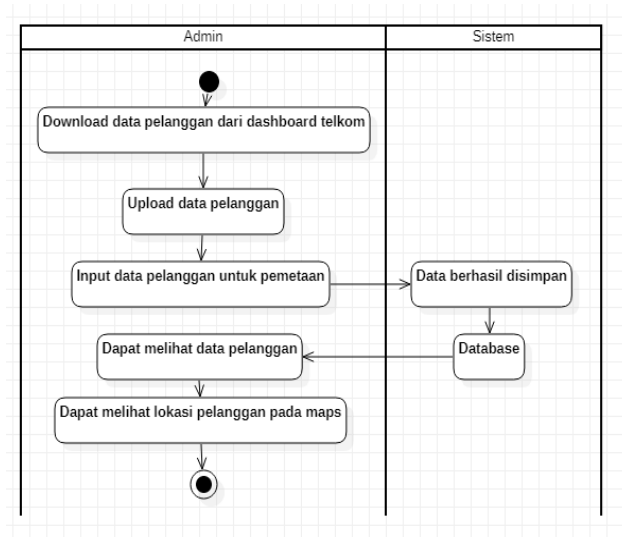
a. *Activity Diagram Login*



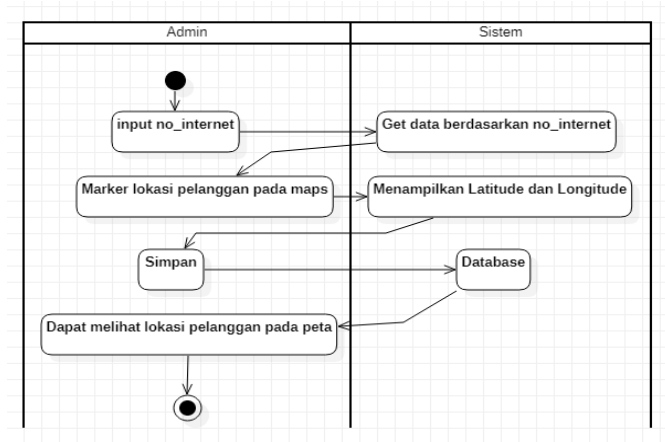
b. *Activity Diagram* Kelola Data Pegawai



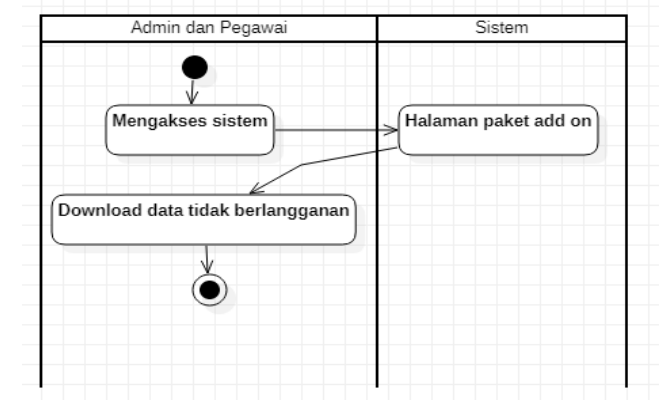
c. *Activity Diagram* Kelola Data Pelanggan



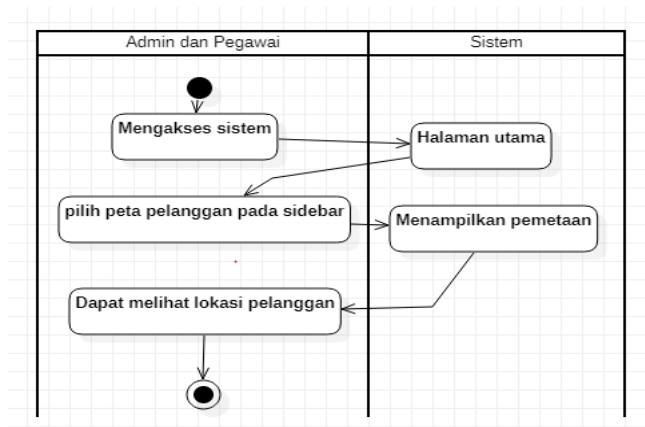
d. *Activity Diagram* Kelola Data Pemetaan



e. *Activity Diagram Download Data Pelanggan*



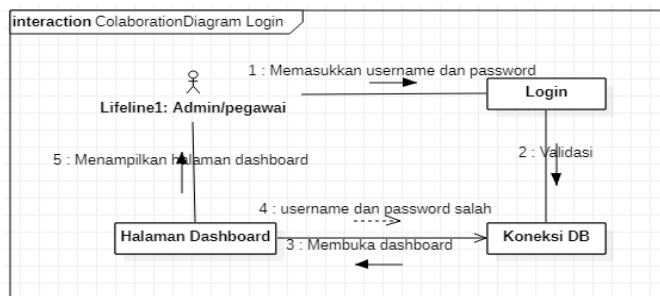
f. *Activity Diagram View Pemetaan*



5. Collaboration Diagram

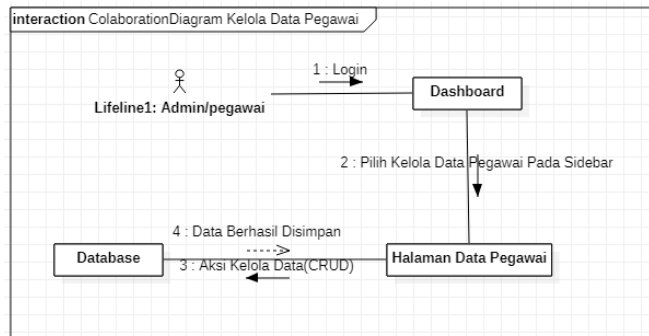
a. Collaboration Diagram Login

Berikut adalah *collaboration diagram* login :

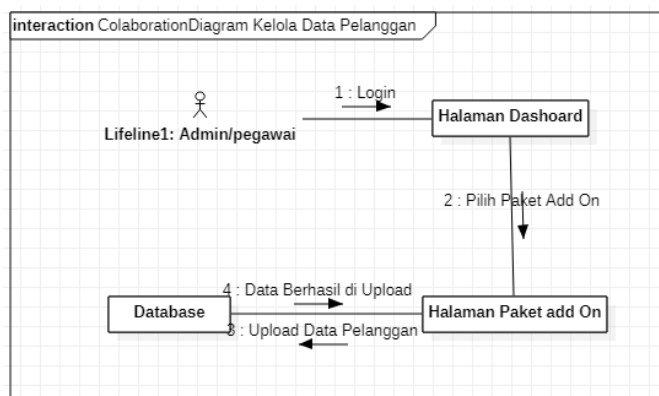


b. Collaboration Diagram Kelola Data Pegawai

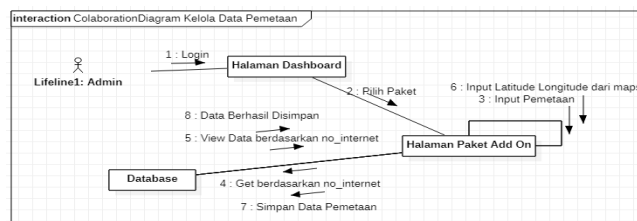
Berikut adalah *collaboration diagram* kelola data pegawai :



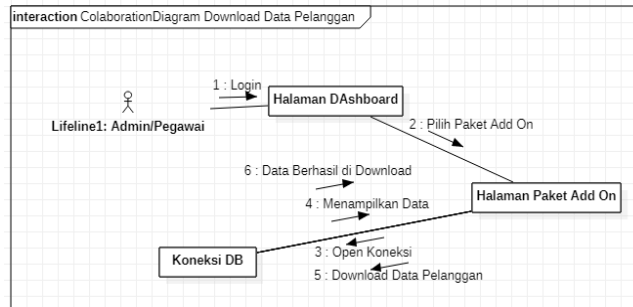
- c. *Collaboration Diagram* Kelola Data Pelanggan
- Berikut adalah *collaboration diagram* kelola data pelanggan :



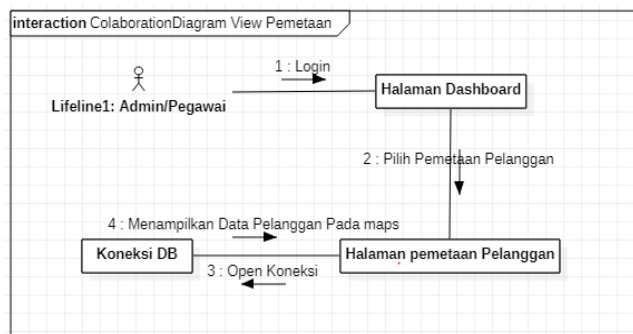
- d. *Collaboration Diagram* Kelola Data Pemetaan
- Berikut adalah *collaboration diagram* kelola data pemetaan :



- e. *Collaboration Diagram Download Data Pelanggan*
Berikut adalah *collaboration diagram download data* pelanggan :



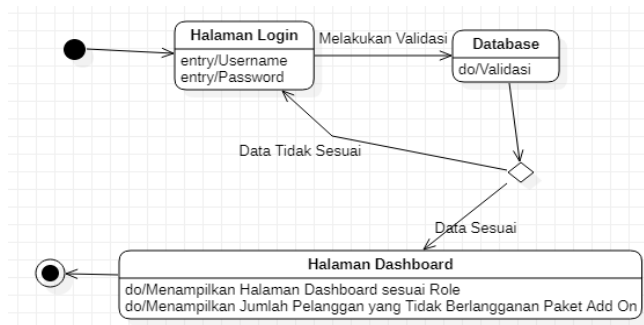
- f. *Collaboration Diagram View Data Pemetaan*
Berikut adalah *collaboration diagram view data* pemetaan :



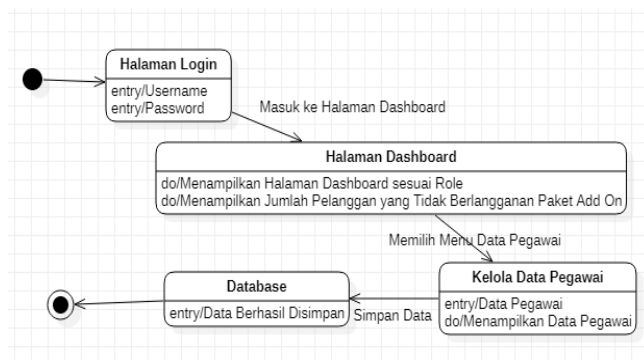
6. Statechart Diagram

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.

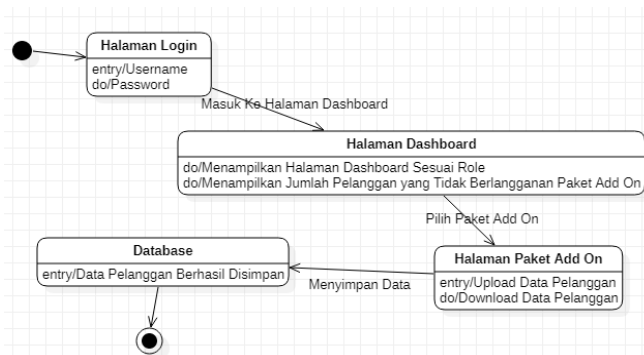
a. *Statechart Diagram Login*



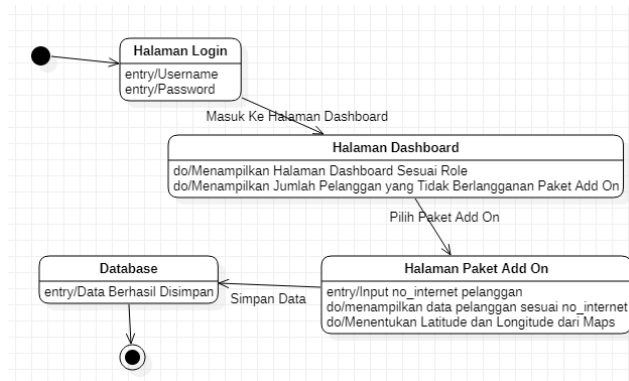
b. *Statechart Diagram Kelola Data Pegawai*



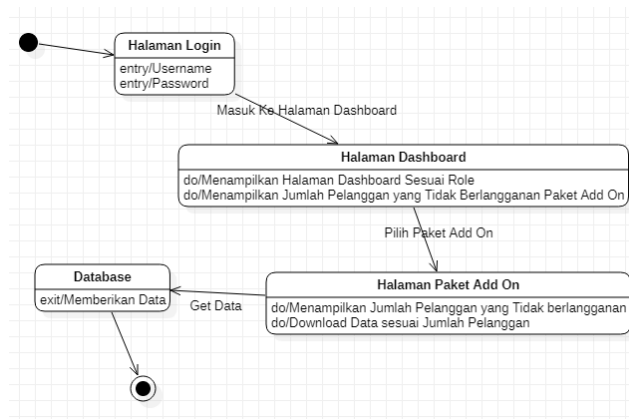
c. *Statechart Diagram Kelola Data Pelanggan*



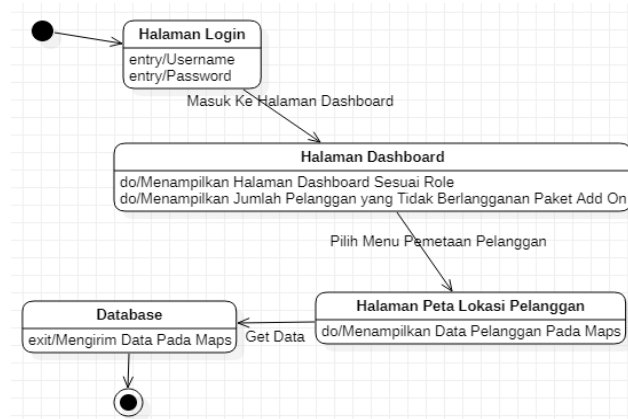
d. *Statechart Diagram Kelola Data Pemetaan*



e. *Statechart Diagram Download Data Pelanggan*

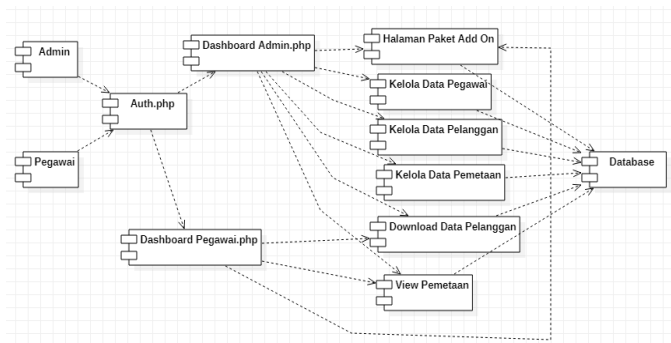


f. *Statechart Diagram View Data Pemetaan*



7. Component Diagram

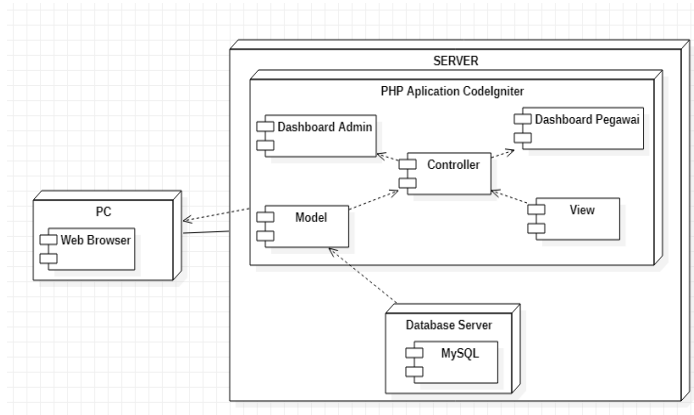
Component diagram digunakan untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.



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.



4.2 Perancangan Basis Data (*Database*)

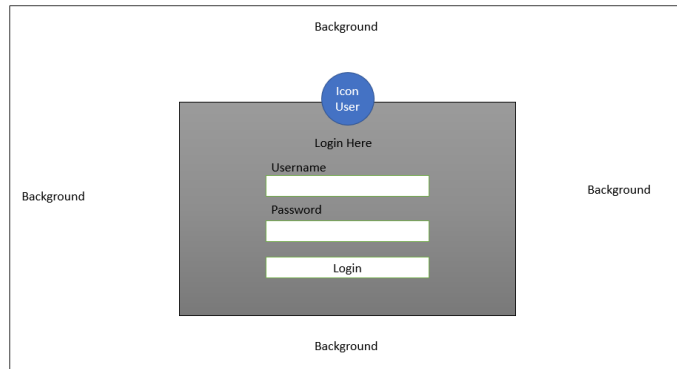
4.3 Perancangan *User Interface* Sistem

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.

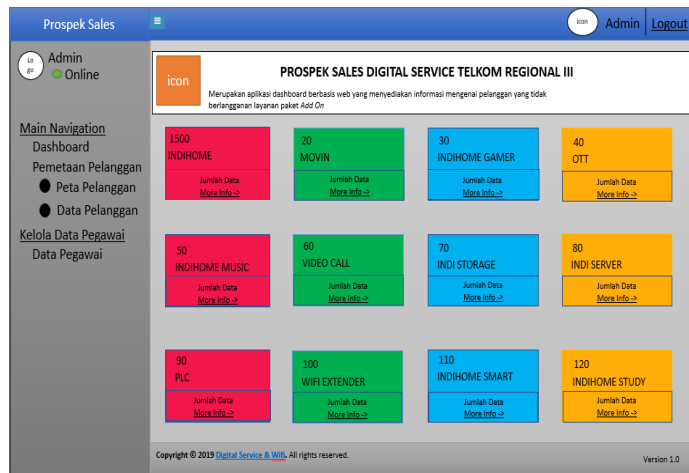
Rancangan *Interface* adalah rancangan pembangunan dari komunikasi antar pemakai (*administrator*) dengan komputer. Antar muka (*interface*) ini terdiri dari proses pemasukan data ke sistem dan menampilkan *output* informasi kepada

administrator. Berikut beberapa gambar bentuk rancangan *user interface* yang sesuai dengan *scoop* penulis:

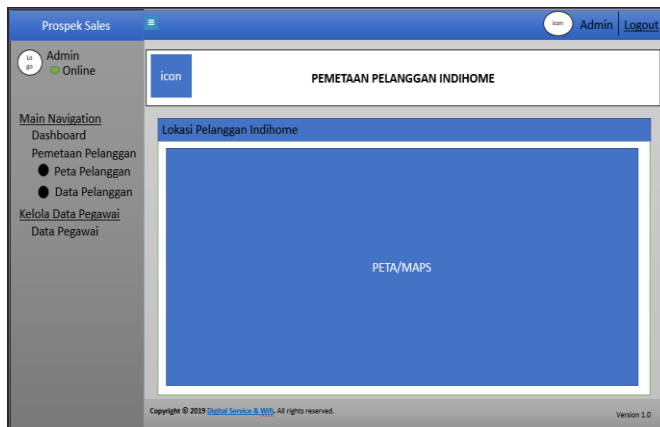
1. *User Interface Login Admin/Pegawai*



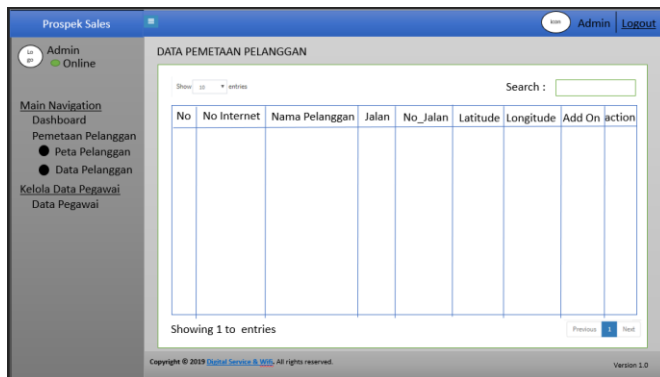
2. *User Interface Dashboard Admin*



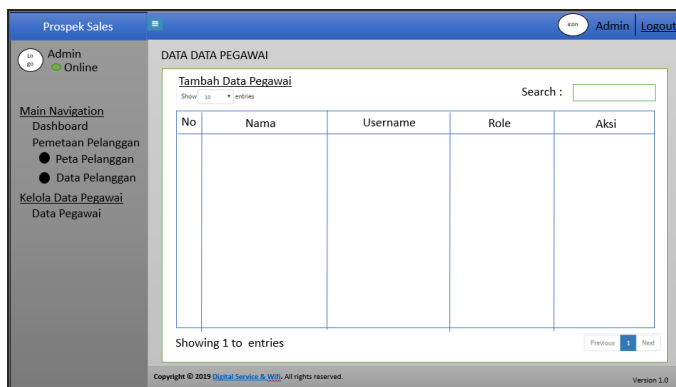
3. *User Interface Halaman Peta Pelanggan*



4. *User Interface* Halaman Data Pelanggan



5. *User Interface* Halaman Data Pegawai



6. *User Interface* Tambah Data Pegawai

Form Input Data Pegawai

Nama Pegawai

Username

Password

Role

--Pilih Role--

Reset

Simpan

7. *User Interface* Halaman Indihome Admin

Prospek Sales

Admin Online

Main Navigation

Dashboard

Pemetaan Pelanggan

● Peta Pelanggan

● Data Pelanggan

Kelola Data Pegawai

Data Pegawai

DATA INDIHOME

Upload Data +

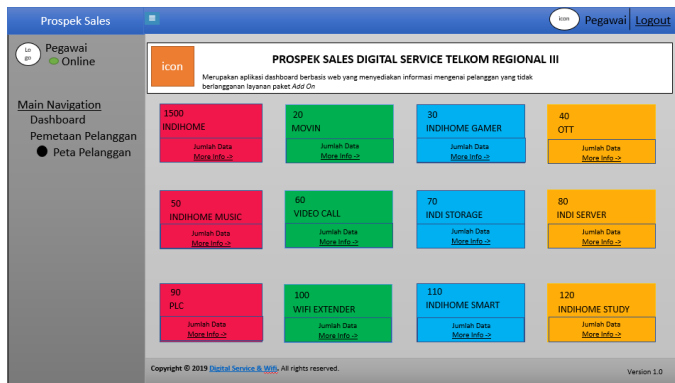
Showing 10 entries

Search :

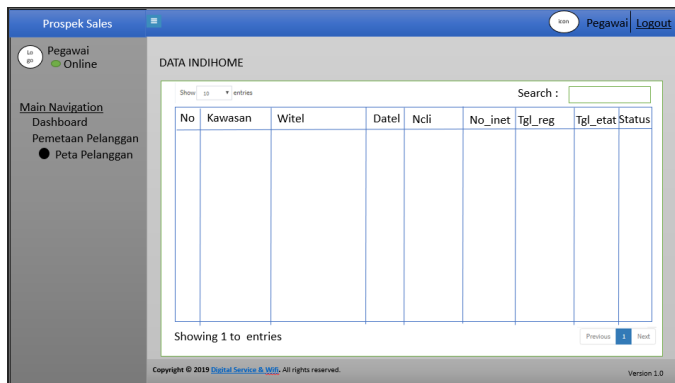
No	Kawasan	Witel	Datel	Ncli	No_inet	Tgl_reg	Tgl_etat	Status

Showing 1 to entries

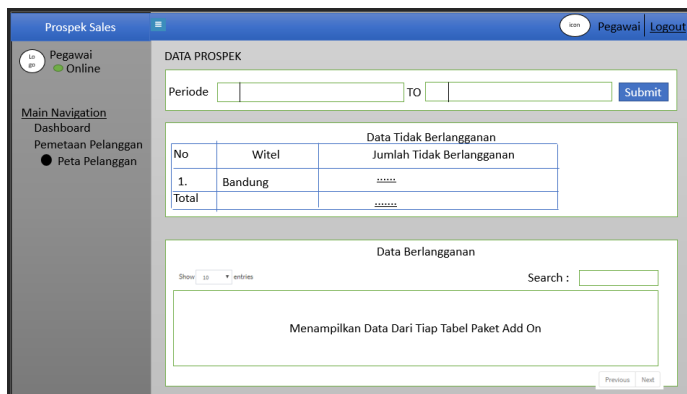
Previous 1 Next



10. User Interface Halaman Indihome Pegawai



11. User Interface Halaman Paket Add On Pegawai



BAB V

IMPLEMENTASI DAN HASIL IMPLEMENTASI

5.1 Implementasi

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

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

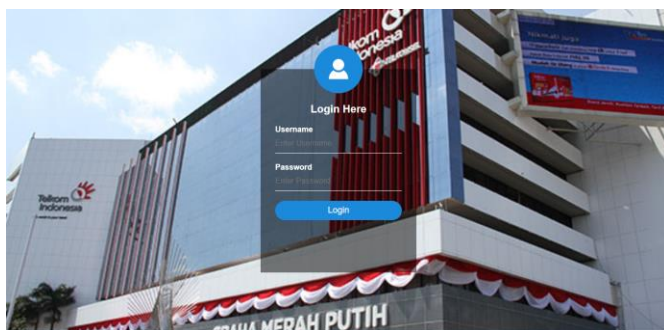
Pada pembuatan aplikasi Prospek Sales ini terdapat perangkat keras pendukung yang digunakan adalah sebagai berikut :

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>

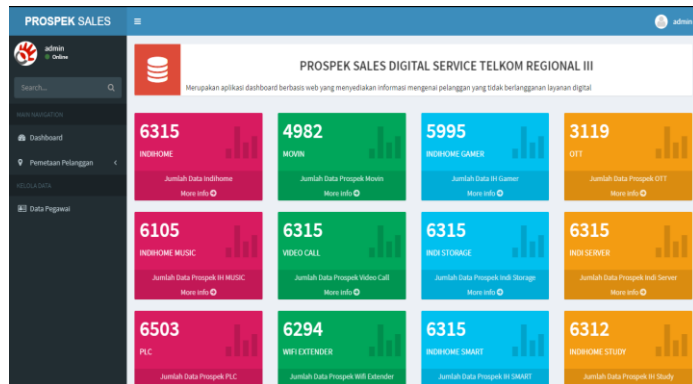
5.2 Hasil Implementasi

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



b. Halaman *Dashboard* Admin



c. Halaman Peta Pelanggan



d. Halaman Data Pemetaan Pelanggan

PROSPEK SALES

admin

Dashboard

Pelanggan Pelanggan

DATA PELANGGAN

Home

Active Data

Data Pelanggan

Tampil Data Pelanggan

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

10

Show

10

entries

Search:

NO	INTERNET	NAMA PELANGGAN	JULAN	NO JULAN	DISTRIK	KOTA	LATITUDE	LONGITUDE	ADD ON	Action
1.	131188115601	MIRNA HASAN BAGIR	W/1X30N 01		CPEDES	KOTA BANGUNG	-4.832212645959086	108.23310522255562	MOVIN	<div></div> <div></div> <div></div>
2.	1311881156627	DAISY LUKMAN	Gg W/ RODO		GEERAKALANG	KOTA BANGUNG	-4.826444161150334	108.23523232648539	INCHOCHE	<div></div> <div></div> <div></div>
3.	131188118651	RINI OKTAWANA	KARANG TINEJANG DALAM		CPEDES	KOTA BANGUNG	-4.827135430706232	108.22259388360302	MOVIN	<div></div> <div></div> <div></div>
4.	131188118651	RINI OKTAWANA	KARANG TINEJANG SIALAT	W/	CPEDES	KOTA BANGUNG	-4.8322247483241568	108.23386113938064	INCHOCHE	<div></div> <div></div> <div></div>

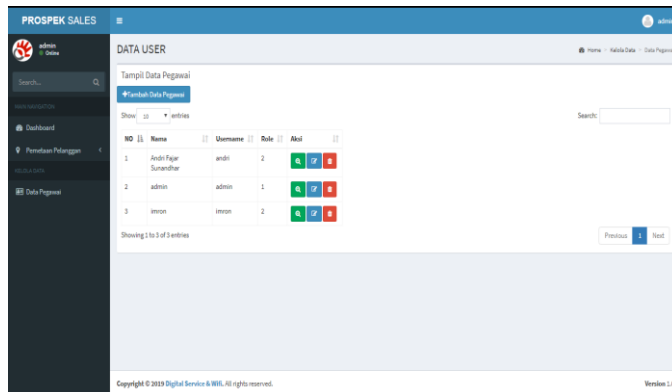
Showing 1 to 4 of 4 entries

Previous

1

Next

e. Halaman Data Pegawai



PROSPEK SALES

DATA USER

Tampilkan Data Pegawai

+ Tambah Data Pegawai

Show 3 entries

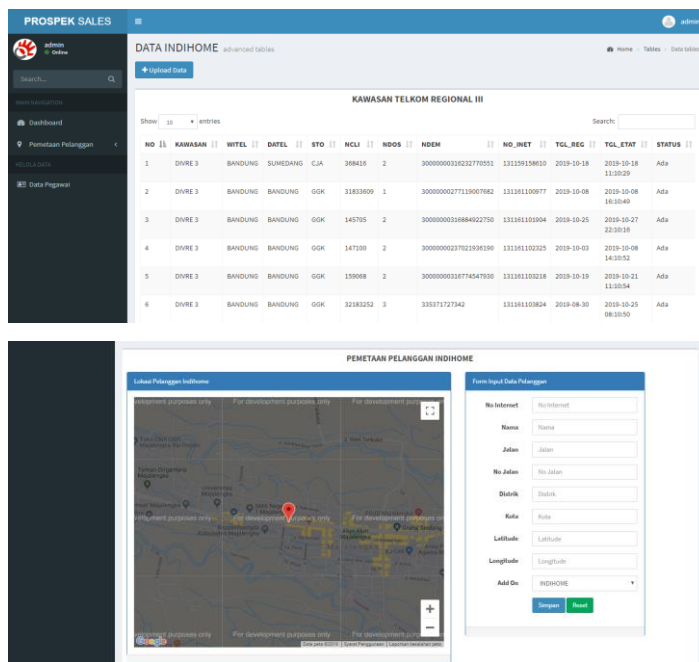
NO	Nama	Username	Role	Aksi
1	Andi Fajar Sumardhar	andi	2	Edit Delete
2	admin	admin	1	Edit Delete
3	imron	imron	2	Edit Delete

Showing 1 to 3 of 3 entries

Previous 1 Next

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

f. Halaman Indihome Admin



PROSPEK SALES

DATA INDIHOME

Upload Data

KAWASAN TELKOM REGIONAL III

Show 6 entries

NO	KAWASAN	WITEL	DATEL	STO	NCLI	NDOS	NDEM	NO_INET	TGL_REG	TGL_ETAT	STATUS
1	DIVRE 3	BANDUNG	SUMEDANG	CJA	358418	2	30000000316322770551	131159158610	2019-10-18	2019-10-18 11:09:29	Ada
2	DIVRE 3	BANDUNG	BANDUNG	GGK	31833809	1	3000000027711007682	131161100877	2019-10-08	2019-10-08 16:10:49	Ada
3	DIVRE 3	BANDUNG	BANDUNG	GGK	145705	2	3000000031884922750	131161101904	2019-10-25	2019-10-27 22:19:16	Ada
4	DIVRE 3	BANDUNG	BANDUNG	GGK	147100	2	30000000237021936190	131161102325	2019-10-03	2019-10-08 14:10:52	Ada
5	DIVRE 3	BANDUNG	BANDUNG	GGK	159068	2	30000000326774247930	131161103218	2019-10-19	2019-10-21 11:10:54	Ada
6	DIVRE 3	BANDUNG	BANDUNG	GGK	32183252	3	335371727342	131161103824	2019-08-30	2019-10-25 06:10:50	Ada

PEMETAAN PELANGGAN INDIHOME

Lokasi Pelanggan Indihome

Form Input Data Pelanggan

No Internet: No Internet

Nama: Nama

Jalan: Jalan

No Jalan: No Jalan

Dititik: Dititik

Kota: Kota

Latitude: Latitude

Longitude: Longitude

Add On: INDIHOME

Simpan Batalkan

g. Halaman Paket Add On Admin

PROSPEK SALES admin

DATA PROSPEK

PERIODE : TO

DATA TIDAK BERLANGGANAN MOVIN

NO	WITEL	Jumlah tidak berlangganan
1	BANDUNG	6315
TOTAL		6315

Terakhir di upload : admin, UPLoad MOVIN, 2019-12-30 04:34:20

PEMETAAN PELANGGAN MOVIN

Lokasi Pelanggan Movin

Form Input Data Pelanggan

No Internet:

Nama:

Jalan:

No Jalan:

Dititik:

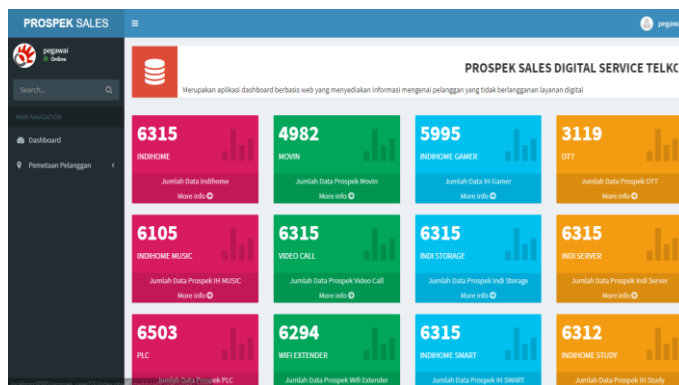
Kota:

Latitude:

Longitude:

Add On:

h. Halaman *Dashboard* Pegawai



i. Halaman *Indihome* Pegawai

PROSPEK SALES

DATA INDIHOME advanced tables

Upload Data

KAWASAN TELKOM REGIONAL III

Show 10 entries

NO	KAWASAN	WITEL	DATEL	STO	NCLI	NDOS	NDGM	NO_INET	TGL_REG	TGL_ETAT	STATUS
1	DIVRE 3	BANDUNG	SUHEDANG	CJA	368416	2	30000000316231770551	131159159610	2019-10-18	2019-10-18 11:10:29	Ada
2	DIVRE 3	BANDUNG	BANDUNG	GOK	31833909	1	30000000271139070882	131161100977	2019-10-08	2019-10-08 16:10:49	Ada
3	DIVRE 3	BANDUNG	BANDUNG	GOK	143705	2	30000000318884922750	131161101394	2019-10-25	2019-10-27 22:10:16	Ada
4	DIVRE 3	BANDUNG	BANDUNG	GOK	147289	2	30000000237021936190	131161102325	2019-10-03	2019-10-08 14:10:52	Ada
5	DIVRE 3	BANDUNG	BANDUNG	GOK	159068	2	30000000316774547930	131161103218	2019-10-19	2019-10-21 11:10:54	Ada
6	DIVRE 3	BANDUNG	BANDUNG	GOK	3218252	3	335371727342	131161103824	2019-08-30	2019-10-25 08:10:50	Ada

j. Halaman Paket *Add On* Pegawai

PROSPEK SALES

DATA PROSPEK

PERIODE 1 TO 2 Submit

DATA TIDAK BERLANGGANAN MOVIN

NO	WITEL	Jumlah tidak berlangganan
1	BANDUNG	625
TOTAL		625

BAB VI

PENGUNAAN APLIKASI PROSPEK SALES

6.1 Admin

6.2 Pegawai

1. Config

A. autoload.php

```
<?php
defined('BASEPATH') OR exit('No direct script access
allowed');
$autoload['packages'] = array();
$autoload['libraries'] =
array('database','session','form_validation','googlemaps');
$autoload['drivers'] = array();
$autoload['helper'] = array('url', 'form');
$autoload['config'] = array();
$autoload['language'] = array();
$autoload['model'] = array('model_auth');
```

B. config.php

```
<?php
defined('BASEPATH') or exit('No direct script access
allowed');
$config['base_url'] = 'http://localhost:8080/prospek_salesDS/';
$config['index_page'] = 'index.php';
$config['url_suffix'] = "";
$config['language'] = 'english';
$config['charset'] = 'UTF-8';
$config['enable_hooks'] = FALSE;
$config['composer_autoload'] = FALSE;
$config['enable_query_strings'] = FALSE;
$config['controller_trigger'] = 'c';
```

```
$config['function_trigger'] = 'm';  
$config['directory_trigger'] = 'd';  
$config['allow_get_array'] = TRUE;  
$config['log_threshold'] = 0;
```

C. database.php

```
<?php  
defined('BASEPATH') OR exit('No direct script access  
    allowed');  
$active_group = 'default';  
$query_builder = TRUE;  
  
$db['default'] = array(  
    'dsn' => "",  
    'hostname' => 'localhost',  
    'username' => 'root',
```

```
    'password' => "",  
    'database' => 'sales',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => "",  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
    'cache_on' => FALSE,  
    'cachedir' => "",  
    'char_set' => 'utf8',  
    'dbcollat' => 'utf8_general_ci',  
    'swap_pre' => "",  
    'encrypt' => FALSE,  
    'compress' => FALSE,  
    'stricton' => FALSE, //rubah true untuk hosting  
    'failover' => array(),  
    'save_queries' => TRUE  
);
```

D. routes.php

```
<?php
defined('BASEPATH') OR exit('No direct script access
    allowed');
$route['default_controller'] = 'Auth/login';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
```

2. Controller

A. Auth.php

```
<?php

class Auth extends CI_Controller{

    public function login()
    {
        $this->form_validation-
        >set_rules('username','Username','required',[
            'required'      => 'Username wajib
            diisi!'
        ]);
        $this->form_validation-
        >set_rules('password','Password','required',[
            'required'      => 'Password wajib
            diisi!'
        ]);
        if ($this->form_validation->run() == FALSE)
        {
            $this->load->view('login2');

        }else{
```

```

        $auth = $this->model_auth-
>cek_login();
        if($auth == FALSE)
        {
            $this->session-
>set_flashdata('pesan','<div class="alert alert-danger"
role="alert">
<strong>Username atau Password anda Salah!!!</strong>
<button type="button" class="close" data-dismiss="alert"
aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>');
            redirect('Auth/login');
        }else{
            $this->session-
>set_userdata('username',$auth->username);
            $this->session-
>set_userdata('role_id',$auth->role_id);

            switch($auth->role_id){
                case 1 :
                    redirect('C_dashboard_admin/dashboard');

                    break;

                case 2 :
                    redirect('C_dashboard_user/dashboard');

                    break;

                default: break;

            }

        }

    }
}

```

```

        public function logout()
        {
            $this->session->sess_destroy();
            redirect('Auth/login');
        }
    }

    ?>

```

B. C_dashboard_admin.php

```

<?php
defined('BASEPATH') OR exit('No direct script access
    allowed');

class C_dashboard_admin extends CI_Controller {

    /**
     * Index Page for this controller.
     *
     * Maps to the following URL
     *      http://example.com/index.php/welcome
     *  - or -
     *
     *      http://example.com/index.php/welcome/index
     *  - or -
     * Since this controller is set as the default controller
     in
     * config/routes.php, it's displayed at
     http://example.com/
     *
     * So any other public methods not prefixed with an
     underscore will
     * map to /index.php/welcome/<method_name>
     * @see
     https://codeigniter.com/user_guide/general/urls.html
     */

```

```

public function __construct(){
    parent::__construct();
    $this->load->model('m_dashboard_admin');

    if($this->session->userdata('role_id')!='1'){
        $this->session-
        >set_flashdata('pesan','<div class="alert alert-danger"
        role="alert">
        <strong>Anda belum Login !!!</strong>
        <button type="button" class="close" data-dismiss="alert"
        aria-label="Close">
        <span aria-hidden="true">&times;</span></button>
        </div>');
        redirect('auth/login');
    }
}

public function dashboard()
{
    $data['indihome'] = $this-
    >m_dashboard_admin->indihome();
    $data['movin'] = $this-
    >m_dashboard_admin->movin();
    $data['indihome_gamer'] = $this-
    >m_dashboard_admin->indihome_gamer();
    $data['ott'] = $this->m_dashboard_admin-
    >ott();
    $data['indihome_music'] = $this-
    >m_dashboard_admin->indihome_music();
    $data['video_call'] = $this-
    >m_dashboard_admin->video_call();
    $data['indi_storage'] = $this-
    >m_dashboard_admin->indi_storage();
    $data['indi_server'] = $this-
    >m_dashboard_admin->indi_server();
}

```

```

        $data['plc'] = $this-
>m_dashboard_admin->plc();
        $data['wifi_extender'] = $this-
>m_dashboard_admin->wifi_extender();
        $data['indihome_smart'] = $this-
>m_dashboard_admin->indihome_smart();
        $data['indihome_study'] = $this-
>m_dashboard_admin->indihome_study();

        $this->load->view('templates/header');
        $this->load->view('templates/sidebar');
        $this->load->view('v_dashboard_admin',
        $data);
        $this->load->view('templates/footer');

    }
}

```

C. C_dashboard_user.php

```

<?php
defined('BASEPATH') OR exit('No direct script access
        allowed');

class C_dashboard_user extends CI_Controller {

    /**
     * Index Page for this controller.
     *
     * Maps to the following URL
     *      http://example.com/index.php/welcome
     *    - or -
     *
     *      http://example.com/index.php/welcome/index
     *    - or -
     * Since this controller is set as the default controller
     in

```

```

* config/routes.php, it's displayed at
http://example.com/
*
* So any other public methods not prefixed with an
underscore will
* map to /index.php/welcome/<method_name>
* @see
https://codeigniter.com/user_guide/general/urls.html
*/
public function __construct(){
    parent::__construct();
    $this->load->model('m_dashboard_user');

    if($this->session->userdata('role_id')!='2'){
        $this->session-
>set_flashdata('pesan','<div class="alert alert-danger"
role="alert">
<strong>Anda belum Login !!!</strong>
<button type="button" class="close" data-dismiss="alert"
aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>');
        redirect('auth/login');
    }
}
public function dashboard()
{
    $data['indihome'] = $this-
>m_dashboard_user->indihome();
    $data['movin'] = $this-
>m_dashboard_user->movin();
    $data['indihome_gamer'] = $this-
>m_dashboard_user->indihome_gamer();
    $data['ott'] = $this-
>m_dashboard_user->ott();
}

```



```

        $data['indihome_music'] = $this-
>m_dashboard_user->indihome_music();
        $data['video_call'] = $this-
>m_dashboard_user->video_call();
        $data['indi_storage'] = $this-
>m_dashboard_user->indi_storage();
        $data['indi_server'] = $this-
>m_dashboard_user->indi_server();
        $data['plc'] = $this-
>m_dashboard_user->plc();
        $data['wifi_extender'] = $this-
>m_dashboard_user->wifi_extender();
        $data['indihome_smart'] = $this-
>m_dashboard_user->indihome_smart();
        $data['indihome_study'] = $this-
>m_dashboard_user->indihome_study();

        $this->load->view('templates/header');
        $this->load->view('templates/sidebar_user');
        $this->load->view('v_dashboard_user', $data);
        $this->load->view('templates/footer');

    }
}

```

D. C_data_user.php

```

<?php
defined('BASEPATH') or exit('No direct script access
    allowed');

class C_data_user extends CI_Controller
{

    public function __construct()
    {
        parent::__construct();
    }
}

```

```

$this->load->model('m_data_user');

if ($this->session->userdata('role_id') != '1') {
    $this->session->set_flashdata('pesan', '<div
class="alert alert-danger" role="alert">
<strong>Anda belum Login !!!</strong>
<button type="button" class="close" data-
dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>');
    redirect('auth/login');
}
}

public function index()
{
    $data['data_user'] = $this->m_data_user->tampil_data()-
>result();

    $this->load->view('templates/header');
    $this->load->view('templates/sidebar');
    $this->load->view('data_user/v_data_user', $data);
    $this->load->view('templates/footer');
}

public function tambah_aksi()
{
    $nama        = $this->input->post('nama');
    $username     = $this->input->post('username');
    $password     = $this->input->post('password');
    $role_id     = $this->input->post('role_id');

    $data = array(
        'nama'      => $nama,
        'username'  => $username,

```

```

        'password'    => $password,
        'role_id'     => $role_id
    );

    $this->m_data_user->input_data($data, 'login');
    // $this->session->set_flashdata('message', '
    // <div class="alert alert-success alert-dismissible"
    //     role="alert">
    // <button type="button" class="close" data-
    //     dismiss="alert" aria-label="Close"><span aria-
    //     hidden="true">&times;</span></button>
    // Data Berhasil Ditambahkan
    // </div>');
    redirect('c_data_user/index');
}

public function hapus($id)
{
    $where = array('id' => $id);
    $this->m_data_user->hapus_data($where, 'login');
    // $this->session->set_flashdata('message', '
    // <div class="alert alert-danger alert-dismissible"
    //     role="alert">
    // <button type="button" class="close" data-
    //     dismiss="alert" aria-label="Close"><span aria-
    //     hidden="true">&times;</span></button>
    // Data Berhasil Dihapus
    // </div>');
    redirect('c_data_user/index');
}

public function edit($id)
{
    $where = array('id' => $id);
    $data['data_user'] = $this->m_data_user-
        >edit_data($where, 'login')->result();

```

```

$this->load->view('templates/header');
$this->load->view('templates/sidebar');
$this->load->view('data_user/v_edit_user', $data);
$this->load->view('templates/footer');
}

```

```

public function update()
{
    $id      = $this->input->post('id');
    $nama     = $this->input->post('nama');
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $role_id  = $this->input->post('role_id');

    $data = array(
        'nama'      => $nama,
        'username'  => $username,
        'password'  => $password,
        'role_id'   => $role_id
    );
    $where = array('id' => $id);
    $this->m_data_user->update_data($where, $data,
        'login');
    $this->session->set_flashdata('message', '
<div class="alert alert-info alert-dismissible"
    role="alert">
<button type="button" class="close" data-
    dismiss="alert" aria-label="Close"><span aria-
    hidden="true">&times;</span></button>
Data Berhasil Diupdate
</div>');
    redirect('c_data_user/index');
}

```

```

public function detail($id)

```

```

    {
        $detail = $this->m_data_user->detail_data($id);
        $data['detail'] = $detail;

        $this->load->view('templates/header');
        $this->load->view('templates/sidebar');
        $this->load->view('data_user/v_detail_user', $data);
        $this->load->view('templates/footer');
    }
}

```

E. C_pemetaan.php

```

<?php
defined ('BASEPATH') OR exit('No direct script access
    allowed');

/**
 *
 */
class C_pemetaan extends CI_Controller{

    public function __construct()
    {
        parent::__construct();
        // $this->simple_login->cek_login();
        $this->load->library(array('googlemaps'));
        $this->load->model('m_gis');
        $this->load->library("Session");
    }

    public function index()
    {
        //Menampilkan Lokasi
        $this->load->library('googlemaps');
        $config['center'] = '-6.898086, 107.619366';
        $config['zoom'] = '15';
        $this->googlemaps->initialize($config);
    }
}

```

```

$pelanggan=$this->m_gis->datapelanggan();
foreach ($pelanggan as $key => $value) {
    $marker = array();
    $marker['animation'] = 'DROP';
    $marker['position'] = "$value-
>LATITUDE, $value->LONGITUDE";
    $marker['infowindow_content'] = '<div
class="media" style="width:300px;">';
    $marker['infowindow_content'] .= '<div
class="media-left">';
    $marker['infowindow_content'] .=
'</div>';
    $marker['infowindow_content'] .= '<div
class="media-body">';
    $marker['infowindow_content'] .= '<h5
class="media-heading">'.$value->ADD_ON.'</h5>';
    // $marker['infowindow_content'] .=
'<a>'.$value->JALAN.'</a>';
    // $marker['infowindow_content'] .=
'<a>'.$value->NOJALAN.'</a><br>';
    // $marker['infowindow_content'] .=
'<a>'.$value->DISTRİK.'</a><br>';
    $marker['infowindow_content'] .=
'<a>'.$value->NAMA.'</a><br>';
    $marker['infowindow_content'] .=
'<a>'.$value->KOTA.'</a><br>';
    $marker['infowindow_content'] .=
'</div>';
    $marker['infowindow_content'] .=
'</div>';
    if ($value->ADD_ON ==
'TINDIHOME') {
        $marker['icon'] =
base_url('assets/icon/indihome3.png');
    }
}

```

```

        }elseif ($value->ADD_ON ==
'MOVIN') {
            $marker['icon'] =
base_url('assets/icon/movin1.png');
        }elseif ($value->ADD_ON ==
'INDIHOME GAMER'){
$marker['icon'] =
base_url('assets/icon/indihome_gamer1.png');
        }else{
            $marker['icon'] =
base_url('assets/icon/ott123.png');
        }

        $this->googlemaps-
>add_marker($marker);
    }
    $this->googlemaps->initialize($config);
    $data['map'] = $this->googlemaps-
>create_map();

    $this->load->view('templates/header');
    if ($this->session->userdata('role_id') === '1') {
        $this->load->view('templates/sidebar');
    }elseif ($this->session->userdata('role_id')
=== '2'){
        $this->load-
>view('templates/sidebar_user');
    }else{
        $this->session-
>set_flashdata('pesan','<div class="alert alert-danger"
role="alert">
    <strong>Anda belum Login !!!</strong>
    <button type="button" class="close" data-
dismiss="alert" aria-label="Close">
    <span aria-hidden="true">&times;</span>

```

```

        </button>
    </div>');

        redirect('auth/login');
    }
    $this->load->view('pemetaan/v_pemetaan',
    $data, FALSE);
    $this->load->view('templates/footer');

    }

}

```

F. C_indihome.php

```

<?php
class C_indihome extends CI_Controller{

    public function __construct(){
        parent::__construct();
        $this->load-
    >library(array('googlemaps'));
        $this->load->model('m_indihome');
        $this->load->library('Excel');

    }
    public function index(){

        $data['indihome1'] = $this->m_indihome-
    >indihome1();
        $data['witel'] = $this->m_indihome->witel();
        $data['datel'] = $this->m_indihome->datel();

        //Menampilkan Lokasi
        $this->load->library('googlemaps');
        $config['center'] = '-6.898086, 107.619366';
        $config['zoom'] = '15';
        $this->googlemaps->initialize($config);
    }
}

```



```

        $marker['position'] = '-6.898086, 107.619366';
        $marker['draggable'] = true;
        $marker['ondragend'] =
'setMapToForm(event.latLng.lat(),event.latLng.lng());'
;
        $this->googlemaps->add_marker($marker);

        //validasi input
        $valid=$this->form_validation;
        // $valid->set_rules('NO_INET','No
Internet','required|is_unique[tbl_pemetaan.NO_INET]'
);
        $valid->set_rules('NO_INET','No
Internet','required');
        $valid->set_rules('NAMA','Nama');
        $valid->set_rules('JALAN','Jalan');
        $valid->set_rules('NOJALAN','No Jalan');
        $valid->set_rules('DISTRİK','Distrik');
        $valid->set_rules('KOTA','Kota');
        $valid-
>set_rules('latitude','Latitude','required');
        $valid-
>set_rules('longitude','Longitude','required');
        $valid->set_rules('ADD_ON','Add
On','required');

        if($valid->run()===FALSE)
        {

                $data['map'] = $this->googlemaps-
>create_map();

        $this->load->view('templates/header');
        if ($this->session->userdata('role_id') ==='1') {
                $this->load->view('templates/sidebar');

```

```

        }elseif($this->session->userdata('role_id')
===2'){
            $this->load-
>view('templates/sidebar_user');
        }else{
            $this->session-
>set_flashdata('pesan','<div class="alert alert-danger"
role="alert">
<strong>Anda belum Login !!!</strong>
    <button type="button" class="close" data-
dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&times;</span>
    </button>
</div>');
            redirect('auth/login');
        }
        $this->load-
>view('indihome/v_indihome',$data, FALSE);
        $this->load->view('templates/footer');

    }else{
        $i=$this->input;
        $data = array('NO_INET'=> $i-
>post('NO_INET'),
            'NAMA' => $i->post('NAMA'),
            'JALAN' => $i->post('JALAN'),
            'NOJALAN' => $i-
>post('NOJALAN'),
            'DISTRİK' => $i-
>post('DISTRİK'),
            'KOTA' => $i->post('KOTA'),
            'latitude' => $i->post('latitude'),
            'longitude' => $i-
>post('longitude'),

```

```

                                'ADD_ON' => $i-
>post('ADD_ON')
    );
    $this->m_indihome-
>inputpelanggan($data);
    $this->session->set_flashdata('sukses',
'Data Pelanggan Berhasil diTambahan');

    redirect(base_url('c_indihome'),'refresh');
    }

}

public function user(){

    $data['indihome1'] = $this->m_indihome-
>indihome1();
    $data['witel'] = $this->m_indihome->witel();
    $data['datel'] = $this->m_indihome->datel();

    //Menampilkan Lokasi
    $this->load->library('googlemaps');
    $config['center'] = '-6.898086, 107.619366';
    $config['zoom'] = '15';
    $this->googlemaps->initialize($config);
    $marker['position'] = '-6.898086, 107.619366';
$marker['draggable'] = true;
    $marker['ondragend'] =
'setMapToForm(event.latLng.lat(),event.latLng.lng());'
;
    $this->googlemaps->add_marker($marker);

    //validasi input
    $valid=$this->form_validation;

```

```

        // $valid->set_rules('NO_INET','No
Internet','required|is_unique[tbl_pemetaan.NO_INET]'
);
        $valid->set_rules('NO_INET','No
Internet','required');
        $valid->set_rules('NAMA','Nama');
        $valid->set_rules('JALAN','Jalan');
        $valid->set_rules('NOJALAN','No Jalan');
        $valid->set_rules('DISTRİK','Distrik');
        $valid->set_rules('KOTA','Kota');
        $valid-
>set_rules('latitude','Latitude','required');
        $valid-
>set_rules('longitude','Longitude','required');
        $valid->set_rules('ADD_ON','Add
On','required');

        if($valid->run()===FALSE)
        {

                $data['map'] = $this->googlemaps-
>create_map();

                $this->load->view('templates/header');
                if ($this->session->userdata('role_id') ==='1') {
                        $this->load->view('templates/sidebar');
                }elseif($this->session->userdata('role_id')
=== '2'){
                        $this->load-
>view('templates/sidebar_user');
                }else{
                        $this->session-
>set_flashdata('pesan','<div class="alert alert-danger"
role="alert">
<strong>Anda belum Login !!!</strong>

```

```

        <button type="button" class="close" data-
dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&times;</span>
        </button>
    </div>');

        redirect('auth/login');
    }
    $this->load-
>view('indihome/v_indihome_user',$data, FALSE);
    $this->load->view('templates/footer');

    }else{
    $i=$this->input;
        $data = array('NO_INET'=> $i-
>post('NO_INET'),
        'NAMA' => $i->post('NAMA'),
        'JALAN' => $i->post('JALAN'),
        'NOJALAN' => $i-
>post('NOJALAN'),
        'DISTRİK' => $i-
>post('DISTRİK'),
        'KOTA' => $i->post('KOTA'),
        'latitude' => $i->post('latitude'),
        'longitude' => $i-
>post('longitude'),
        'ADD_ON' => $i-
>post('ADD_ON')
        );
        $this->m_indihome-
>inputpelanggan($data);
        $this->session->set_flashdata('sukses',
'Data Pelanggan Berhasil diTambahkan');

        redirect(base_url('c_indihome'),'refresh');
    }

```

```
}
```

```
public function getInet()
{
    $id = $_GET['c'];
    $GetNoInet = $this->m_indihome-
>getInet($id);
    echo json_encode($GetNoInet);
}

public function view_list($id){
    $explode = explode('~', $id);
    $witel = $explode[0];
    $datel = $explode[1];

    $data['indihome'] = $this->m_indihome-
>indihome($witel, $datel);

    $this->load->view('indihome/view_list',
    $data);
}

public function import(){
    if(isset($_FILES["file"]["name"]))
    {
        $path =
        $_FILES["file"]["tmp_name"];
        $object =
        PHPExcel_IOFactory::load($path);
        foreach($object-
>getWorksheetIterator() as $worksheet)
        {
```

```

$highestRow =
$worksheet->getHighestRow();
$highestColumn =
$worksheet->getHighestColumn();
for($row=2;
$row<=$highestRow; $row++)
{
    $kawasan =
$worksheet->getCellByColumnAndRow(0, $row)-
>getValue();
    $witel =
$worksheet->getCellByColumnAndRow(1, $row)-
>getValue();
    $datel =
$worksheet->getCellByColumnAndRow(2, $row)-
>getValue();
    $sto =
$worksheet->getCellByColumnAndRow(3, $row)-
>getValue();
    $ncli =
$worksheet->getCellByColumnAndRow(4, $row)-
>getValue();
    $ndos =
$worksheet->getCellByColumnAndRow(5, $row)-
>getValue();
    $ndem =
$worksheet->getCellByColumnAndRow(6, $row)-
>getValue();
    $no_inet =
$worksheet->getCellByColumnAndRow(7, $row)-
>getValue();
    $nd =
$worksheet->getCellByColumnAndRow(8, $row)-
>getValue();

```

```

$Schanel =
$worksheet->getCellByColumnAndRow(9, $row)-
>getValue();

$Scitem_speedy
= $worksheet->getCellByColumnAndRow(10, $row)-
>getValue();

$Skecepatan =
$worksheet->getCellByColumnAndRow(11, $row)-
>getValue();

$Sdeskripsi =
$worksheet->getCellByColumnAndRow(12, $row)-
>getValue();

$Stgl_reg =
$worksheet->getCellByColumnAndRow(13, $row)-
>getValue();

$Stgl_etat =
$worksheet->getCellByColumnAndRow(14, $row)-
>getValue();

$Sstatus =
$worksheet->getCellByColumnAndRow(15, $row)-
>getValue();

$Snama =
$worksheet->getCellByColumnAndRow(16, $row)-
>getValue();

$Skcontact =
$worksheet->getCellByColumnAndRow(17, $row)-
>getValue();

$Sstatus_order =
$worksheet->getCellByColumnAndRow(18, $row)-
>getValue();

$Salpro =
$worksheet->getCellByColumnAndRow(19, $row)-
>getValue();

$Scat =
$worksheet->getCellByColumnAndRow(20, $row)-
>getValue();

```



```

$jalan =
$worksheet->getCellByColumnAndRow(21, $row)-
>getValue();

$nojalan =
$worksheet->getCellByColumnAndRow(22, $row)-
>getValue();

$distrik =
$worksheet->getCellByColumnAndRow(23, $row)-
>getValue();

$kota =
$worksheet->getCellByColumnAndRow(24, $row)-
>getValue();

$cpack =
$worksheet->getCellByColumnAndRow(25, $row)-
>getValue();

$DateNewTglreg = new
DateTime($tgl_reg);

$DateChangeTglreg =
$DateNewTglreg->format('Y-m-d');

$DateNewTglEtat = new
DateTime($tgl_etat);

$DateChangeTglEtat =
$DateNewTglEtat->format('Y-m-d H:i:s');
$cek_duplicat = $this-
>m_indihome->chek_duplicat($ncli);
if ($cek_duplicat !=
NULL) {
    if
($cek_duplicat->ncli == $ncli) {
        $this-
>m_indihome->update_duplicat($kawasan, $switel,
$datel, $sto, $ncli, $ndos, $ndem, $no_inet, $nd,
$schanel, $citem_speedy, $skecepatan, $deskripsi,
$DateChangeTglreg, $DateChangeTglEtat, $status,

```

```

$nama, $kcontact, $status_order, $alpro, $ccat, $jalan,
$nojalan, $distrik, $kota, $cpack);
        }else{
            $this-
>m_indihome->upload($kawasan, $witel, $datel, $sto,
$nccli, $ndos, $ndem, $no_inet, $nd, $chanel,
$scitem_speedy, $kecepatan, $deskripsi,
$DateChangeTglreg, $DateChangeTglEtat, $status,
$nama, $kcontact, $status_order, $alpro, $ccat, $jalan,
$nojalan, $distrik, $kota, $cpack);
        }
    }else{
        $this-
>m_indihome->upload($kawasan, $witel, $datel, $sto,
$nccli, $ndos, $ndem, $no_inet, $nd, $chanel,
$scitem_speedy, $kecepatan, $deskripsi,
$DateChangeTglreg, $DateChangeTglEtat, $status,
$nama, $kcontact, $status_order, $alpro, $ccat, $jalan,
$nojalan, $distrik, $kota, $cpack);
    }
}

}

}

        redirect('c_dashboard_admin/dashboard');
    }
}
}

```

G. C_movin.php

```

<?php
class C_movin extends CI_Controller{

    public function __construct(){
        parent::__construct();
    }
}

```

```

        $this->load-
>library(array('googlemaps'));
        $this->load->model('m_movin');
        $this->load->library('Excel');
    }
    public function index(){

        $data['movin1'] = $this->m_movin->movin1();
        $data['UserLastActivity'] = $this->m_movin-
>GetLastActivity();

        //Menampilkan Lokasi
        $this->load->library('googlemaps');
        $config['center'] = '-6.898086, 107.619366';
        $config['zoom'] = '15';
        $this->googlemaps->initialize($config);
        $marker['position'] = '-6.898086, 107.619366';
        $marker['draggable'] = true;
        $marker['ondragend'] =
'setMapToForm(event.latLng.lat(),event.latLng.lng());'
;
        $this->googlemaps->add_marker($marker);

        //validasi input
        $valid=$this->form_validation;
        // $valid->set_rules('NO_INET','No
Internet','required|is_unique[tbl_pemetaan.NO_INET]'
);
        $valid->set_rules('NO_INET','No
Internet','required');
        $valid->set_rules('NAMA','Nama');
        $valid->set_rules('JALAN','Jalan');
        $valid->set_rules('NOJALAN','No Jalan');
        $valid->set_rules('DISTRIK','Distrik');
        $valid->set_rules('KOTA','Kota');

```

```

        $valid-
>set_rules('latitude','Latitude','required');
        $valid-
>set_rules('longitude','Longitude','required');
        $valid->set_rules('ADD_ON','Add
On','required');

        if($valid->run()==FALSE)
        {

                $data['map'] = $this->googlemaps-
>create_map();

                $this->load->view('templates/header');
                if ($this->session->userdata('role_id') ==='1') {
                        $this->load->view('templates/sidebar');
                }elseif($this->session->userdata('role_id')
=== '2'){
                        $this->load-
>view('templates/sidebar_user');
                        }else{
                                $this->session-
>set_flashdata('pesan','<div class="alert alert-danger"
role="alert">
        <strong>Anda belum Login !!!</strong>
        <button type="button" class="close" data-
dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&times;</span>
        </button>
        </div>');
                                redirect('auth/login');
                        }
                $this->load->view('movin/v_movin',$data,
FALSE);
                $this->load->view('templates/footer');

```

```

        }else{
            $i=$this->input;
            $data = array('NO_INET'=> $i-
>post('NO_INET'),
                                'NAMA' => $i->post('NAMA'),
                                'JALAN' => $i->post('JALAN'),
                                'NOJALAN' => $i-
>post('NOJALAN'),
                                'DISTRİK' => $i-
>post('DISTRİK'),
                                'KOTA' => $i->post('KOTA'),
                                'latitude' => $i->post('latitude'),
                                'longitude' => $i-
>post('longitude'),
                                'ADD_ON' => $i-
>post('ADD_ON')
            );
            $this->m_movin-
>inputpelanggan($data);
            $this->session->set_flashdata('sukses',
'Data Pelanggan Berhasil diTambahan');
            redirect(base_url('c_movin'),'refresh');
        }

    }

    public function movinUser(){

        $data['movin1'] = $this->m_movin->movin1();
        $data['UserLastActivity'] = $this->m_movin-
>GetLastActivity();

        //Menampilkan Lokasi

```

```

        $this->load->library('googlemaps');
        $config['center'] = '-6.898086, 107.619366';
        $config['zoom'] = '15';
        $this->googlemaps->initialize($config);
        $marker['position'] = '-6.898086, 107.619366';
        $marker['draggable'] = true;
        $marker['ondragend'] =
'setMapToForm(event.latLng.lat(),event.latLng.lng());'
;
        $this->googlemaps->add_marker($marker);

        //validasi input
        $valid=$this->form_validation;
        // $valid->set_rules('NO_INET','No
Internet','required|is_unique[tbl_pemetaan.NO_INET]'
);
        $valid->set_rules('NO_INET','No
Internet','required');
        $valid->set_rules('NAMA','Nama');
        $valid->set_rules('JALAN','Jalan');
        $valid->set_rules('NOJALAN','No Jalan');
        $valid->set_rules('DISTRİK','Distrik');
        $valid->set_rules('KOTA','Kota');
        $valid-
>set_rules('latitude','Latitude','required');
        $valid-
>set_rules('longitude','Longitude','required');
        $valid->set_rules('ADD_ON','Add
On','required');

        if($valid->run()===FALSE)
        {

                $data['map'] = $this->googlemaps-
>create_map();

```

```

        $this->load->view('templates/header');
        if ($this->session->userdata('role_id') === '1') {
            $this->load->view('templates/sidebar');
        }elseif($this->session->userdata('role_id')
=== '2'){
            $this->load-
>view('templates/sidebar_user');
        }else{
            $this->session-
>set_flashdata('pesan','<div class="alert alert-danger"
role="alert">
        <strong>Anda belum Login !!!</strong>
        <button type="button" class="close" data-
dismiss="alert" aria-label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>');
            redirect('auth/login');
        }
        $this->load-
>view('movin/v_movin_user',$data, FALSE);
        $this->load->view('templates/footer');

    }else{
        $i=$this->input;
        $data = array('NO_INET'=> $i-
>post('NO_INET'),
            'NAMA' => $i->post('NAMA'),
            'JALAN' => $i->post('JALAN'),
            'NOJALAN' => $i-
>post('NOJALAN'),
            'DISTRİK' => $i-
>post('DISTRİK'),
            'KOTA' => $i->post('KOTA'),

```

```

        'latitude' => $i->post('latitude'),
        'longitude' => $i-
>post('longitude'),
        'ADD_ON' => $i-
>post('ADD_ON')
    );
    $this->m_movin-
>inputpelanggan($data);
    $this->session->set_flashdata('sukses',
'Data Pelanggan Berhasil diTambahan');
    redirect(base_url('c_movin'),'refresh');
    }

}

public function getInet()
{
    $id = $_GET['c'];
    $GetNoInet = $this->m_movin->getInet($id);
    echo json_encode($GetNoInet);
}

public function view_movin($id){
    $explode = explode('~', $id);
    $periode_awal = $explode[0];
    $periode_akhir = $explode[1];
    $data['periode_awal'] = $periode_awal;
    $data['periode_akhir'] = $periode_akhir;
    $data['movin'] = $this->m_movin-
>movin($periode_awal, $periode_akhir);

    $this->load->view('movin/view_list', $data);
}

public function import(){
    if(isset($_FILES["file"]["name"]))
    {

```



```

        $path =
$_FILES["file"]["tmp_name"];
        $object =
PHPExcel_IOFactory::load($path);
        foreach($object-
>getWorksheetIterator() as $worksheet)
        {
                $highestRow =
$worksheet->getHighestRow();
                $highestColumn =
$worksheet->getHighestColumn();
                for($row=2;
$row<=$highestRow; $row++)
                {
                        $witel      =
$worksheet->getCellByColumnAndRow(0, $row)-
>getValue();
                        $ncli      =
$worksheet->getCellByColumnAndRow(1, $row)-
>getValue();
                        $ndos      =
$worksheet->getCellByColumnAndRow(2, $row)-
>getValue();
                        $ndem      =
$worksheet->getCellByColumnAndRow(3, $row)-
>getValue();
                        $no_inet    =
$worksheet->getCellByColumnAndRow(4, $row)-
>getValue();
                        $item       =
$worksheet->getCellByColumnAndRow(5, $row)-
>getValue();
                        $price      =
$worksheet->getCellByColumnAndRow(6, $row)-
>getValue();

```

```

$worksheet->getCellByColumnAndRow(7, $row)-
>getValue();
$worksheet->getCellByColumnAndRow(8, $row)-
>getValue();
$worksheet->getCellByColumnAndRow(9, $row)-
>getValue();

//
$DateNewTglva = new DateTime($tgl_va);
//
$DateChangeTglva = $DateNewTglva->format('Y-m-
d');

//
$DateNewTglps = new DateTime($tgl_ps);
//
$DateChangeTglps = $DateNewTglps->format('Y-m-
d');

$DateNewTglva
= $tgl_va;

$DateChangeTglva = date('Y-m-d',
strtotime($DateNewTglva));

$DateNewTglps
= $tgl_ps;

$DateChangeTglps = date('Y-m-d',
strtotime($DateNewTglps));

//menghilangkan
tanda kutip satu dan kutip dua

```

```
$testKcontact =  
$skcontact;  
  
$newkcontact =  
str_replace("'", " ", $testKcontact);  
  
$cek_duplicat = $this->m_movin->cek_duplicat($no_inet);  
if ($cek_duplicat !=  
NULL) {  
    if  
($cek_duplicat->no_inet == $no_inet) {  
        $this->m_movin->update_duplicat($witel, $ncli, $ndos,  
$ndem, $no_inet, $item, $price, $DateChangeTglva,  
$DateChangeTglps, $newkcontact);  
    }else{  
        $this->m_movin->upload($witel, $ncli, $ndos, $ndem,  
$no_inet, $item, $price, $DateChangeTglva,  
$DateChangeTglps, $newkcontact);  
    }  
}else{  
    $this->m_movin->upload($witel, $ncli, $ndos, $ndem, $no_inet, $item,  
$price, $DateChangeTglva, $DateChangeTglps,  
$newkcontact);  
}  
  
}  
  
}  
  
}$dataactivity = array(  
    'NAMA' => $this->session->userdata('username'),
```

```

                                'DATE_TIME' =>
date('Y-m-d H:i:s'),
                                'ACTION' => 'UPLOAD
MOVIN'
                                );
                                $this->m_movin-
>insertactivity($dataactivity);
                                // $this->m_movin-
>upload($data);

                                redirect('c_dashboard_admin/dashboard');

                                }
                                }

public function downloadexcel($id) {
    $explode = explode('~', $id);
    $periode_awal = $explode[0];
    $periode_akhir = $explode[1];
    $data['excelmovin'] = $this->m_movin-
>downloadmovin($periode_awal, $periode_akhir);

    $this->load->view('movin/v_downloadexcel',
    $data);
    }
}

```

