

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

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

Aip Suprpto Munari

Student

Penulis

ISBN

Editor

Penyunting Dan

lain-lain

Quotes

CONTRIBUTORS

CONTENTS IN BRIEF

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

LISTINGS

KATA PENGANTAR

Buku ini merupakan panduan lengkap mengenai sistem *dashboard* monitoring kinerja mahasiswa *internship* berbasis gps (*global positioning system*)

ACKNOWLEDGMENTS

ACRONYMS

SYMBOL

INTRODUCTION

BAB I

1.1 Pengenalan PHP

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

Dengan alasan untuk meningkatkan performa, Rasmus Lerdorf membuat ulang kode program tersebut dalam bahasa C. Lerdorf menyebut kode program ini sebagai *Personal Home Page*. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan skrip PERL yang dibuatnya untuk membuat halaman *web*-nya menjadi dinamis.

Awalnya, PHP bekerja di lingkungan Linux, karena merupakan ‘turunan’ bahasa PEARL yang merupakan bahasa pemrograman native di lingkungan Linux dan Unix. Namun seiring perkembangan dan kebutuhan maka PHP dibuat secara *cross platform*. PHP dapat bekerja di lingkungan Windows maupun Linux. PHP mengeksekusi *server script* lebih cepat dari *script* manapun. PHP menempati ranking pertama dalam hal kecepatan eksekusi, diikuti oleh ASP dan JSP.

PHP difokuskan pada scripting *server-side*, jadi anda dapat melakukan apa yang bisa dilakukan CGI dengan menggunakan PHP seperti mengambil data inputan form, mengenerate konten halaman dinamis, mengirim dan menerima *cookies* dan masih banyak lagi. Kemampuan dan *support*-nya untuk *database* juga sangat dapat diandalkan.

1.2 Pengertian PHP

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

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

1.3 Fungsi PHP

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

merupakan sintaksis dasar pada PHP.

a. Pembatas

PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP apapun diluar pembatas tidak diproses oleh PHP. Pembatas paling umum adalah “<?php” untuk membuka dan “?>” untuk menutup kode PHP.

b. Variabel

Variabel dalam PHP diawali dengan simbol *dollar* “\$”, kemudian diikuti dengan huruf sebagai karakter pertama. Setelah itu, dapat dilanjutkan dengan kombinasi huruf dan angka. Variabel tidak boleh mengandung spasi maupun tanda baca di dalamnya, kecuali *underscore* “_”. Variabel pada PHP bersifat “case sensitif”, yang berarti anda harus memperhatikan penulisan huruf besar dan urutan kecil. Variabel \$jumlah berbeda dengan \$Jumlah.

Contoh penamaan variabel yang benar :

\$nama_pemakai

\$kota_3

\$user1

Contoh penamaan variabel yang salah :

\$nama pemakai

\$3kota

\$us\er1

c. Komentar

Ada kalanya kita sebagai pemrogram, karena banyaknya kode program atau variabel dalam program, perlu menandai atau memberi komentar pada program. Komentar pada program merupakan tulisan pada program yang tidak dieksekusi. Pada PHP, ada 3 macam cara penulisannya :

- /* kometar */

Tulisan apapun berada di antara ‘/*’ akan dianggap sebagai komentar. Cara seperti ini sangat berguna dan efisien untuk pemberian komentar yang memakan banyak baris.

- // komentar

Tulisan di baris yang sama setelah ‘//’ akan dianggap sebagai komentar. Cara ini berguna untuk pemberian komentar singkat yang tak lebih dari 1 baris saja.

- # komentar

Sama seperti ‘//’, tulisan di baris yang sama setelah ‘#’ akan dianggap sebagai komentar. Cara ini berguna untuk pemberian komentar singkat yang tak lebih dari 1 baris saja.

1.4 Keunggulan & Kekurangan

Keunggulan bahasa pemrograman PHP, antara lain :

- a. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. Web server yang mendukung PHP dapat ditemukan di mana-mana dari mulai *apache*, *IIS*, *Lightpad*, hingga *xitami* dengan konfigurasi relatif mudah.
- c. Dalam sisi pengembangan lebih mudah, karena banyak *developer* yang siap membantu dalam pengembangan.
- d. PHP adalah bahasa *open source* yang digunakan berbagai mesin seperti *Linux*, *Unix*, *Macintosh*, *Windows* dan dapat dijalankan secara runtime melalui *console* serta dapat menjalankan perintah-perintah sistem.
- e. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.

Kekurangan bahasa pemrograman PHP, antara lain :

- a. Tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya.
- b. PHP mempunyai kelemahan security tertentu yang mana jika programmer kurang teliti dalam melakukan pemrograman dan mengabaikan isu dan konfigurasi PHP.
- c. Tidak ideal jika untuk pengembangan skala besar.
- d. Tidak dapat memisahkan antara tampilan dengan logik dengan baik.

1.5 Pengenalan OOP

Object-Oriented Programming (OOP) adalah sebuah pendekatan untuk pengembangan *development* suatu *software* dimana dalam struktur *software* tersebut didasarkan kepada interaksi *object* dalam penyelesaian suatu proses/tugas atau sebuah metodologi atau cara berpikir dalam melakukan pemrograman dimana pendefinisian tipe data disertai dengan pendefinisian fungsi. Struktur data yang seperti ini disebut dengan istilah *object*.

Paradigma pemrograman OOP dapat dilihat sebagai interaksi sebuah *object* dalam melakukan tugasnya. *Object-oriented* programs terdiri dari *objects* yang berinteraksi satu sama lainnya untuk menyelesaikan sebuah tugas. Konsep OOP dimulai pertengahan 1960-an dengan sebuah bahasa program SIMULA 1 (1962-65) dan simula 67 (1967). Kemudian dilanjutkan di era 1970-an dengan *SMALLTALK* menjadi yang pertama kali disebut *object-oriented*.

Meskipun *developer software* tidak secara intensif mengembangkan OOP. Bahasa pemrograman C adalah salah satu

bahasa pemrograman komputer. Dibuat pada tahun 1970-an untuk sistem operasi Unix oleh Bell Labs (Ken Thompson dan Dennis M. Ritchie). Merupakan kelanjutan dari bahasa BCPL. Bahasa pemrograman C merupakan salah satu bahasa pemrograman yang paling sering dipakai oleh pemrogram diseluruh dunia, terutama karena C memperbolehkan pengakses memori secara manual. C telah memperngaruhi bahasa-bahasa pemrograman yang lain, terutama C++. Bahkan C seringkali dipakai untuk membuat bahasa-bahasa pemrograman lain. Umumnya C dipakai untuk membuat program sistem dan jaringan, walaupun tidak jarang dipakai untuk membuat program aplikasi.

1.6 Pengertian OOP

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

Paradigma OOP dapat dilihat sebagai interaksi dari objek yang saling berhubungan satu sama lain untuk melakukan tugasnya. Masing-masing objek dapat memiliki tugasnya. Pada paradigma ini objek tersebut dapat bekerja sendiri dan juga saling berhubungan seperti menerima, mengirim data kepada objek lainnya dan memproses data.

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

1.7 Konsep Object Oriented Programming

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

pesan yang sama sesuai dengan sifat masing-masing objek.

1.8 Jenis-jenis OOP pada bahasa pemrograman

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

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

1.9 Keunggulan & Kekurangan OOP

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

1.10 Pengertian Internship

Internship adalah bentuk pendidikan pengalaman yang mengintegrasikan pengetahuan dan teori yang dipelajari di kelas, dengan aplikasi praktis dan pengembangan keterampilan dalam lingkungan kerja yang profesional. Siswa dapat memperoleh kredit

akademik, sebagaimana ditentukan oleh lembaga pendidikan, atau ada koneksi lain ke lembaga pendidikan pemberi gelar.

Program *internship* merupakan program yang dikhususkan bagi mahasiswa yang telah memiliki pengetahuan (*knowledge*) Program Studi D4 Teknik Informatika minimal 5 Semester. Kebutuhan untuk pemenuhan kemampuan (*skill*) yang dimiliki menjadi dasar dalam program *internship* ini. Upaya mendekatkan kurikulum yang berbasiskan kompetensi dan kebutuhan industri mendorong program ini dilakukan dengan pola yang disesuaikan dengan kondisi pegawai/pekerja. Hal ini dimaksudkan untuk interpretasi *knowledge* yang dimiliki serta meningkatkan (*enrichment*) pengetahuan dalam bidang Teknik Informatika melalui praktek langsung. Harapan besar agar pengetahuan menjadi kompetensi yang unggul melalui pengenalan dan penguasaan terhadap kemampuan (*skill*) dalam *internship* ini.

1.11 Pengertian Monitoring

Kegiatan monitoring lebih (terfokus) pada kegiatan yang sedang dilaksanakan. Monitoring dilakukan dengan cara menggali untuk mendapatkan informasi secara regular berdasarkan indikator tertentu, dengan maksud mengetahui apakah kegiatan yang sedang berlangsung sesuai dengan perencanaan dan prosedur yang telah disepakati. Indikator monitoring mencakup esensi aktivitas dan target yang ditetapkan pada perencanaan program. Apabila monitoring dilakukan dengan baik akan bermanfaat dalam memastikan pelaksanaan kegiatan tetap pada jalurnya (sesuai pedoman dan perencanaan program). Juga memberikan informasi kepada pengelola program apabila terjadi hambatan dan penyimpangan, serta sebagai masukan dalam melakukan evaluasi. Secara prinsip, monitoring dilakukan sementara kegiatan sedang berlangsung guna memastikan kesesuaian proses dan capaian sesuai rencana atau tidak. Bila ditemukan penyimpangan atau kelambanan maka segera dibenahi sehingga kegiatan dapat berjalan sesuai rencana dan targetnya. Jadi, hasil monitoring menjadi input bagi kepentingan proses selanjutnya. Sementara Evaluasi dilakukan pada akhir kegiatan, untuk mengetahui hasil atau capaian akhir dari kegiatan atau program. Hasil Evaluasi bermanfaat bagi rencana pelaksanaan program yang sama di waktu dan tempat lainnya.

1.12 Pengertian GPS

Global positioning system (GPS) adalah sistem koordinat global yang dapat menentukan koordinat posisi benda dimana saja di bumi baik koordinat lintang, bujur, maupun ketinggiannya.

1.13 Pengertian Google Maps API

Google Maps API merupakan pengembangan teknologi dari google yang digunakan untuk menanamkan Google Map di suatu aplikasi yang tidak dibuat oleh Google. Google Maps API merupakan

suatu library dengan bentuk javascript yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai dengan kebutuhan. Dalam perkembangannya Google Maps API diberikan kemampuan untuk mengambil gambar peta statis. Melakukan geocoding, dan memberikan penuntun arah. Google Maps API bersifat gratis untuk publik. Penggunaan Google Maps API pada pengembangan aplikasi android dengan menggunakan Eclipse dan komputer menggunakan sistem operasi windows. Google Maps API terbagi menjadi 4 platform yaitu web, Webservice, Android, dan iOS.

1.13.1 Platform Google Maps API

1. Google Maps API for Android

- Google Maps Android API

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

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

- Google Place Api for Android

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

Digunakan untuk aplikasi yang membutuhkan fitur mendeteksi lokasi disekitar user di Android dengan bahas JAVA. Misalkan daftar restoran terdekat atau pom bensin terdekat.

2. Google Maps API for iOS

Api disini digunakan untuk membuat aplikasi iPad atau iPhone.

- Google Maps SDK for iOS

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

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

- Google Place Api for iOS

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

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

3. Google Maps API for Web

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

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

API untuk web ini terbagi menjadi :

- Google Map Javascript API, digunakan untuk menampilkan

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

- Google Street View Image API biasa digunakan untuk menampilkan lokasi jalan dari StreetView. API ini sifatnya interaktif, sangat cocok untuk menampilkan lokasi dengan sudut pandang 360 derajat.

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

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

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

Berikut ini adalah komponennya :

- Google Maps GeoCoding API ini berfungsi untuk mengkonversi koordinat menjadi alamat jalan atau lokasi menjadi koordinat (reverse GeoCoding).

- Google Place Webservice API ini memiliki fungsi yang sama dengan Google Place API, yang membedakan yaitu webservice ini lebih luas pemakaiannya dan tidak dibatasi oleh bahasa pemrograman java, Objective-C atau javascript.

- Google Maps Elevation API ini memiliki fungsi menampilkan ketinggian suatu lokasi dari atas permukaan laut. Misalkan kota Yogyakarta ketinggian dari permukaan laut 10 Meter, sedangkan jika lokasi koordinatnya adalah puncak gunung merapi, ketinggiannya 3000 Meter.

- Google Maps Road API ini memiliki fungsi sangat spesifik, yaitu memastikan Programmer GPS tracker menampilkan log perjalanan persis di jalan yang dilalui. Biasanya dipakai perusahaan GPS tracking atau asset Tracking.

- Google Map GeoLocation API ini memiliki fungsi yang sangat canggih, dikarenakan bisa mendeteksi lokasi user walaupun GPS di smartphone dimatikan. GeoLocation API memanfaatkan sinyal wifi untuk mendeteksi/melacak

keberadaan lokasi user. Artinya perangkat yang tidak memiliki GPS masih bisa dideteksi lokasinya walaupun tidak seakurat sensor GPS.

- Google Maps Direction API. Dipakai untuk menunjukkan jalur perjalanan dari lokasi A ke lokasi B. Biasanya lokasi A adalah lokasi User saat ini dan lokasi adalah Point of interest yang dituju.

- Google Maps Timezone API digunakan untuk mendeteksi Timezone suatu koordinat/lokasi. Misalkan input Jakarta, maka timezone GMT+7.

- Google Maps Distance Matrix API ini sangat cocok untuk mengkalkulasi jarak dan waktu tempuh ke suatu lokasi. API ini masih berhubungan dengan Direction API.

1.13.2 Kelebihan dan kekurangan Google Maps API

a. Kelebihan Google Maps API

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

b. Kekurangan Google Maps API

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

1.14 Pengenalan 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.

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

1.14.2 Penjelasan singkatan XAMPP

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

1.14.3 Fitur-fitur XAMPP

Berikut ini merupakan beberapa fitur-fitur dari xampp :

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

1.14.4 Bagian-bagian XAMPP

Berikut ini adalah merupakan bagian-bagian dari XAMPP:

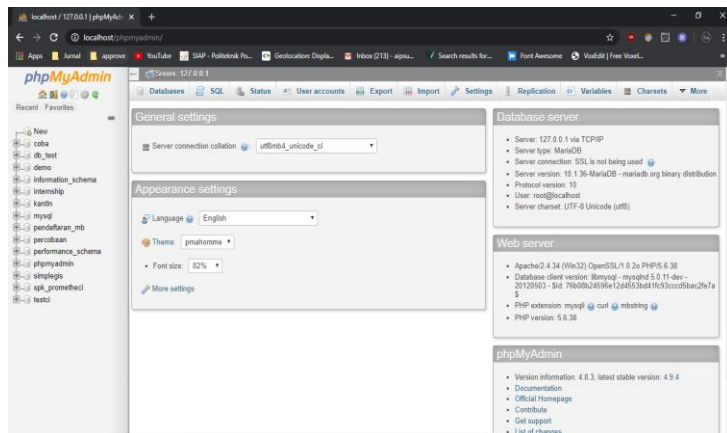
1. Htdocs

Name	Date modified	Type	Size
anonymous	19/10/2019 14.55	File folder	
apache	19/10/2019 14.56	File folder	
cgi-bin	19/10/2019 15.09	File folder	
contrib	19/10/2019 14.56	File folder	
FileZillaFTP	19/10/2019 15.09	File folder	
htdocs	07/01/2020 07.21	File folder	

Gambar 1.1 Htdocs

Htdocs adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain.

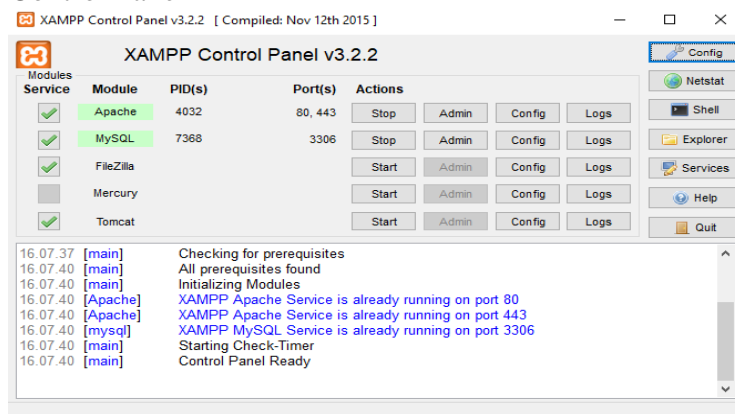
2. PhpMyadmin



Gambar 1.2 PhpMyadmin

PhpMyadmin merupakan bagian untuk mengelola basis data MySQL yang ada dikomputer. Untuk membukanya, buka browser internet lalu ketikkan alamat <http://localhost/phpMyadmin/>, maka akan muncul halaman phpMyAdmin.

3. Control Panel



Gambar 1.3 Control Panel

Control panel yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

1.14.5 Kelebihan dan kekurangan XAMPP

Berikut ini terdapat beberapa kelebihan dan kekurangan XAMPP, terdiri atas :

a. Kelebihan :

1. *Database Storage Engine* ini banyak digunakan oleh programmer apalagi oleh *web developer* karena sifatnya yang *free*.
2. Keamanan datanya cukup aman walaupun tidak sehebat PostgreSQL apalagi Oracle.
3. *Engine* ini *multiplatform* sehingga mampu diaplikasikan diberbagai sistem operasi.
4. Kelebihan paling utama *engine* ini adalah kecepatan.

5. Kemampuannya sudah bisa diandalkan, mempunyai kapasitas yang cukup mumpuni sekitar 60.000 tabel dengan jumlah *record* mencapai 5.000.000.000 bahkan untuk yang terbaru sudah lebih.
- b. Kekurangan :
 1. Tidak cocok untuk menangani data dengan jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data.
 2. Memiliki keterbatasan kemampuan kinerja pada server ketika data yang disimpan telah melebihi batas maksimal kemampuan daya tampung server karena tidak menerapkan konsep *Technology Cluste*.

1.15 Pengenalan Codeigniter

Codeigniter merupakan sebuah *web framework* yang dikembangkan oleh Rick Ellis dari Ellis Lab. Codeigniter dirancang untuk menjadi sebuah web framework yang ringan dan mudah untuk digunakan. Bahkan pengakuan dari Rasmus Lerdorf, sang pencipta bahasa pemrograman PHP mengatakan bahwa Codeigniter merupakan web framework yang mudah, cepat dan handal.

Codeigniter dirilis pertama kali pada tanggal 28 Februari 2006, namun pada bulan Juli 2013 Ellis Lab mengumumkan bahwa mereka mencari pemilik baru untuk Codeigniter karena pada lingkup internal sendiri tidak memiliki cukup fokus untuk mengembangkan Codeigniter, dan pada akhirnya pada bulan Oktober 2014, kepemilikan Codeigniter berpindah ke tangan British Columbia Institute of Technology, yakni salah satu sekolah tinggi teknologi di Kanada.

Setelah hampir lima bulan lamanya sejak peralihan kepemilikan, BCIT akhirnya merilis Codeigniter 3.0. Dibanding versi sebelumnya tentunya Codeigniter 3 mempunyai fitur yang lebih kaya seperti pengembangan *Database Driver*, terdapat pustaka yang baru dan juga PDO Codeigniter kini telah berfungsi secara penuh dengan subdriver.

Sebelum mencoba Codeigniter, perlu kita ketahui istilah *web framework* itu sendiri. Menurut *Microsoft Computer Dictionary*, *web* adalah sekumpulan dokumen yang saling terhubung dalam sistem *hypertext* yang penggunaannya akan menjelajahi web melalui halaman beranda. Sedangkan *framework* adalah desain struktur dasar yang dapat digunakan kembali (*reusable*) yang terdiri dari *abstract class* dan *concrete class* di pemrograman yang berorientasi objek.

Dirilis pada dokumentasi Codeigniter, Codeigniter merupakan toolkit bagi orang yang ingin membangun aplikasi *web* menggunakan PHP. Tujuannya adalah membuat pengembangan proyek menjadi lebih cepat dibandingkan dengan menulis kode dari awal (*strachth*). Codeigniter menyediakan kumpulan *library* untuk tugas-tugas yang sering dilakukan dan sangat mudah untuk mengakses *library* yang tersedia di Codeigniter. Dengan Codeigniter, kita cukup fokus pada

pengembangan *project* dan meminimalisir kode yang akan ditulis.

1.15.1 Fitur-fitur Codeigniter

Menilai baik dan buruknya sebuah aplikasi dari fiturnya, bukanlah sebuah cara yang tepat. Karena ini tidak menggambarkan bagaimana sebuah aplikasi dirancang. Fitur-fitur tersebut tidak menggambarkan kualitas kode, performance dan keamanan (*security*). Satu-satunya cara untuk mengetahui yaitu dengan melihat kodenya.

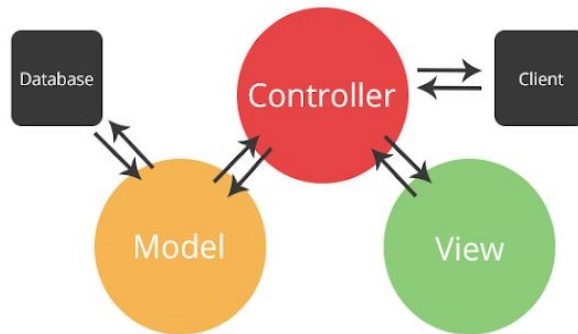
Berikut ini adalah fitur-fitur yang didukung oleh Codeigniter :

- ✓ Model-View-Controller Based
- ✓ PHP 7 Compatible
- ✓ Extremely Light Weight
- ✓ Full Featured database classes with support for several platform
- ✓ Active Record Database Support
- ✓ Forms and Data Validation
- ✓ Security and XSS Filtering
- ✓ Session Management
- ✓ Email Sending Class, Support Attachment, HTML/Text email, multiple protocols (sendmail, SMTP, and Mail)
- ✓ Image Manipulation Library (cropping, resizing, rotation) Support GD, Image Magic, and NetPBM
- ✓ File Uploading Class
- ✓ FTP Class
- ✓ Localization
- ✓ PAGINATION
- ✓ Data Encryption
- ✓ Dan lainnya.

1.15.2 Mengenal Konsep MVC

Codeigniter menggunakan pendekatan Model-View-Controller, yang bertujuan untuk memisahkan logika dan presentasi. Konsep ini mempunyai keunggulan dimana desainer dapat bekerja pada template file, sehingga redundansi kode presentasi dapat diperkecil. Berikut adalah konsep Model-View-Controller yang diterapkan pada Codeigniter.

1. Model, menggambarkan struktur data. Biasanya kelas model akan berisi fungsi yang digunakan untuk mengambil, menambah dan memperbaharui informasi yang ada didatabase.
2. View, informasi yang diperlihatkan kepada user. View adalah halaman web yang terdiri dari HTML, CSS dan Javascript, akan tetapi pada codeigniter, view dapat juga sebagai potongan halaman seperti header atau footer. Bahkan dapat juga halaman RSS atau tipe halaman lainnya.
3. Controller, perantara Model, View, dan resource lainnya yang dibutuhkan untuk menangani HTTP request dan menghasilkan halaman web.



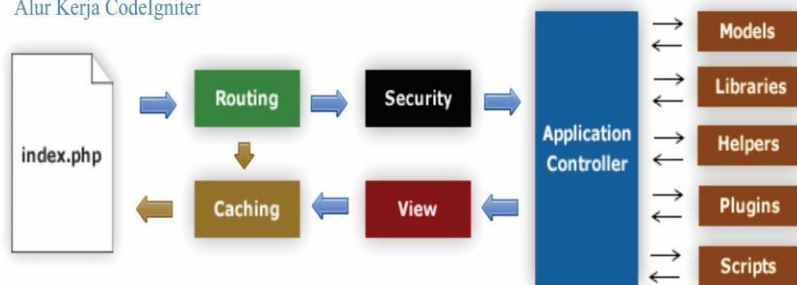
Gambar xx Cara kerja MVC

Pada Codeigniter, programmer juga dapat mengabaikan pemakaian Model dan cukup menggunakan Controller dan View. Cara tersebut sah-sah saja dipakai, tetapi akan lebih baik jika kita memakai Model juga agar hierarki MVC tercapai dan disamping itu beban di Controller akan berkurang.

1.15.3 Sistem Kerja Codeigniter

Untuk melengkapi pemahaman mengenai Codeigniter, berikut adalah gambaran sederhana yang menjelaskan bagaimana Codeigniter bekerja :

Alur Kerja CodeIgniter



Gambar xx. Alur Kerja Codeigniter

- a. Index.php: berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- b. Router : router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program.
- c. Cache File : apabila dalam program sudah terdapat “*cache file*” maka file tersebut akan langsung dikirim ke browser. File cache inilah yang dapat membuat sebuah *website* dapat dibuka dengan lebih cepat. *Cache file* dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.
- d. *Security* : Sebelum *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*: bagia akhir akan dilakukan pengecekan semua program yang ada dalam *view* kemudian file akan mengirimkannya ke *browser* supaya dapat dilihat. Apabila file *view* sudah ada yang di “*cache*” maka file *view* baru yang belum ter-*cache* akan *update* file *view* yang sudah ada.

1.15.3 Struktur Folder Codeigniter

Pada Codeigniter terdapat hirarki yang dikepalai oleh tiga folder utama, yaitu *application*, *system*, dan *user_guide*. Agar aplikasi semakin ringkas maka folder *user_guide* lebih baik dihapus saja, sehingga kita akan memakai 2 folder di dalam folder Codeigniter, yaitu *application* dan *system*. Di dalam folder *application* terdapat banyak sub-folder, folder-folder tersebut berguna untuk menyimpan berbagai jenis file sesuai dengan nama foldernya, misalnya file-file model akan disimpan di dalam folder *models*, file-file *view* akan disimpan di folder *views*, dan seterusnya.

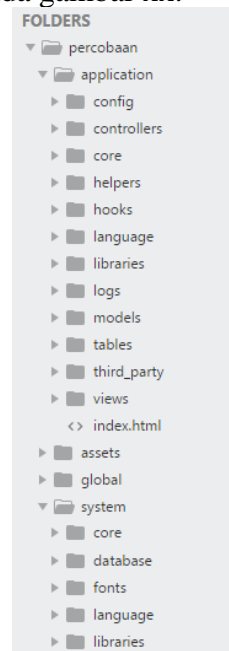
Adapun susunan folder Codeigniter secara default adalah sebagai berikut :

1. Folder *application*, disini aplikasi yang akan kitan bangun diletakkan.
 - ✓ Folder *cache*, tempat menyimpan semua *cache* yang dibuat *caching library*.
 - ✓ Folder *config*, tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari *database*, *router* dan *autoload* aplikasi.
 - ✓ Folder *controllers*, tempat menyimpan semua file *controller*.
 - ✓ Folder *core*, class-class utama yang sangat vital bagi Codeigniter. Jika ada salah satu atau beberapa class yang termasuk *core library* tidak ada, maka Codeigniter tidak dapat berfungsi dengan baik.
 - ✓ Folder *helpers*, tempat menyimpan *helper-helper* yang bukan berasal dari Codeigniter. Gunanya untuk melakukan tugas tertentu yang sederhana, misalnya memformat tanggal, membuat link, membuat input form dan sebagainya.
 - ✓ Folder *hooks*, tempat menyimpan *hook* yang digunakan mengubah alur fungsi dari *core* Codeigniter.
 - ✓ Folder *language*, tempat menyimpan bahasa-bahasa yang akan digunakan.
 - ✓ Folder *libraries*, tempat menyimpan semua *library* buatan kita sendiri.
 - ✓ Folder *logs*, tempat menyimpan semua *logs* generated oleh CI.
 - ✓ Folder *models*, tempat menyimpan semua model.
 - ✓ Folder *third_party*, file ini adalah *library* pihak ketiga. Maksudnya *library* yang berasal dari sumber lain

(bukan berasal dari Codeigniter). Misalnya untuk membuat laporan dengan format PDF, maka kita perlu memerlukan library yang dapat merubah string HTML menjadi PDF.

- ✓ Folder views, tempat menyimpan semua file view aplikasi.
- 2. Folder system, menyimpan semua file system pada core *framework*.
 - ✓ Folder core, tempat menyimpan semua class library bawaan Codeigniter.
 - ✓ Folder database, tempat menyimpan semua driver database drivers dan class yang akan digunakan.
 - ✓ Folder fonts, tempat menyimpan semua font yang digunakan *image manipulation* library.
 - ✓ Folder helpers, tempat menyimpan semua helper core Codeigniter.
 - ✓ Folder language, tempat menyimpan file bahasa.
 - ✓ Folder libraries, tempat menyimpan semua library.

Untuk lebih jelasnya mengenai susunan folder di Codeigniter, anda dapat melihat pada gambar xx.



Gambar xx. Struktur Directory Codeigniter

Bila anda perhatikan pada Gambar xx di dalam folder application ada nama folder yang sama yaitu core, helpers, libraries.

- Folder core, helper dan libraries yang ada di folder system tempat semua class library dan helper asli yang terbawa oleh Codeigniter.
- Folder core, helpers dan libraries yang ada di folder

application adalah tempat untuk semua custom class library dan custom helper yang akan mungkin akan dibuat oleh programmer. Dengan adanya custom library dan custom helper, maka kita dapat membuat helper atau library sendiri. Hal ini terjadi karena terkadang library atau helper bawaan Codeigniter tidak menyediakan fungsionalitas yang dibutuhkan.

Folder system berisi file-file inti Codeigniter, sebaiknya anda tidak merubah-ubah isi di dalam folder ini. Pada prakteknya, jika kita membuat aplikasi memakai Codeigniter maka sebagian besar waktu akan dihabiskan untuk bekerja dengan folder application dan sub folder yang ada di dalamnya.

1.15.4 Helper

Helper adalah file-file program pada Codeigniter yang berisi fungsi -fungsi bantuan yang dapat dipakai untuk mengerjakan tugas tertentu. Isi dari helper adalah fungsi-fungsi biasa, bukan class. File-file native helper (helper asli bawaan Codeigniter) tersimpan di system/helpers.

Codeigniter memiliki beberapa helper, namun di dalam buku ini kita tidak membahas secara detail penggunaan dari masing-masing helper tersebut. Kita hanya akan membahas manfaat helper tersebut dan cara memanggilnya. Untuk mengetahui fungsi-fungsi yang ada pada suatu helper secara detail, anda dapat menemukannya pada user_guide di bagian helper yang dimaksud.

Untuk memanggil suatu helper, anda dapat menggunakan perintah :

```
$this->load->helper('nama_helper');
```

Jika ingin memanggil beberapa helper sekaligus, maka nama helper diberikan dalam suatu array :

```
$helpers = ['helper1', 'helper', 'helper3'];
```

```
$this->load->helper($helpers);
```

Atau dibisa dipanggil secara langsung :

```
$this->load->helper(['helper1', 'helper2', 'helper3']);
```

Berikut ini adalah beberapa helper yang ada di dalam Codeigniter :

- ✓ Array Helper, berisi fungsi-fungsi yang berkaitan dengan array :

```
$this->load->helper('array');
```

- ✓ Captcha Helper, berisi fungsi-fungsi untuk pembuatan captcha. Captcha adalah cara untuk membedakan apakah user adalah seseorang atau bot/robot. Biasanya dalam bentuk karakter gabungan angka ataupun huruf ataupun teka-teki soal untuk diisikan ke dalam suatu input *text*.

Captcha yang dihasilkan helper ini merupakan captcha dasar berupa gambar berisi campuran antara huruf dan angka :

```
$this->load->helper('captcha');
```

- ✓ Cookie Helper, berisi fungsi-fungsi yang berkaitan dengan

pembuatan cookie :

```
$this→load→helper('cookie');
```

- ✓ Date Helper, berisi fungsi-fungsi yang berkaitan dengan waktu dan tanggal :

```
$this→load→helper('date');
```

- ✓ Directory Helper, berisi fungsi-fungsi berkaitan dengan folder :

```
$this→load→helper('directory');
```

- ✓ Download Helper, manfaat utama helper ini adalah untuk membuat force download, yaitu suatu fungsi yang akan memaksa user untuk mendownload suatu file ketika link menuju suatu file di-klik oleh user.

```
$this→load→helper('download');
```

- ✓ Email Helper, berisi fungsi-fungsi yang berkaitan dengan pengiriman email :

```
$this→load→helper('email');
```

- ✓ File Helper, berisi fungsi-fungsi yang berkaitan dengan file, misalnya membaca isi file text, menyimpan file text dan sebagainya :

```
$this→load→helper('file');
```

- ✓ HTML Helper, berisi fungsi-fungsi yang berkaitan dengan pembuatan element-element HTML :

```
$this→load→helper('html');
```

- ✓ Inflector Helper, berisi fungsi-fungsi yang berkaitan dengan penformatan untuk string , misalnya membuat bentuk plural dari string “book” menjadi “books”, atau sebaliknya. Bisa juga untuk mengubah string ke dalam bentuk camelCase :

```
$this→load→helper('inflector');
```

- ✓ Language Helper, berisi fungsi-fungsi yang berkaitan dengan bahasa :

```
$this→load→helper('language');
```

- ✓ Number Helper, berisi fungsi-fungsi yang berkaitan dengan angka, misalnya untuk memformat satuan sata dalam bytes :

```
$this→load→helper('number');
```

- ✓ Path Helper, berisi fungsi yang berkaitan dengan file path server :

```
$this→load→helper('path');
```

- ✓ Security Helper, berisi fungsi yang berkaitan dengan keamanan, misalnya terkait dengan XSS Filtering :

```
$this→load→helper('security');
```

- ✓ Smiley Helper, berisi fungsi-fungsi yang membantu manage emoticons :

```
$this→load→helper('smiley');
```

- ✓ String Helper, berisi fungsi yang berkaitan dengan string, misalnya untuk menghapus double slash pada alamat URL, atau untuk menghasilkan random string dan sebagainya :

`$this→load→helper('string');`

- ✓ Text Helper, berisi fungsi-fungsi yang berkaitan dengan teks. Misalnya untuk membatasi jumlah kata yang akan ditampilkan pada suatu artikel, membatasi jumlah karakter, menjalankan fungsi sensor yang menggantikan daftar kata-kata yang terlarang dengan kata-kata pengganti lainnya, dan sebagainya :

`$this→load→helper('text');`

- ✓ Typography Helper, berisi fungsi-fungsi untuk memformat teks agar tepat secara semantic, misalnya mengubah / ganti baris menjadi tag `
`, mengubah suatu karakter ke dalam entitas HTML dan sebagainya :

`$this→load→helper('typography');`

- ✓ URL Helper, berisi fungsi-fungsi yang berkaitan dengan URL. Misalnya untuk membuat link, mendapatkan nilai base url, mendapatkan nilai URL, melakukan redirect, dan sebagainya :

`$this→load→helper('url');`

- ✓ XML Helper, berisi fungsi untuk bekerja dengan XML :

`$this→load→helper('xml');`

1.15.5 Library

Library Codeigniter membantu anda dalam mengerjakan operasi tertentu dengan lebih cepat dan efisien. Tiap library adalah sebuah class PHP dengan berbagai method yang dapat digunakan begitu class tersebut di load oleh Controller. Beberapa class sangat berguna, dan beberapa diantaranya mungkin akan selalu anda butuhkan di berbagai Controller.

Jika demikian, class yang termasuk dalam kategori ini sebaiknya di set autoload (dijalankan secara otomatis) misalnya library database ataupun library session.

Library-library yang terdapat dalam Codeigniter di antaranya adalah :

- ✓ Benchmarking, library benchmarking selalu aktif. Dapat digunakan untuk menghitung penggunaan memory.
- ✓ Calendaring, library ini digunakan untuk membuat kalender bulan dan tahu tertentu.
- ✓ Config, library config diinisialisasi secara otomatis oleh system. Digunakan untuk menerima informasi mengenali konfigurasi system.
- ✓ Database, library database sangat berguna sekali jika aplikasi terhubung dengan database, dilengkapi dengan berbagai method untuk memanipulasi database.
- ✓ Email, digunakan untuk mengirim email.
- ✓ Encryption, berguna untuk operasi enkripsi.
- ✓ File uploading. Library digunakan untuk melakukan upload file. Menyediakan fitur untuk validasi, agar file sesuai dengan *mime type*, ukuran (size dalam kilobyte) atau dimensi untuk gambar.

- ✓ FTP, library FTP digunakan untuk mentransfer file ke semua server.
- ✓ HTML Table, jika anda membutuhkan cara yang cepat untuk menggenerate table html, maka gunakan library ini. Library html table akan menggenerate table dari array atau hasil dari query database.
- ✓ Image Manipulation, digunakan untuk mereseize, membuat thumbnail, crop atau rotate gambar, serta watermark. Beberapa fungsi membutuhkan support PHP yang lebih baik, misalnya GD/GD2.
- ✓ Input and Security, digunakan untuk pre-process input data dari form dan URL serta untuk handle beberapa fungsi security (misalnya untuk menghadapi serangan XSS attacks).
- ✓ Language, library ini digunakan untuk memload beberapa bahasa yang berbeda.
- ✓ Loader, library ini secara otomatis di-load oleh system. Anda akan sering menggunakan library ini untuk memload view di dalam controller, selain itu juga untuk memload libraries.
- ✓ Output, otomatis di-load oleh system. Library ini mempunyai sebuah fungsi utama yaitu mengirimkan halaman web ke browser yang merequest. Juga digunakan untuk proses caching.
- ✓ Pagination, digunakan untuk membuat pagination dari hasil query database. Dapat diatur beberapa record yang ingin ditampilkan dalam satu halaman. Anda juga dapat mengatur tampilan pagination.
- ✓ Session, digunakan untuk *maintenance state* seorang user. Library ini tidak menggunakan session milik PHP, tetapi menghasilkan data untuk session sendiri.
- ✓ Template Parser, digunakan untuk membuat template yang berisi pseudocode. Jika anda pernah menggunakan Smarty, maka Template Parser Codeigniter tidak selengkap dan sepowerfull Smarty tetapi cukup berguna.
- ✓ Trackback, digunakan untuk mengirim dan menerima data Trackback.
- ✓ Unit Testing, library ini digunakan untuk unit testing.
- ✓ URI Class, library ini di-load secara otomatis oleh system. Digunakan untuk memparser URL, membaginya ke dalam segment-segment yang dapat dilewatkan melalui Controller atau disimpan sebagai variabel.
- ✓ User Agent, digunakan untuk mengidentifikasi browser, mobile device atau robot (misalnya milik search engine) yang berkunjung ke situs anda.
- ✓ Form Validation, sebuah class yang sangat berguna. Digunakan untuk memvalidasi input form.
- ✓ XML-RPC, digunakan untuk menseting XML-RPC client

dan server.

- ✓ Zip Encoding, digunakan untuk membuat arsip Zip untuk text dan data binary.

Untuk lebih jelasnya mengenai library pada Codeigniter, anda dapat melihat user_guide nya

1.15.6 Keuntungan Codeigniter

Sebagai *web framework* populer yang menggunakan bahasa pemrograman PHP, Codeigniter mempunyai beberapa keunggulan seperti yang telah disebutkan di dokumentasinya, keunggulan-keunggulan tersebut meliputi :

1. *Free*, karena berada di bawah lisensi open source, kita dapat melakukan apapun dengan Codeigniter. Lisensi lengkapnya dapat dokumentasi.
2. *Light weight*, sistem inti Codeigniter memerlukan library yang sedikit. Sangat berbeda dengan *framework* lainnya yang membutuhkan banyak sumber daya tambahan. *Library* tambahan akan digunakan ketika *request* secara dinamis, membuat *system* yang dibangun menjadi efisien dan cukup cepat.
3. *Fast*, menurut dokumentasi, performa yang dimiliki Codeigniter terbukti cepat setelah dibandingkan dengan *framework* lainnya.
4. Menggunakan kaidah MVC, dengan menggunakan Model-View-Controller, kita dapat memisahkan bagian logic dan presentasion dari aplikasi yang kita bangun. Hal ini tentu sangat cocok dan bagus untuk proyek yang memfokuskan. Desainer fokus pada template file dan programmer fokus pada pembangunan logic dari aplikasi yang dibangun.
5. Menghasilkan URL yang bersih, URL yang dihasilkan oleh Codeigniter bersih dan ramah terhadap *search engine*. Codeigniter menggunakan pendekatan *segment-based* dibandingkan dengan query string yang biasa digunakan oleh programmer yang tidak menggunakan *web framework*.
6. *Packs a Punch*, Codeigniter hadir dengan berbagai *library* yang akan membantu tugas-tugas di pengembangan web yang sudah umum dan sering dilakukan seperti mengakses database, mengirim email, validasi data dari form, mengelola *session*, memanipulasi gambar, dan masih banyak lagi.
7. *Extensible*, kita dapat menambahkan library atau helper yang kita ciptakan sendiri ke dalam Codeigniter. Selain itu kita dapat juga menambahkan fitur lewat class *extensions* atau system hooks.
8. *Thoroughly Documented*, hampir semua fitur, *library* dan helper yang ada di Codeigniter telah terdokumentasi dengan lengkap dan tersusun dengan baik. Ketika mendapatkan unduhan Codeigniter, dokumentasinya sudah tersedia dan siap

digunakan.

9. Mempunyai komunitas yang ramah, bergabung di komunitas Codeigniter tentunya akan membantu sekali para pengguna Codeigniter yang masih pemula atau yang sudah mahir untuk saling berbagai ilmu pengetahuan. Komunitas ini dapat ditemui di codeigniter.com/forums.

1.15.6 Kekurangan Codeigniter

1. Codeigniter tidak ditujukan untuk pembuatan *web* dengan skala besar.
2. *Library* yang sanga terbatas. Hal ini dikarenakan sangat sulit mencari plugin tambahan yang terverifikasi secara resmi, karena pada situsnya Codeigniter tidak menyediakan plugin-plugin tambahan untuk mendukung pengembangan aplikasi dengan CI.
3. Belum adanya *editor* khusus Codeigniter, sehingga dalam melakukan *create project* dan modul-modulnya harus berpindah-pindah folder.

1.16 Pengertian Framework

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

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

Secara sederhana dapat dijelaskan bahwa *framework* merupakan sekumpulan fungsi (*libraries*), maka seorang *programmer* tidak perlu lagi membuat fungsi-fungsi (biasanya disebut kumpulan *Library*) dari awal, *programmer* tinggal memanggil kumpulan *library* atau fungsi yang sudah ada didalam *framework*, tentunya cara menggunakan fungsi-fungsi itu sudah ditentukan oleh *framework*. Berikut merupakan beberapa contoh fungsi standar *framework* diantaranya fungsi grafik, tabel bergaya zebra, validasi, *upload*, *captcha*, proteksi terhadap XSS (XSS filtering), *template*, kompresi, XML, *paging*, enkripsi, email, SEO, *session*, *security*, kalender, bahasa, manipulasi gambar, dan lain-lain.

1.16.1 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 didalman sistem lebih stabil dan handal karena berbasis pada *framework* yang sudah teruji

stabilitas dan keandalannya.

4. *Coding Style* konsisten, memudahkan dalam membaca kode dalam menemukan bugs.
5. *Security Concern* dan *framework* akan mengantisipasi dengan memasang perisai terhadap adanya berbagai masalah keamanan yang mungkin terjadi.
6. Dokumentasi, *framework* dapat mendisiplinkan kita untuk menulis dokumen apa yang kita tulis.

1.16.2 Kelebihan dan kekurangan *Framework*

a. 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 dari pada harus menulis kode dari awal. Selain itu, tidak perlu menulis berulang-ulang 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.

b. Kekurangan *Framework* :

1. Kurangnya pemahaman bahasa pemrograman

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

2. Memiliki batasan

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

3. Kode Publik

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

1.17 Pengenalan Bootstrap

Bootstrap adalah paket aplikasi siap pakai untuk membuat halaman *front-end* maupun *back-end* dari sebuah *website*. Dapat dikatakan, bootstrap adalah *template* desain *web* dengan fitur plus (*framework* css). Bootstrap diciptakan untuk mempermudah proses desain web bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar mengenai HTML dan CSS, anda pun siap menggunakan bootstrap.

Paket bootstrap berisi sekumpulan file CSS, *font* dan javascript yang siap diintegrasikan ke sebuah dokumen HTML menggunakan kaidah-kaidah tertentu. Dokumen HTML yang dihasilkan pun secara dinamis akan tampil dalam *layout* yang disesuaikan dengan ukuran layar piranti pengunjung. Ada dua alternative menggunakan bootstrap. Cara pertama adalah dengan men-download lalu memasang file bootstrap ke *project web* anda. Cara kedua adalah dengan merujuk file-file bootstrap secara langsung dari CDN ke *project web* tanpa mendownload file apapun.

Karena dalam buku ini menggunakan file bootstrap yang di download maka *project* dapat dikerjakan secara *offline* (tanpa terhubung ke internet). Untuk mendownload bootstrap versi terbaru, anda dapat mengunjungi situs resmi bootstrap di <http://getbootstrap.com>.

BAB II

INSTALASI APLIKASI YANG AKAN DIGUNAKAN

1. Visual Studio Code

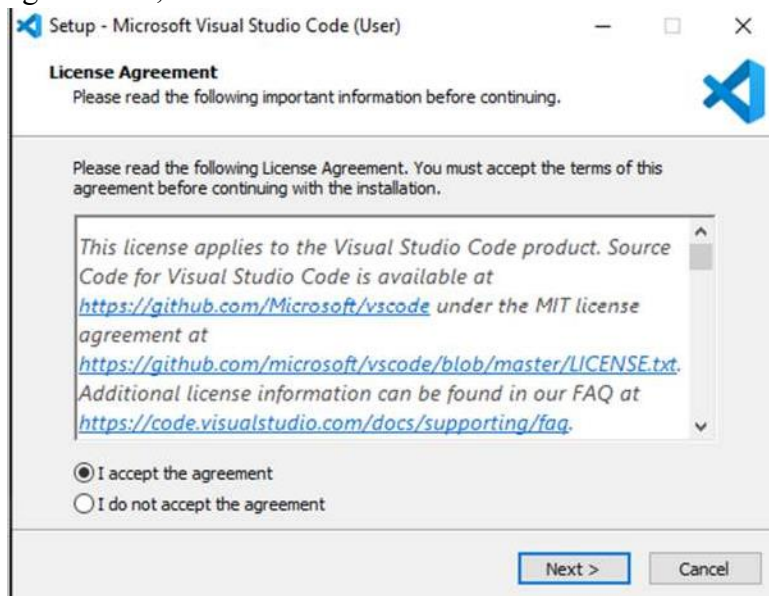
Visual studio code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node.js, serta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

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

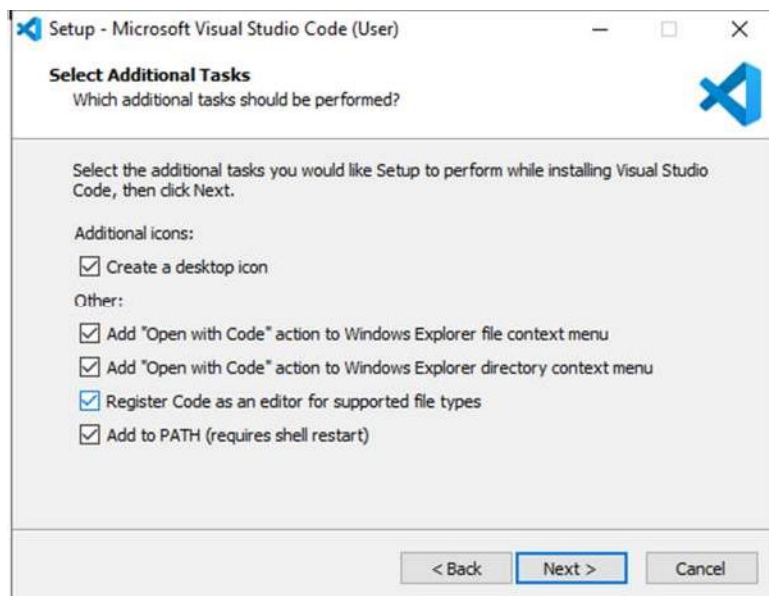
- a. Pertama, *download* terlebih dahulu file installer visual studio code melalui situs resminya pada link berikut ini <https://code.visualstudio.com/download>



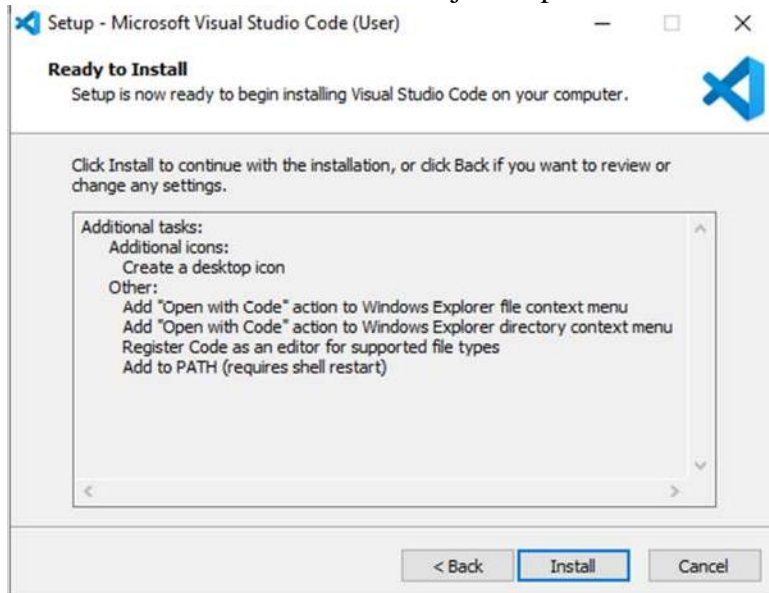
- b. Terdapat 2 pilihan untuk sistem operasi windows, diantaranya ada yang 64 bit dan 32 bit. Pilihlah installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Jika, sudah berhasil di *download*, lanjut pada proses instalasi, pertama klik pada file installer vs code.
- d. Pilih “I accept the agreement”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



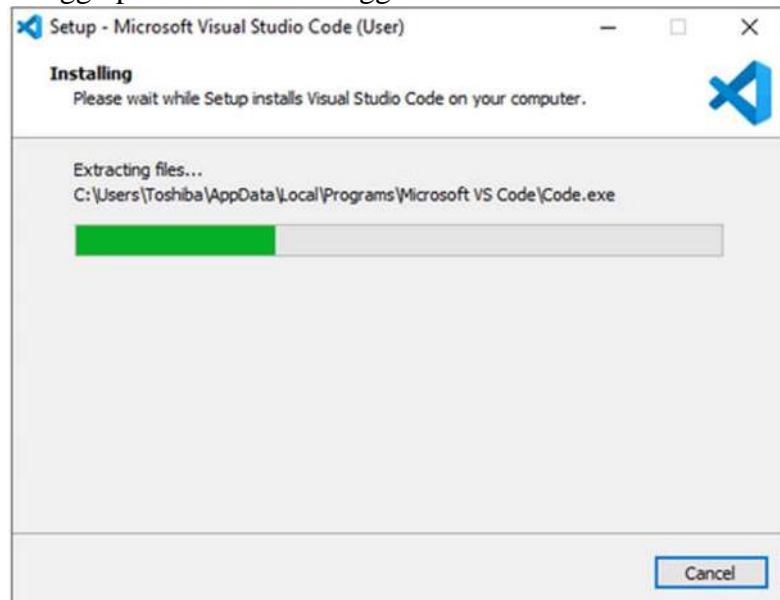
- e. Pada bagian select additional tasks, centang semuanya kemudian klik next.



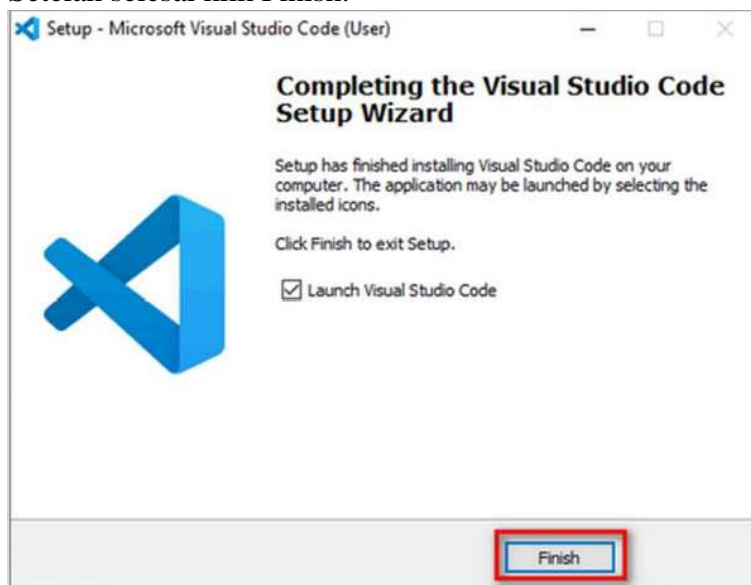
f. Kemudian klik install untuk melanjutkan proses instalasi.



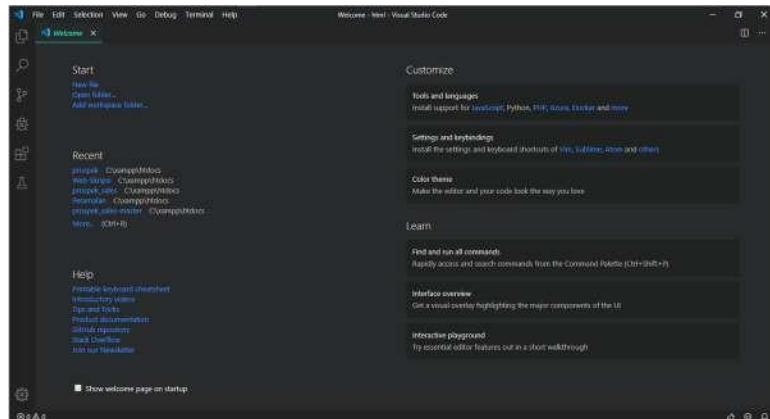
g. Tunggu proses instalasi hingga selesai.



h. Setelah selesai klik Finish.



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



2. Xampp

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

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

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

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

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

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

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

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

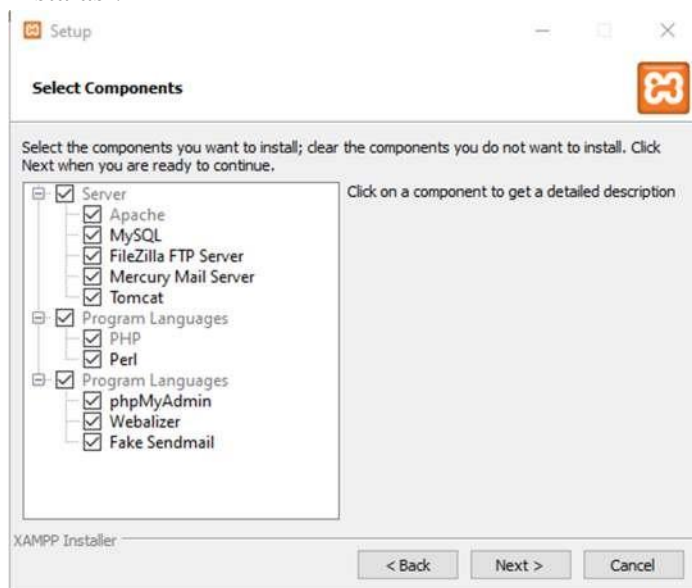
- a. Pertama, *download* terlebih dahulu file installer xampp pada link berikut ini. <https://apachefriends.org>



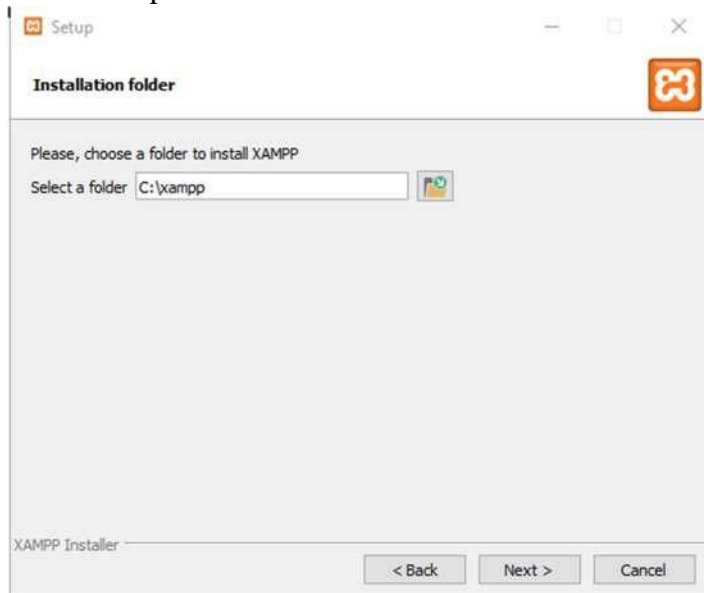
- b. Pilih file installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Klik file xampp yang sudah berhasil di *download*, maka akan muncul tampilan seperti ini.



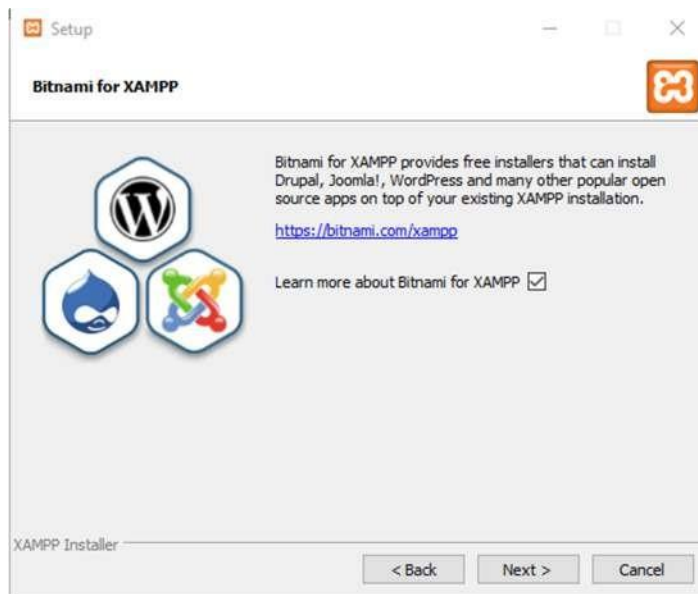
- d. Kemudian, pilih next untuk melanjutkan pada proses instalasi.



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



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



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



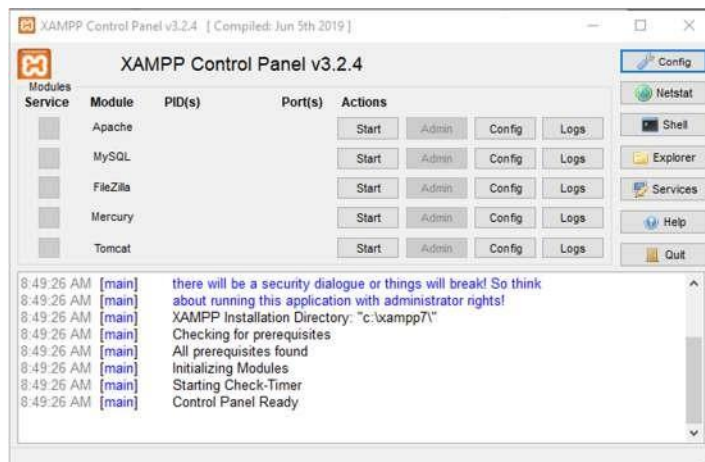
- h. Kemudian tunggu hingga proses instalasi selesai.



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



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



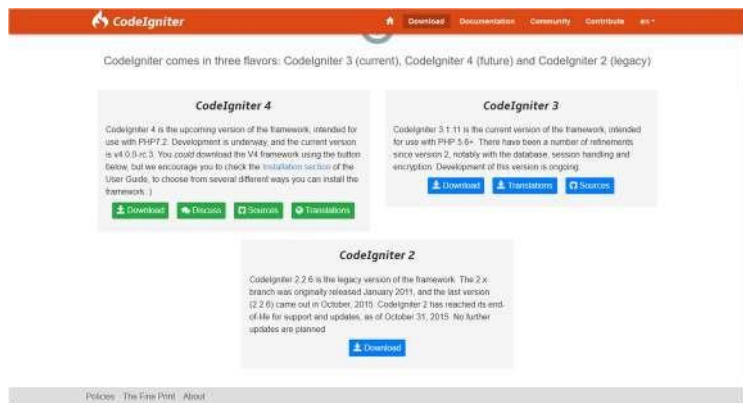
k. Jika berhasil, maka tampilan utama *control panel* dari xampp seperti berikut.

3. Codeigniter

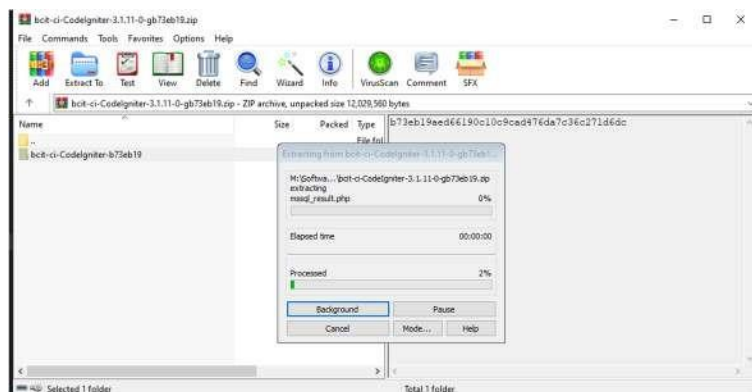
Codeigniter (CI) merupakan sebuah *web application framework* yang bersifat *open source* dimana digunakan untuk membangun aplikasi php dinamis. *Framework* itu sendiri merupakan abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh kode yang dibuat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

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

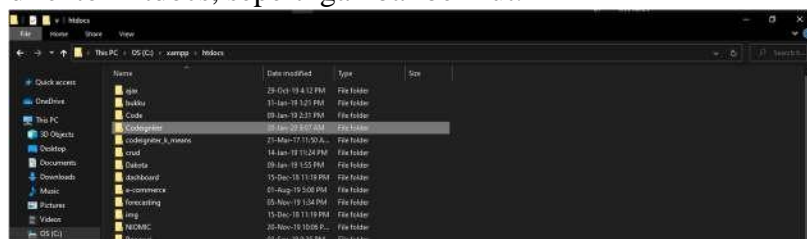
- a. Pertama, *download* terlebih dahulu codeigniter pada *website* resminya pada link berikut.
https(:)//codeigniter(dot)com/download.



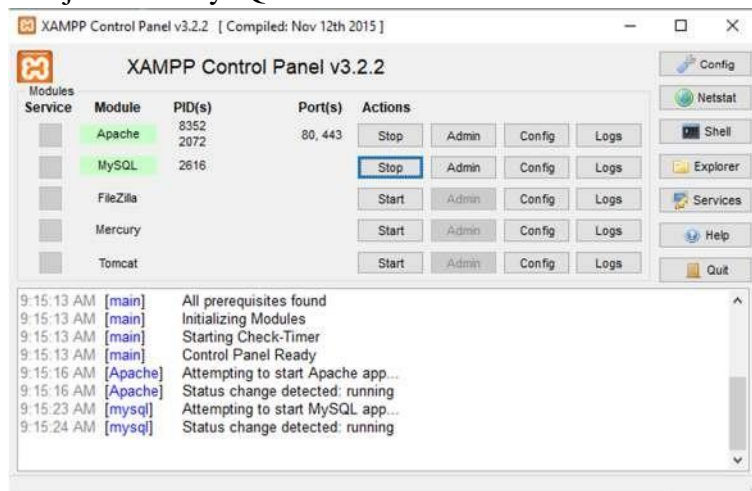
- b. Setelah berhasil di *download*, selanjutnya ekstrak file tersebut.



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



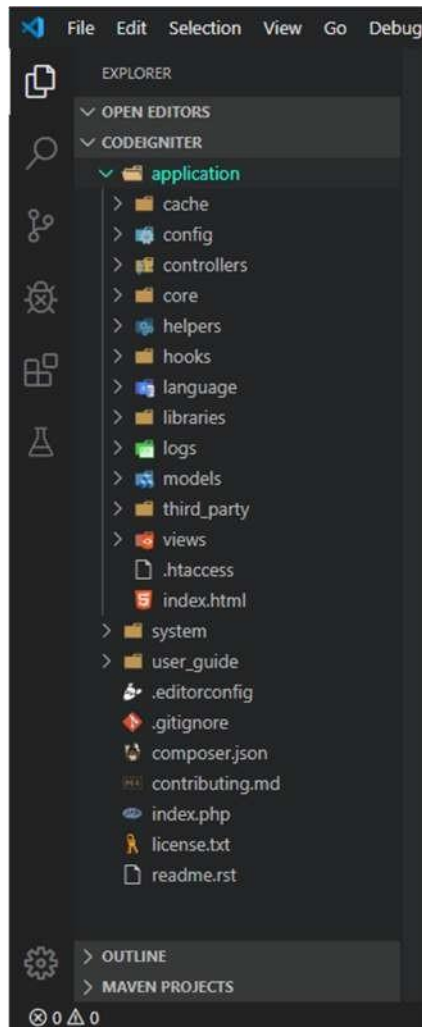
- d. Untuk menjalankannya, kita menggunakan xampp, untuk itu silakan jalankan terlebih dahulu aplikasi xampp-nya seperti berikut.
- e. Buka aplikasi xamppnya, kemudian klik tombol start pada kolom actions baris pertama untuk menjalankan apache, dan klik tombol start pada kolom actions baris kedua untuk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



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



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

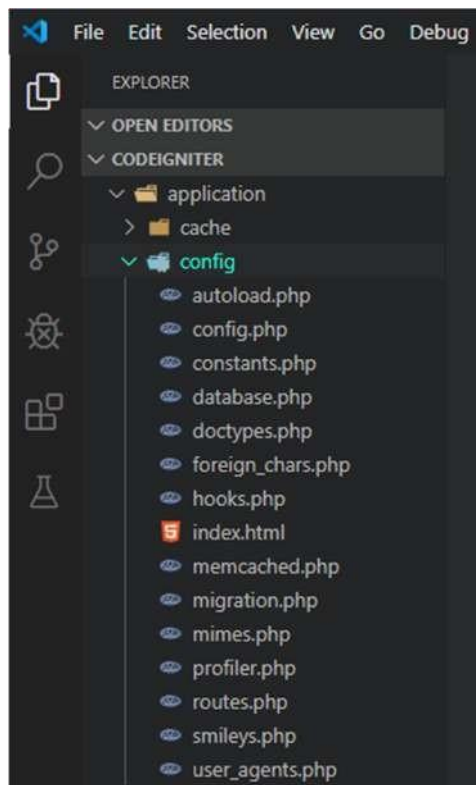


- a) *Application*, merupakan folder yang pada dasarnya menyimpan aplikasi yang sedang kita buat.
- b) *Cache*, merupakan folder yang menyimpan semua cache yang dibuat oleh cache library

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

4. Konfigurasi dasar pada codeigniter

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



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

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

- b. Kemudian temukan kode berikut.

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

- c. Ubah kode tersebut menjadi seperti berikut.

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

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

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

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

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

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


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

f. Kemudian temukan kode berikut.

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

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

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

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

```
autoload.php  config.php  database.php X
application > config > database.php
1 <?php
2 defined('BASEPATH') OR exit('no direct script access allowed');
3
4 /*
5 .....
6 DATABASE CONNECTIVITY SETTINGS
7 .....
8 This file will contain the settings needed to access your database.
9 .....
10 For complete instructions please consult the 'Database Connection'
11 page of the user guide.
12 .....
13 EXPLANATION OF VARIABLES
14 .....
15
16 [ 'dsn'        ] The full DSN string describe a connection to the database.
17 [ 'hostname'   ] the hostname of your database server.
18 [ 'username'   ] the username used to connect to the database
19 [ 'password'   ] the password used to connect to the database
20 [ 'database'   ] the name of the database you want to connect to
21 [ 'driver'     ] the database driver. e.g.: mysqli
22 [ 'prefix'     ] Currently supported:
23                  mysql, mssql, mysql, mysqli, oci8,
24                  odbc, pdo, postgre, sqlite, sqlite3, sqlsrv
25 [ 'dbprefix'   ] You can add an optional prefix, which will be added
26
```

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

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

- j. Kemudian tambahkan kode tersebut seperti gambar berikut.

```

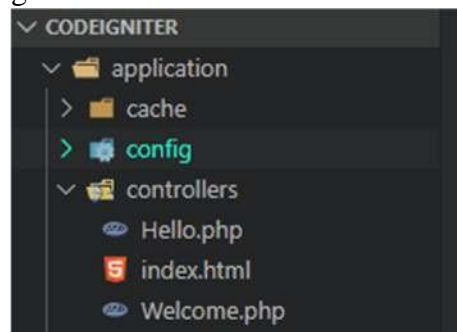
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost', //hostname
79     'username' => 'root', //username
80     'password' => '', //password
81     'database' => 'database_name', //nama database
82     'dbdriver' =>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

```

A. Hello World Codeigniter

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

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



Setelah itu, ketikkan kode seperti gambar berikut.

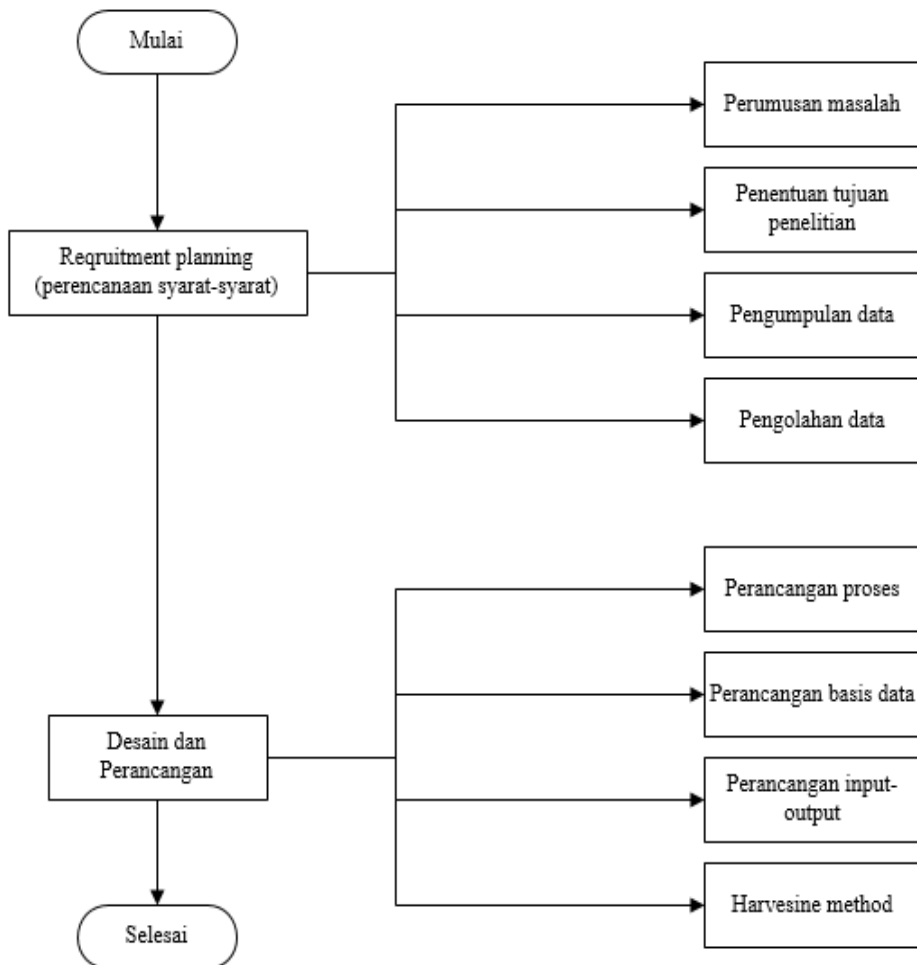
```
application > controllers > Hello.php
1  <?php
2  class Hello extends CI_Controller{
3      function index(){
4          echo "Hello World!";
5      }
6  }
7  >
```

Jika berhasil maka hasilnya akan seperti gambar berikut.



4.1 Diagram Alur Metodologi Penelitian

Metode penelitian yang di gunakan dalam penelitian ini di bagi kedalam beberapa tahap agar peroses yang dilakukan lebih terarah. Secara umum langkah-langkah penelitian yang di lakukan untuk memonitoring kinerja mahasiswa *internship* di Politeknik Pos Indonesia adalah sebagai berikut :



Gambar 4.1 Diagram Alur Metodologi Penelitian

4.2 Tahapan-Tahapan Diagram Alur Metodeologi Penelitian

Tahapan -tahapan diagram alur metodeologi penelitian yang digunakan peneliti adalah sebagai berikut :

4.2.1 Perumusan Masalah

Pada langkah ini penulis akan mencari permasalahan apa yang terjadi di perguruan tinggi, yang selanjutnya akan diteliti sehingga masalah yang akan dibahas menjadi lebih mudah dalam penentuan metode yang digunakan.

Dari hasil peninjauan masalah penulis menemukan bahwa sistem kegiatan mahasiswa *internship* di Politeknik Pos Indonesia ditemukan permasalahan yang terjadi yaitu :

1. Bagaimana memodelkan aplikasi monitoring kinerja mahasiswa *internship*.
2. Bagaimana membuat aplikasi monitoring kinerja mahasiswa *internship* berdasarkan proses bisnis yang ada.
3. Bagaimana menggunakan metode haversine untuk menghitung jarak kedua titik koordinat.

4.2.2 Penentuan Tujuan Penelitian

Pada langkah ini penulis akan menentukan tujuan dari dilakukannya penelitian ini. Tujuan dilakukannya penelitian ini yaitu untuk membangun sebuah sistem sehingga mempermudah bagian admin atau prodi dalam memonitoring kinerja mahasiswa *internship*.

1. Membuat pemodelan sistem sehingga dapat menghasilkan *warning* yang dibutuhkan.

2. Untuk lebih memaksimalkan desain *user interface* monitoring kinerja mahasiswa internship dengan aplikasi berbasis *website*.
3. Untuk menghitung jarak kedua titik koordinat pada monitoring kinerja mahasiswa *internship* dengan menggunakan metode harvesine.

4.2.3 Pengumpulan Data

Pengumpulan data yaitu pengumpulan data yang diperlukan untuk menyelesaikan laporan ini, data-data yang akan dikumpulkan oleh penulis yaitu merupakan data yang dibutuhkan dalam membangun sistem monitoring kinerja mahasiswa internship.

Dalam pengumpulan data ini penulis memperoleh dua data, yaitu :

1. Data primer merupakan data yang diperoleh secara langsung dari sumbernya diamati dan dicatat untuk pertama kalinya, dan mempunyai hubungan erat dengan permasalahan yang dihadapi perusahaan tersebut. Metode wawancara atau interview dipergunakan untuk memperoleh data dengan metode wawancara dengan narasumber yang akan diwawancarai.
2. Data sekunder adalah data yang tidak langsung memberikan data kepada peneliti, misalnya penelitian harus melalui orang lain atau mencari melalui dokumen. Data ini diperoleh dengan menggunakan studi literatur yang dilakukan terhadap banyak buku dan diperoleh berdasarkan catatan-catatan yang berhubungan dengan penelitian, selain itu peneliti mempergunakan data yang diperoleh dari internet.

4.2.4 Pengolahan Data

Setelah data-data yang diperlukan dalam penelitian ini telah terkumpul, maka langkah selanjutnya pengolahan data. Pengolahan data adalah manipulasi data agar menjadi bentuk yang lebih berguna. Data tersebut yaitu :

1. Data laporan kinerja

Dari hasil laporan kinerja yang dilakukan sesuai jadwal masuk kegiatan mahasiswa *internship* dilakukan rekap data yang bisa dijadikan *logbook* dan menghitung data laporan yang masuk serta jarak koordinat yang dihitung.

4.2.5 Analisis dan Perancangan Sistem

Dari data yang telah dikumpulkan selanjutnya dilakukan analisis sistem yang akan dibangun berdasarkan sistem yang sedang berjalan saat ini. Kemudian selanjutnya mulai dilakukan perancangan sistem berdasarkan hasil analisis kebutuhan sistem.

4.2.5.1 Perancangan Proses

Merancang proses sistem ini dengan menggunakan *tool* yang sama dengan tahap analisis sistem yaitu UML (*Unified Modelling Language*) agar lebih memahami langkah awal membangun sistem secara fisik.

4.2.5.2 Perancangan Basisdata

Merancang basis data (*database*) yang dilakukan dengan class diagram yang menggambarkan hubungan antar *entity* yang ada pada *use case* diagram dan spesifikasi tabel.

4.2.5.3 Perancangan Input-Output

Merancang Input-Output dengan membuat rancangan layar tampilan. Setelah rancangan layar tampilan terbentuk maka dilakukan tahap implementasi.

4.2.5.4 Method Harvesine

Metode Haversine Formula dapat digunakan untuk menghitung jarak antara dua titik, berdasarkan posisi garis lintang latitude dan posisi garis bujur longitude sebagai variabel inputan. Haversine Formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinat bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2, lat2 [25]. Metode Haversine Formula tersebut kini sudah mengalami pengembangan, yaitu dengan menggunakan rumus *spherical law of cosine* sederhana, dimana dengan penghitungan komputer dapat memberikan tingkat presisi yang sangat akurat antar dua titik. Pertama ditentukan terlebih dahulu titik awal dan titik tuju, titik awal berupa latitude1(lat1) dan longitude1(long1), titik tuju berupa latitude2(lat2) dan longitude2(long2). Titik awal dan titik tuju tersebut berbentuk desimal derajat yang kemudian dirubah menjadi nilai sudut radian, kemudian lakukan perhitungan dengan rumus Haversine Formula, yaitu:

Rumus Harvesine

$$x = (\text{lng2} - \text{lng1}) * \cos ((\text{lat1} + \text{lat2})/2);$$

$$y = (\text{lat2} - \text{lat1});$$

$$d = \text{sqrt}(x*x + y*y) * R$$

Keterangan :

x = longitude (lintang)

y = latitude (bujur)

d = jarak (km)

R = Radius Bumi = 6371 km

1 derajat = 0.0174532925 radian

Apabila metode ini di implemetasikan pada sistem yang akan dibangun maka proses yang dilakukan menyesuaikan dengan kebutuhan pada aplikasi tersebut yang dimana titik koordinat pertama *latitude, longitude* adalah sebagai titik acuan yang diambil dari data registrasi mahasiswa/i, sedangkan untuk titik koordinat kedua mengambil dari data *report activity* harian yang mana hasil akhir dari perhitungan kedua jarak digunakan sebagai data kehadiran mahasiswa/i dalam bentuk penilaian oleh pembimbing.

5.1. Analisis dan Perancangan Sistem

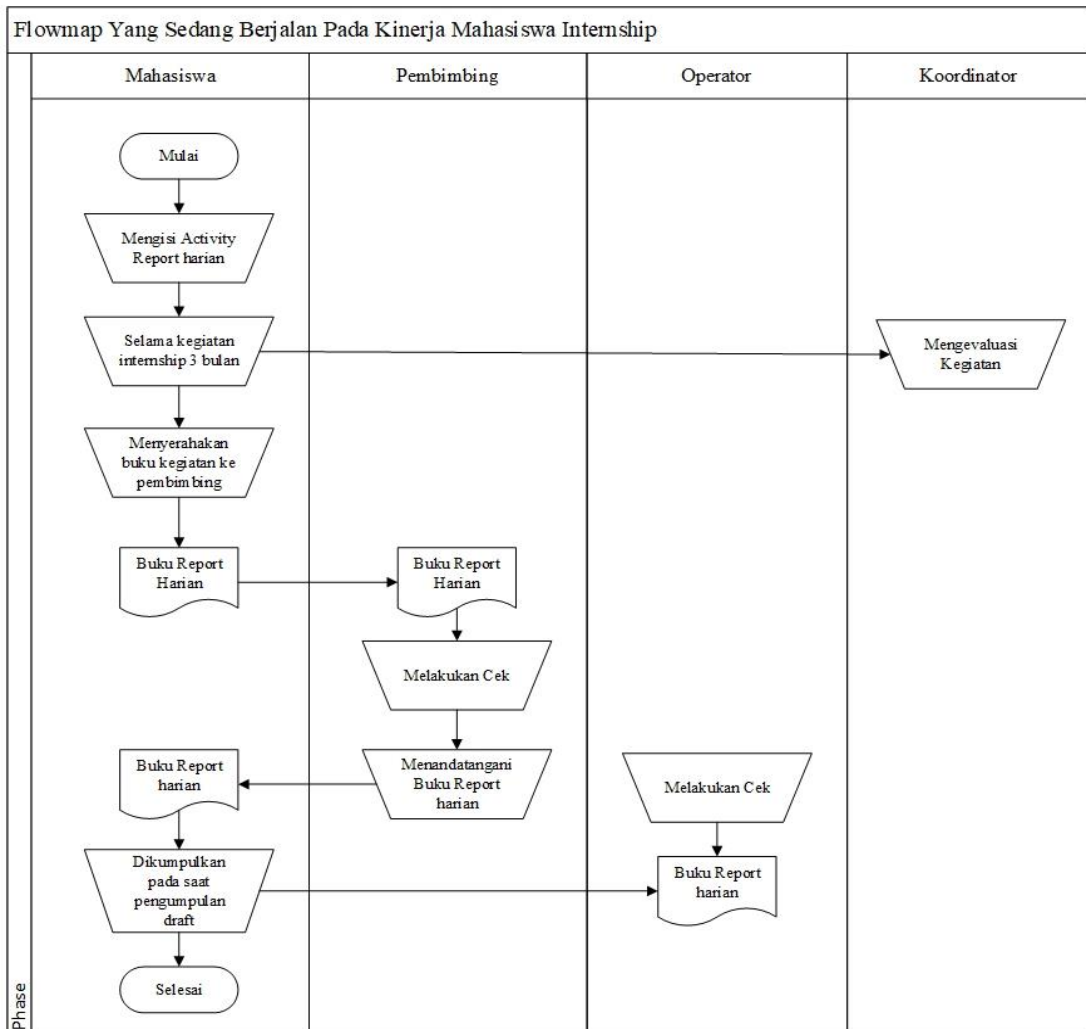
Perencanaan sistem menyangkut estimasi dari kebutuhan-kebutuhan fisik, tenaga kerja dan dana yang dibutuhkan untuk mendukung pengembangan sistem ini serta untuk mendukung operasinya setelah diterapkan sedangkan analisis merupakan proses untuk menentukan bentuk dari kebutuhan sistem yang menunjang kebutuhan pada saat membangun dan implementasi. Secara garis besar disebut juga sebagai proses mempelajari aktifitas system untuk memahami gambaran menyeluruh tentang sehingga perancang telah mengetahui apa saja kebutuhan dari sistem tersebut.

5.1.1. Analisis Sistem Berjalan (*Current System*)

Sistem kegiatan mahasiswa *internship* di Politeknik Pos Indonesia saat ini sedang berjalan masih menggunakan cara manual lebih khususnya dibagian pengawasan kinerja atau laporan kegiatan harian yang dilakukan mahasiswa tersebut. Kendala yang terjadi adalah kurang efektif karena data dapat dimanipulasi apabila dilakukan dengan cara manual sehingga data yang dibutuhkan kurang valid.

5.1.1.1. Analisis Prosedur yang berjalan (*Flowmap*)

Flowmap untuk analisis prosedur yang sedang berjalan dibuat agar alur sistem monitoring kinerja mahasiswa *internship* yang sedang berjalan dapat lebih mudah dipahami. Berikut adalah analisis sistem yang sedang berjalan dalam bentuk *flowmap* atau prosedur :



Gambar 5.1 Flowmap kegiatan harian internship yang sedang berjalan

Keterangan :

Pada *flowmap* tersebut dapat dijelaskan bahwa mahasiswa mengisi formulir kehadiran kegiatan *internship*. Di dalam buku tersebut tercantum form catatan yang harus diisi oleh mahasiswa/i

selama kegiatan *internship* berlangsung yaitu 3 (tiga) bulan untuk sebagai *report* harian, *activity* apa saja yang dilakukan oleh mahasiswa/i pada tempat *internship*nya. Kemudian mahasiswa/i menyerahkan *buku/logbook* tersebut kepada pembimbing untuk dilakukannya pengecekan serta menandatangani *report* harian dan operator melakukan pengecekan pada saat pengumpulan *draft*.

5.1.1.2. Anlisis Dokumen Yang Digunakan

Analisis dokumen yang digunakan merupakan tahap analisis terhadap beberapa dokumen yang digunakan, yaitu :

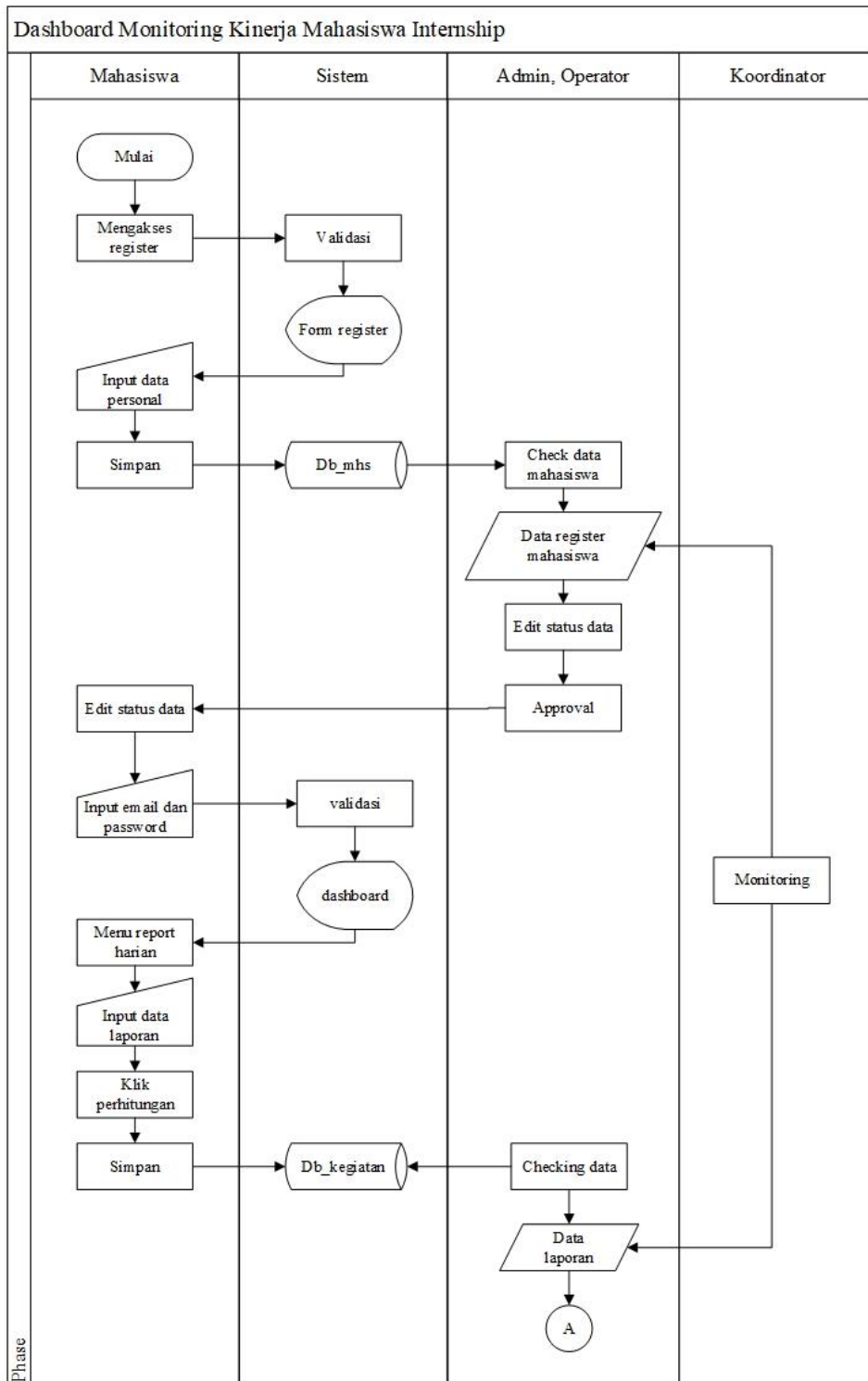
1. Formulir *report* harian , berupa form catatan aktivitas yang dilakukan dan menjadi sebagai buku *logbook*.
2. Dokumen data laporan, menjelaskan mengenai data-data laporan yang dikirimkan oleh mahasiswa berupa laporan atau materi ke pembimbing eksternal/mentor.

5.1.2. Analisis Sistem yang akan dibangun

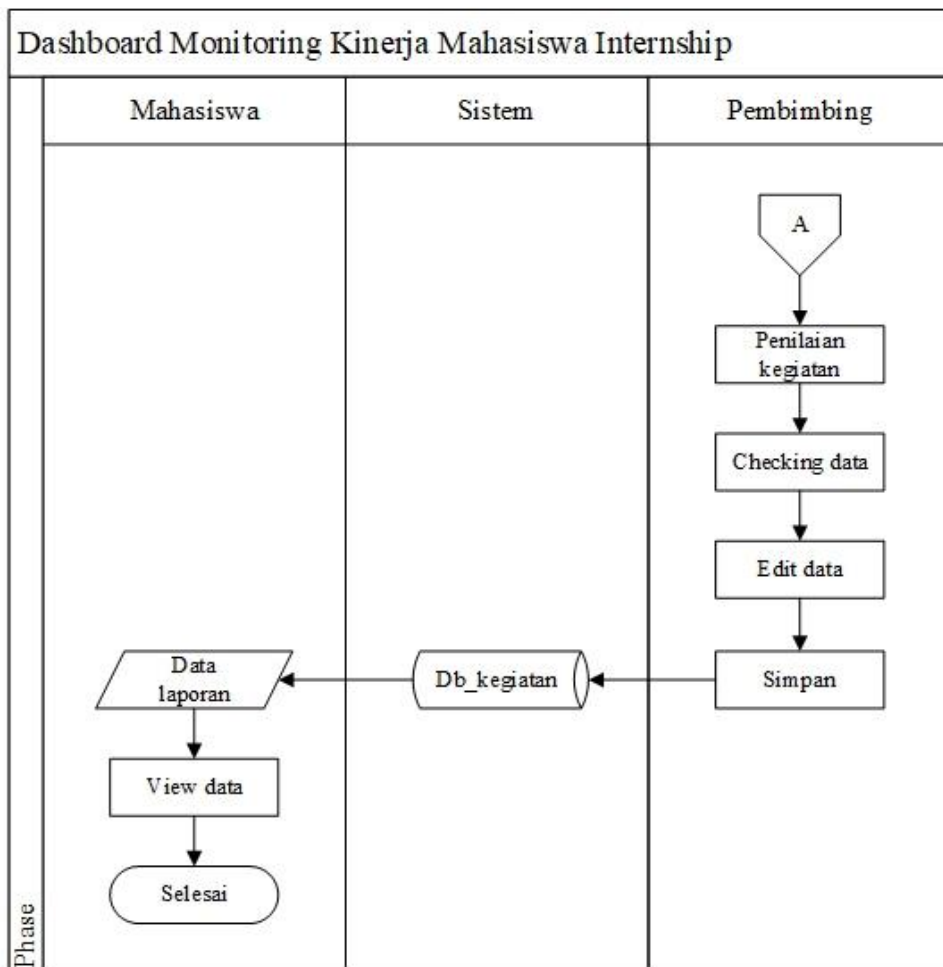
Dari analisis yang telah dilakukan maka akan dibangun sistem yang dapat memenuhi kebutuhan perusahaan/ perguruan tinggi dalam pengolahan data kinerja mahasiswa *internship*. Sistem ini sebelumnya dilakukan secara manual dan akan dibuat terkomputerisasi sehingga proses bisnis yang terjadi dapat lebih baik lagi dan meng-efisien-kan kinerja mahasiswa yang *internship* dan memperkecil kesalahan dalam data laporan.

Proses-proses yang sebelumnya dilakukan secara manual akan dijadikan sistem yang mempermudah user. Sistem yang akan dibangun memfokuskan pada pengolahan data laporan kinerja sehingga proses lebih efektif dapat berjalan sesuai kebutuhan user.

5.1.2.1 Analisis Prosedur yang akan dibangun (*flowmap*)



Gambar 5.2 Flowmap monitoring kinerja mahasiswa internship yang akan dibangun



Gambar 5.3 Flowmap monitoring kinerja mahasiswa internship yang akan dibangun

Keterangan :

Pada gambar *flowmap* diatas dapat dijelaskan aktor pertama dimulai dari mahasiswa melakukan tahap registrasi lalu tahap kedua aktor yang akan menerima data registrasi adalah operator untuk mengubah status data. Setelah data di *approval* maka operator memberikan hak akses untuk login kepada mahasiswa lalu sistem akan memvalidasi akun tersebut. Selanjutnya sistem akan

menampilkan halaman *dashboard* dan mahasiswa memilih menu *report* harian kemudian meng-input data kegiatan yang dilakukan *user* tersebut lalu menyimpannya. Data kegiatan yang sudah dilakukan aksi maka akan menjadi data laporan yang tersimpan pada *database*, kemudian aktor pembimbing prodi DIV Teknik Informatika akan melakukan tahap *check-ing* menyeluruh data laporan untuk dilakukannya penilaian sebagaimana proses tersebut sama dengan koordinator hanya perbedaan dari segi aksi yaitu hanya me-monitoing ke seluruh data.

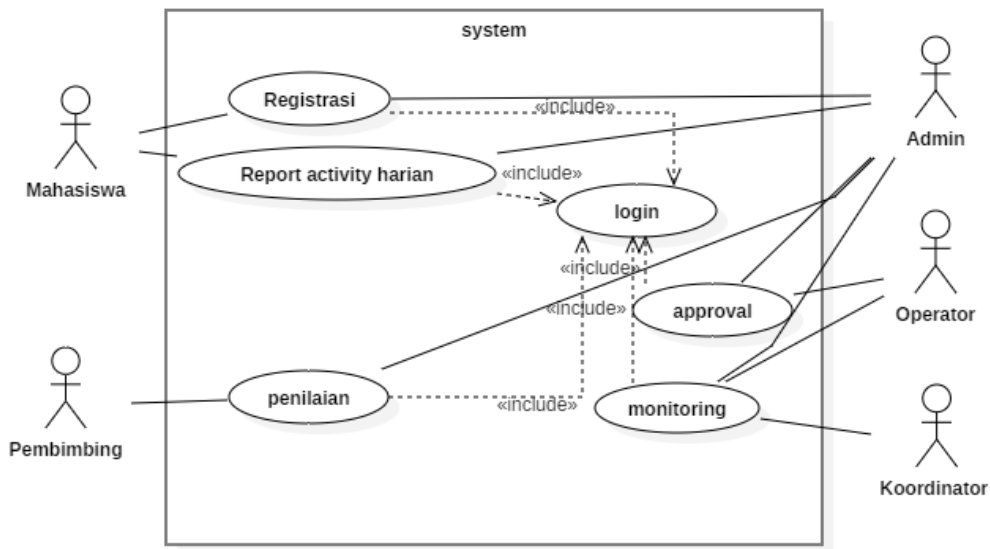
5.1.2.2. Analisis Dokumen yang dibangun

Dalam sistem monitoring kinerja mahasiswa *internship* ini terdapat beberapa dokumen yang digunakan, yaitu :

1. Dokumen data laporan, menjelaskan mengenai data-data laporan yang dikirimkan oleh mahasiswa berupa laporan atau materi ke pembimbing untuk dijadikan sebagai penilaian.

5.2. UML (Unified Modelling Language)

5.2.1. Usecase Diagram



Gambar 5.4 Usecase Diagram Dashboard Monitoring Kinerja

Mahasiswa Internship

5.2.1.1. Definsi Aktor

Pada bagian ini akan dijelaskan aktor-aktor yang terlibatkan didalam sistem:

Tabel 5.1 Definisi Aktor

Id	Aktor	Deskripsi
1	Admin	Admin adalah pihak yang bertanggung jawab atas segala pengolahan data yang ada didalam sistem.
2	Koordinator	Koordinator adalah pihak yang bertanggung jawab serta melakukan koordinasi kegiatan yang dilakukan.
3	Mahasiswa	Mahasiswa adalah pihak yang memiliki hak untuk melakukan pengisian <i>report activity</i> harian.
4	Operator	Operator adalah pihak yang bertugas menjaga, melayani, menjalankan suatu peralatan, mesin, telepon dan sebagainya
5	Pembimbing	Pembimbing adalah pihak yang bertugas untuk melakukan penilaian tugas yang dilakukan oleh mahasiswa.

5.2.1.2. Definisi Usecase

Pada bagian ini akan dijelaskan usecase yang ada didalam sistem

:

Tabel 5.2 Definisi Usecase

	Usecase	Deskripsi
	registrasi	Merupakan aktivitas untuk melakukan pendaftaran data diri sehingga terhubung dengan sistem.
	Report activity harian	Merupakan aktivitas untuk mencatat kegiatan yang dilakukan dalam bentuk sebagai laporan.
	Approval	Merupakan aktivitas untuk melakukan penerimaan data apabila sesuai dengan kebutuhan.
	Penilaian	Merupakan aktivitas untuk melakukan pengolahan informasi dan mengukur terkait pencapaian hasil kinerja yang telah dilakukan.
	Monitoring	Merupakan suatu kegiatan mengamati secara seksama suatu keadaan atau kondisi, termasuk juga perilaku atau kegiatan tertentu.
	Akses	Merupakan pemberian hak akses kepada pihak yang terkait dengan sistem (aktor).

5.2.1.3. Skenario Usecase

Skenario untuk tiap masing-masing usecase diatas adalah :

Tabel 5.3 Skenario Diagram Usecase Registrasi

Identifikasi	
Usecase	C1
Nama	registrasi
Deskripsi	Merupakan aktivitas untuk melakukan pendaftaran data diri sehingga terhubung dengan sistem.
Deskripsi	
Aktor	Mahasiswa/i
Skenario Utama	
Kondisi Awal	Form sudah tersedia
Kondisi Akhir	Data registrasi mahasiswa telah berhasil ditambahkan.
Main Flow Event	
Aksi aktor	Aksi sistem
1. Mahasiswa memilih <i>link register a new membership</i>	2. Sistem akan menampilkan form registrasi mahasiswa
3. Mahasiswa akan melakukan <i>insert</i> data diri pada form yang tersedia.	4. Mengambil data dari <i>database</i> kemudian terjadi eksekusi terhadap perintah yang diberikan dan nantinya akan disimpan di <i>database</i>
Exceptional Flow of Event	Jika aktor salah dalam memasukkan data maka sistem akan menampilkan pesan <i>error</i> . Sedangkan jika data valid maka data akan disimpan di

	<i>database.</i>
--	------------------

Tabel 5.4 Skenario Diagram Usecase Report activity harian

Identifikasi	
Usecase	C2
Nama	<i>report activity harian</i>
Deskripsi	Merupakan aktivitas untuk mencatat kegiatan yang dilakukan dalam bentuk sebagai laporan.
Deskripsi	
Aktor	Mahasiswa/i
Skenario Utama	
Kondisi Awal	Form sudah tersedia
Kondisi Akhir	Data <i>report</i> harian telah berhasil ditambahkan.
Main Flow Event	
Aksi aktor	Aksi sistem
1. Mahasiswa memilih menu <i>report</i> harian	2. Sistem akan menampilkan form <i>report activity harian</i> .
3. Mahasiswa akan melakukan <i>insert</i> data laporan form yang tersedia dan melakukan perhitungan.	4. Mengambil data dari <i>database</i> kemudian terjadi eksekusi terhadap perintah yang diberikan dan nantinya akan disimpan di <i>database</i>
Exceptional Flow of Event	Jika aktor salah dalam memasukkan data maka sistem akan menampilkan pesan <i>error</i> . Sedangkan jika data valid maka data akan disimpan di <i>database</i> .

Tabel 5.5 Skenario Diagram Usecase approval

Identifikasi	
Usecase	U3
Nama	approval
Deskripsi	Merupakan aktivitas untuk melakukan penerimaan data apabila sesuai dengan kebutuhan.
Deskripsi	
Aktor	Admin, Operator
Skenario Utama	
Kondisi Awal	Form sudah tersedia
Kondisi Akhir	Data mahasiswa telah berhasil dikelola (<i>insert, update, delete, view</i>).
Main Flow Event	
Aksi aktor	Aksi sistem
1. Aktor memilih menu data mahasiswa	2. Sistem akan menampilkan form isian data mahasiswa
3. Aktor akan melakukan <i>insert, update, view</i> atau <i>delete</i> data.	4. Mengambil data dari <i>database</i> kemudian terjadi eksekusi terhadap perintah yang diberikan dan nantinya akan disimpan di <i>database</i>
Exceptional Flow of Event	Jika aktor salah dalam memasukkan data maka sistem akan menampilkan pesan <i>error</i> . Sedangkan jika data valid maka data akan disimpan di <i>database</i> .

Tabel 5.6 Skenario Diagram Usecase penilaian

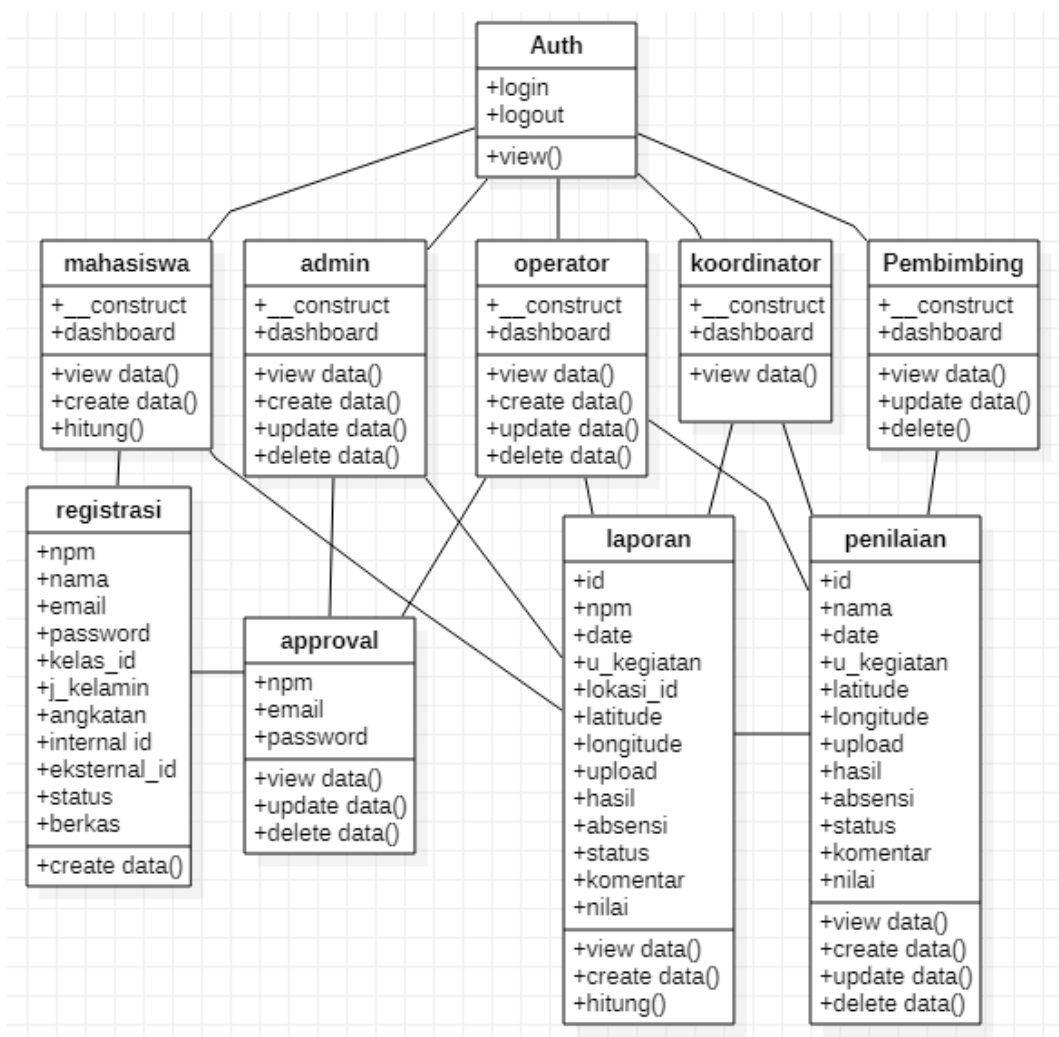
Identifikasi	
Form	C4
Nama	Penilaian
Fungsi	Merupakan aktivitas untuk melakukan pengolahan informasi dan mengukur terkait pencapaian hasil kinerja yang telah dilakukan.
Deskripsi	
Aktor	Penyembimbing
Skenario Utama	
Kondisi Awal	Form sudah tersedia
Kondisi Akhir	Data laporan telah berhasil dikelola (<i>insert, update, delete, view</i>).
Main Flow Event	
Aksi aktor	Aksi sistem
1. Aktor memilih menu penilaian, selanjutnya mengklik <i>detail</i> data.	2. Sistem akan menampilkan form isian data laporan mahasiswa per- <i>progress</i>
3. Aktor akan melakukan <i>insert, update, view</i> atau <i>delete</i> data.	4. Mengambil data dari <i>database</i> kemudian terjadi eksekusi terhadap perintah yang diberikan dan nantinya akan disimpan di <i>database</i>
Exceptional Flow of Event	Jika aktor salah dalam memasukkan data maka sistem akan menampilkan pesan <i>error</i> . Sedangkan jika data valid maka data akan disimpan di <i>database</i> .

Tabel 5.7 Skenario Diagram Monitoring

Identifikasi	
Domor	C5
nama	Monitoring
Definisi	Merupakan aktivitas untuk melakukan pengolahan informasi dan mengukur terkait pencapaian hasil kinerja yang telah dilakukan.
Deskripsi	
Aktor	Koordinator
Skenario Utama	
Kondisi Awal	Form sudah tersedia
Kondisi Akhir	Formulir data yang dikelola. (<i>view</i>).
Main Flow Event	
Aksi aktor	Aksi sistem
1. koordinator semua menu yang ada pada aplikasi.	2. Sistem akan menampilkan form isian data dari setiap menu yang dibuka
3. Koordinator akan <i>view</i> semua data.	4. Mengambil data dari <i>database</i> kemudian terjadi eksekusi terhadap perintah yang diberikan dan nantinya akan disimpan di <i>database</i>
Exceptional Flow of Event	Jika aktor salah dalam memasukkan data maka sistem akan menampilkan pesan <i>error</i> . Sedangkan jika data valid maka data akan disimpan di <i>database</i> .

5.3 Class Diagram

5.3.1 Class Diagram

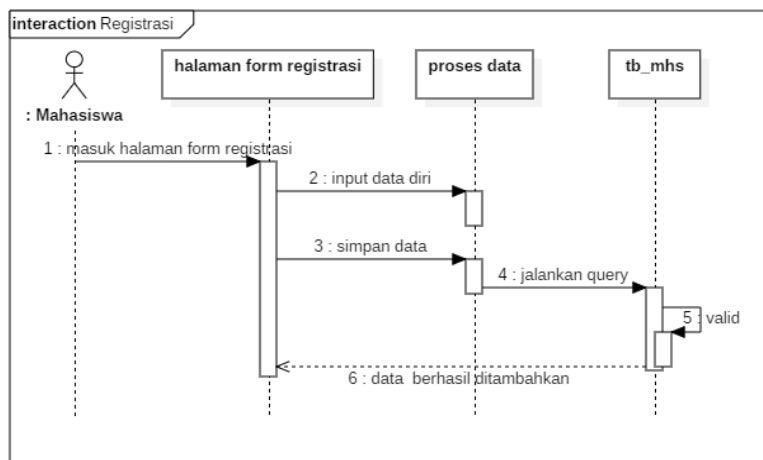


Gambar 5.5 Class Diagram

5.3.2 Sequence Diagram

Sequence diagram digunakan untuk memodelkan pengiriman pesan (*message*) antar *objects*. *Sequence* diagram menjelaskan secara detail urutan proses yang dilakukan dalam system untuk mencapai tujuan dari *use case*, intraksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi dan informasi yang diperlukan oleh masing-masing operasi.

1. Sequence Diagram Registrasi



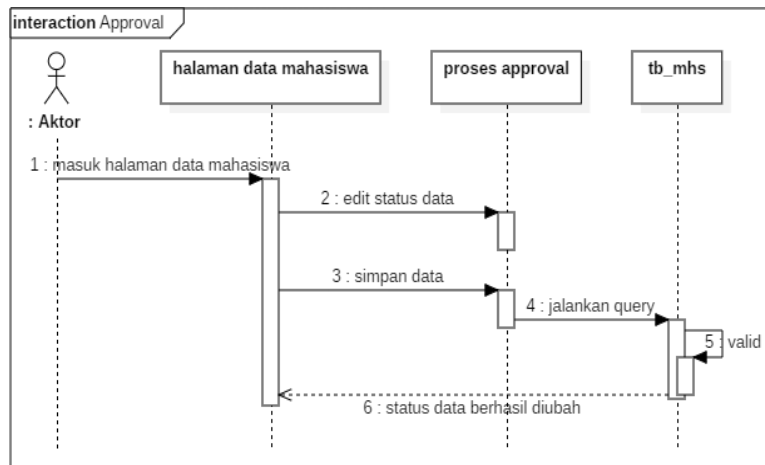
Gambar 5.6 Sequence Diagram Registrasi

Keterangan :

Pada *sequence* diagram ini menjelaskan proses registrasi. Aktor yang melakukan proses tersebut adalah mahasiswa. Pertama mahasiswa masuk pada halaman *form* registrasi lalu meng-input-kan data dirinya. Apabila data diri mahasiswa sudah dilengkapi maka proses selanjutnya adalah menyimpan data dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu

tb_mahasiswa. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal yang dimunculkan dengan *reply message(alert)*.

2. Sequence Diagram Approval

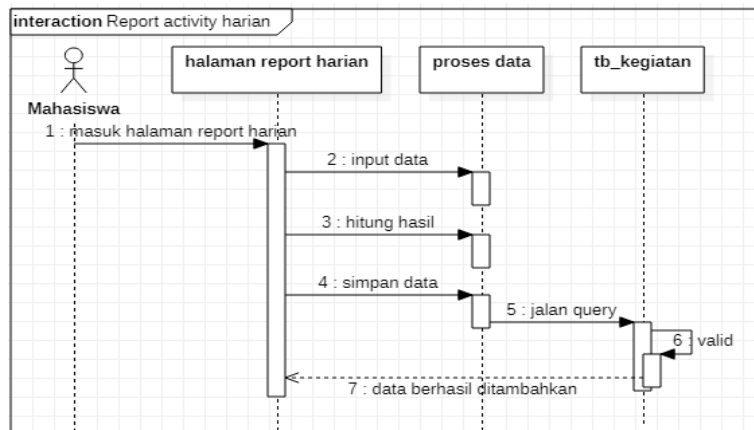


Gambar 5.7 Sequence Diagram Approval

Keterangan :

Pada *sequence* diagram ini menjelaskan proses *approval* yang dilakukan oleh admin dan operator. Proses yang dilakukan adalah menerima data registrasi mahasiswa/i untuk memberi hak akses masuk ke aplikasi. Pertama aktor masuk ke halaman data mahasiswa lalu meng-*edit* status datanya, kemudian melakukan perintah simpan data dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu tb_mahasiswa. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal yang dimunculkan dengan *reply message(alert)*.

3. Sequence Diagram Report Activity Harian

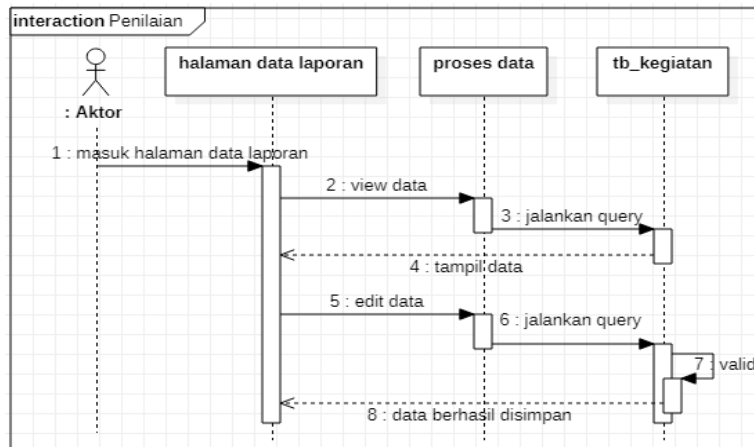


Gambar 5.8 Sequence Diagram Report Activity
Harian

Keterangan :

Pada *sequence* diagram ini menjelaskan proses untuk meng-*input* data kegiatan laporan harian yang dilakukan ditempat *internship* dan aktor yang melakukannya adalah mahasiswa. Untuk melakukan proses tersebut mahasiswa masuk ke halaman *report* harian yang sudah tersedia *form input*-an, lalu mengisi data sesuai dengan kegiatan yang dilakukan serta melakukan perhitungan jarak dan proses selanjutnya menyimpan data, kemudian dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tb_kegiatan*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal yang dimunculkan dengan *reply message(alert)*.

4. Sequence Diagram Penilaian

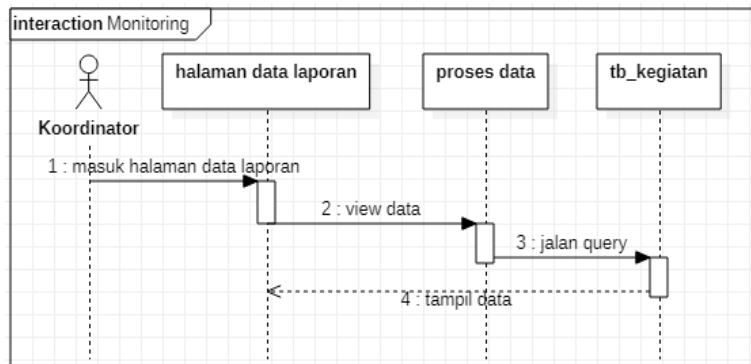


Gambar 5.9 Sequence Diagram Penilaian

Keterangan :

Pada *sequence* diagram ini menjelaskan proses penilaian terhadap kinerja mahasiswa dalam konteks me-monitoring dari data laporan masuk atau kegiatan yang dilakukan yang dituangkan dalam laporan. Dibalik proses tersebut aktor yang melakukannya adalah pembimbing, pertama aktor masuk ke halaman data laporan. Kemudian sistem akan menampilkan semua data laporan masuk, lalu aktor akan melakukan *edit* data untuk memberikan keterangan absensi, status, nilai dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tb_penilaian*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal yang dimunculkan dengan *reply message(alert)*.

5. Sequence Diagram Monitoring



Gambar 5.10 Sequence Diagram Monitoring

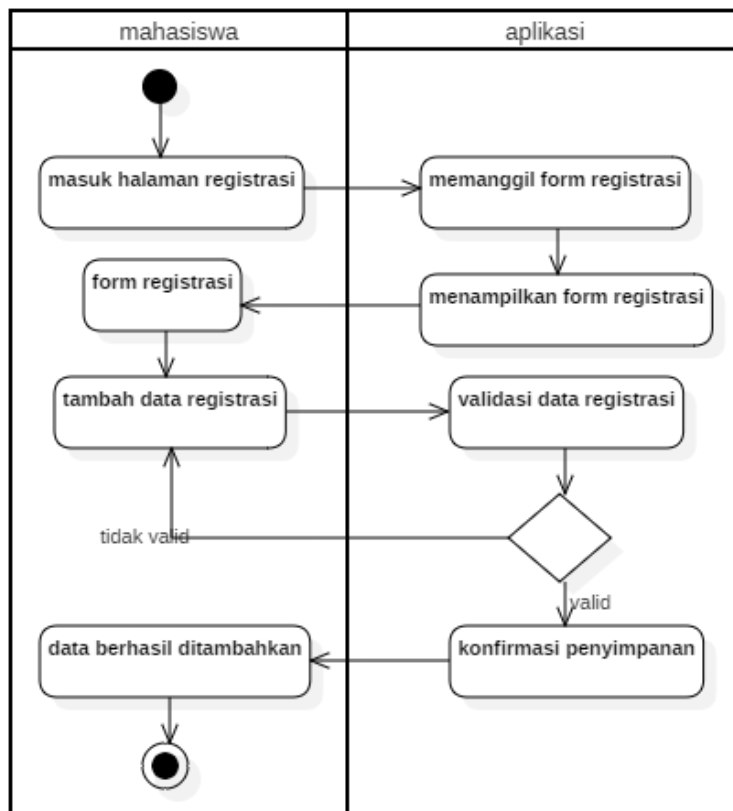
Keterangan :

Pada *sequence* diagram ini menjelaskan proses kelola laporan yang dilakukan oleh aktor koordinator. Pertama aktor masuk ke halaman data laporan dan sistem akan menampilkan semua data laporan serta menampilkan *button* aksi yaitu *view data*. Selanjutnya aktor meng-klik *button* tersebut dan dilanjutkan dengan proses *OpenTable* ke tabel yang dituju yaitu *tb_kegiatan*. Selanjutnya sistem akan memvalidasi hasil dari operasi, apakah berhasil atau gagal yang dimunculkan dengan *reply message(alert)*.

5.3.3 Activity Diagram

Activity diagram menggambarkan berbagi aliran aktivitas dalam system yang sudah dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berhasil. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

1. Activity Diagram Registrasi



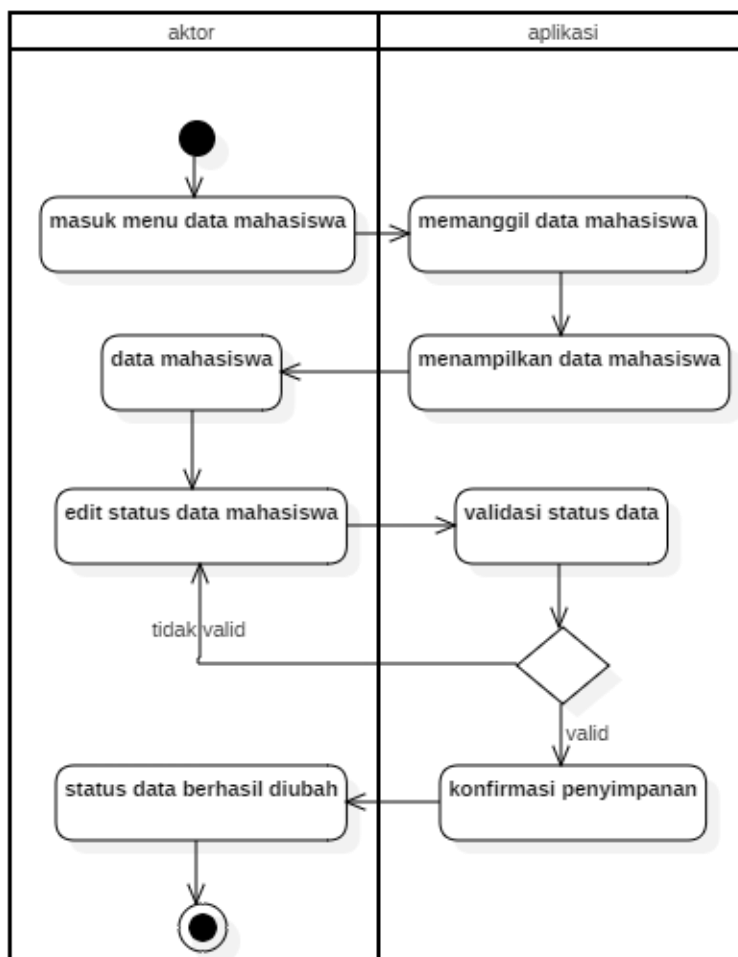
Gambar 5.11 Activity Diagram Registrasi

Keterangan :

Pada *activity* diagram ini menjelaskan proses registrasi yang dilakukan oleh aktor mahasiswa. Mahasiswa masuk ke

halaman registrasi lalu sistem menampilkan halaman *form* registrasi ke mahasiswa. Setelah itu mahasiswa dapat melakukan pengolahan data registrasi. Pengolahan yang pertama yaitu tambah data registrasi yang dimana sistem akan melakukan pengecekan apakah data sudah ditambahkan atau belum ditambahkan, apabila telah ditambahkan maka sistem dapat mengkonfirmasi penyimpanan dan menampilkan *alert*.

2. Activity Diagram Approval

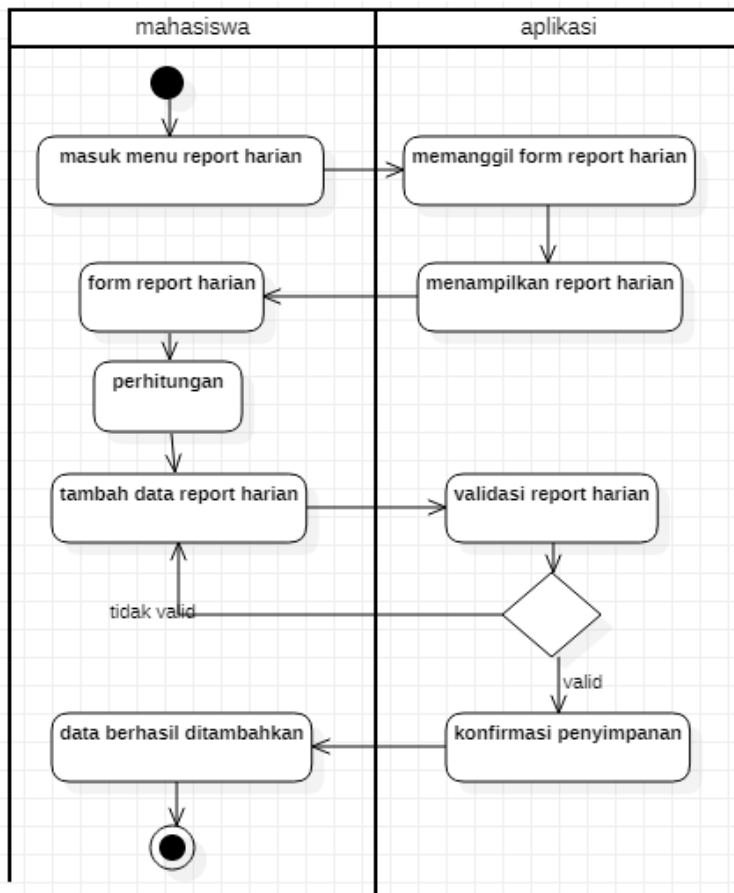


Gambar 5.12 Activity Diagram Approval

Keterangan :

Pada activity diagram ini menjelaskan proses approval yang dilakukan oleh aktor admin dan operator. Aktor masuk menu data mahasiswa lalu sistem menampilkan halaman data mahasiswa ke aktor. Setelah itu aktor dapat pengolahan data mahasiswa. Pengolahan yang pertama yaitu edit status data mahasiswa yang dimana sistem akan melakukan pengecekan apakah data sudah diubah atau sebaliknya, apabila telah diubah sistem dapat mengkonfirmasi penyimpanan dan menampilkan *alert*.

3. Activity Diagram Report Activity Harian

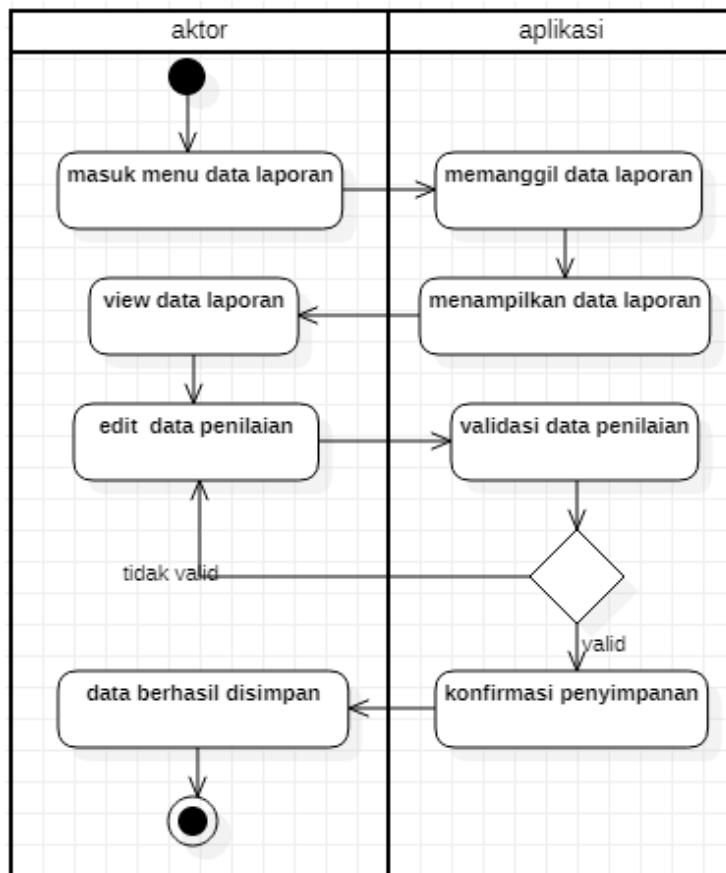


Gambar 5.13 Activity Diagram Report Activity Harian

Keterangan :

Pada *activity* diagram ini menjelaskan proses *report* harian yang dilakukan oleh aktor mahasiswa. Mahasiswa masuk menu *report* harian kemudian sistem menampilkan halaman data laporan ke mahasiswa. Setelah itu mahasiswa dapat melakukan pengolahan data laporan. Pengolahan yang pertama yaitu tambah data laporan dan perhitungan yang dimana sistem akan melakukan pengecekan apakah data sudah ditambahkan atau belum ditambahkan, apabila telah ditambahkan maka sistem dapat mengkonfirmasi penyimpanan dan menampilkan *alert*.

4. Activity Diagram Penilaian



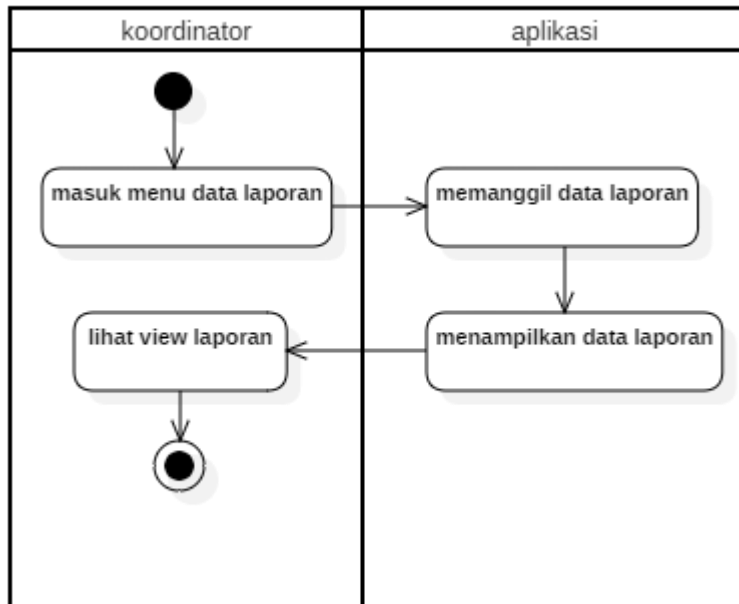
Gambar 5.14 Activity Diagram Penilaian

Keterangan :

Pada *activity* diagram ini menjelaskan proses penilaian yang dilakukan oleh aktor pembimbing. Admin masuk menu data laporan lalu sistem menampilkan halaman data laporan ke admin. Setelah itu admin dapat melakukan pengolahan data laporan. Pengolahan yang pertama yaitu detail data laporan yang dimana sistem akan menampilkan data-data keterangan dari laporan itu sendiri dan pengolahan yang kedua yaitu tambah data penilaian yang dimana sistem akan melakukan pengecekan apakah data sudah ditambahkan atau belum ditambahkan, apabila telah

ditambahkan maka sistem dapat mengkonfirmasi penyimpanan dan menampilkan *alert*.

5. Activity Diagram Monitoring



Gambar 5.15 Activity Diagram Monitoring

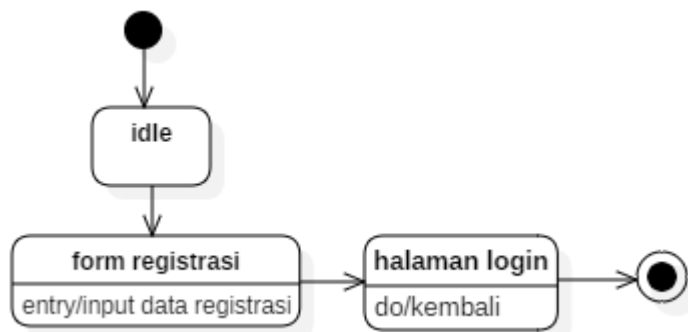
Keterangan :

Pada *activity* diagram ini menjelaskan proses kelola laporan yang dilakukan oleh aktor koordinator. Aktor masuk menu data laporan dan sistem menampilkan halaman data laporan ke aktor. setelah itu aktor dapat melakukan *check*-ing data laporan dengan memberikan aksi pada *button view*, lalu sistem menampilkan keterangan data laporan yang lengkap.

5.3.4 Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan dari suatu *state* ke *state* lainnya. Suatu objek pada *system* sebagai akibat dari stimulasi yang diterima *statechart* diagram mendeskripsikan bagaimana suatu objek mengalami perubahan status adanya *trigger* dan *event-event*. Menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek.

1. *Statechart* Diagram Registrasi

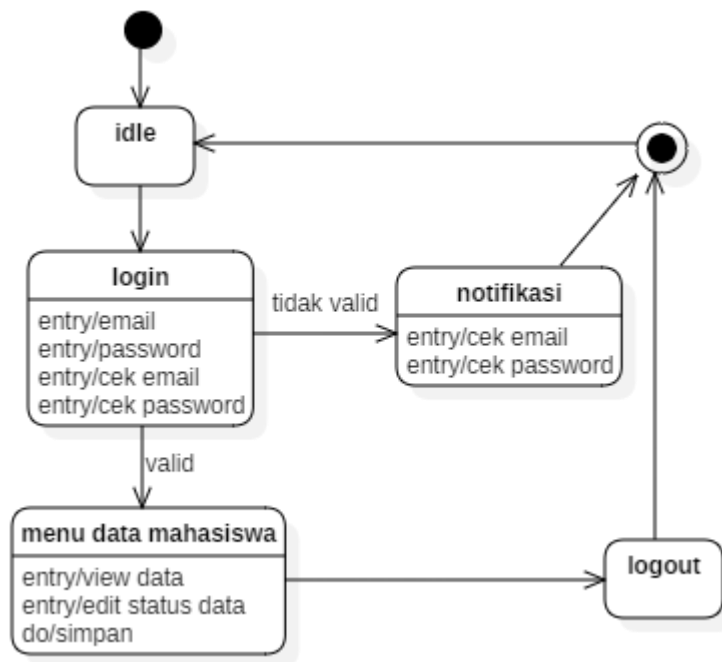


Gambar 5.16 Statechart Diagram Registrasi

Keterangan :

Pada *statechart* diagram ini menjelaskan proses program registrasi yang dilakukan oleh aktor mahasiswa. Aplikasi dalam keadaan *idle* dan status aplikasi berubah saat mahasiswa melakukan aksi yaitu meng-*input* data pada form registrasi. Setelah melakukan operasi tersebut, aktor kembali pada halaman *login*.

2. Statechart Diagram Approval

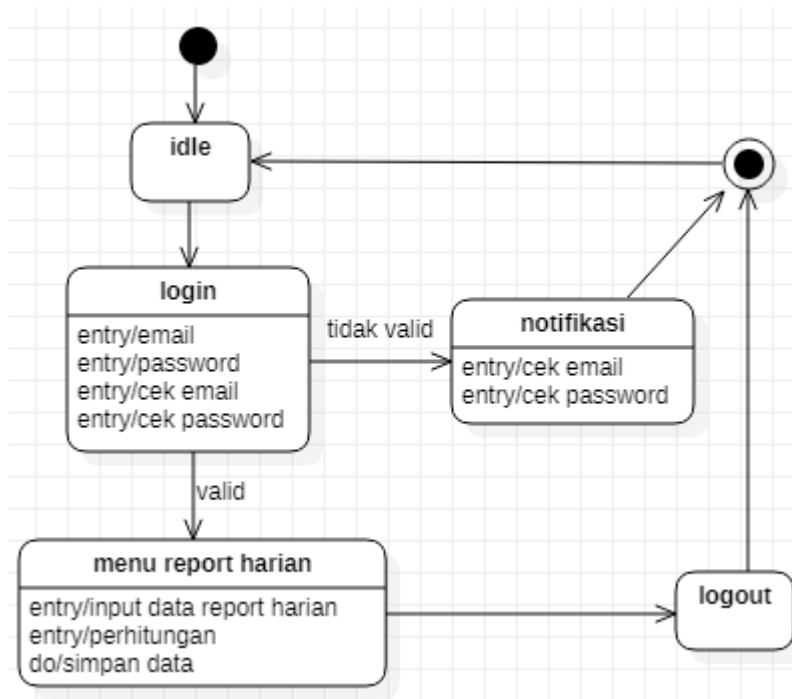


Gambar 5.17 Statechart Diagram Approval

Keterangan :

Pada *statechart* diagram ini menjelaskan proses *approval* data mahasiswa yang dilakukan oleh aktor admin dan operator. Aplikasi dalam keadaan *idle*, kemudian status aplikasi berubah saat aktor melakukan *login*. Apabila login valid maka aktor di alihkan ke halaman utama dan jika *login* tidak valid maka aktor kembali ke halaman *login*. Setelah aktor berhasil melakukan *login* dan masuk ke menu data mahasiswa maka aktor dapat melakukan pengolahan berupa *view*, *edit* dan hapus data mahasiswa. Setelah melakukan pengolahan data mahasiswa maka aktor dapat melakukan *logout*.

3. Statechart Diagram Report Activity Harian

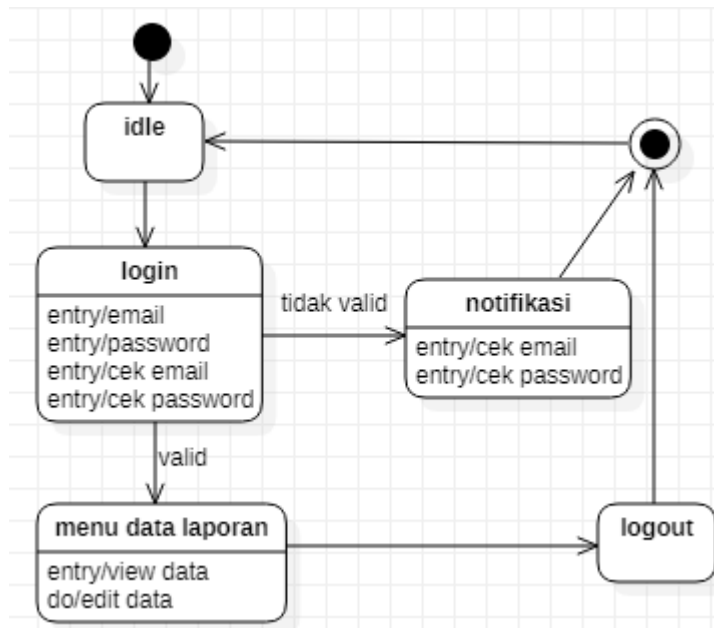


Gambar 5.18 Statechart Diagram Report Activity Harian

Keterangan :

Pada *statechart* diagram ini menjelaskan proses meng-*input* data laporan kegiatan harian yang dilakukan oleh aktor mahasiswa. Aplikasi dalam keadaan *idle* lalu status aplikasi berubah saat mahasiswa melakukan *login*. Apabila *login* valid maka mahasiswa dialihkan ke halaman utama dan apabila *login* tidak valid maka mahasiswa kembali ke halaman login. Setelah mahasiswa berhasil melakukan *login* dan masuk ke halaman utama maka mahasiswa memilih menu *report* harian dan melakukan pengelolaan berupa input, perhitungan dan simpan data *report* harian. Setelah melakukan pengelolaan data *report* harian maka mahasiswa dapat melakukan *logout*.

4. Statechart Diagram Penilaian

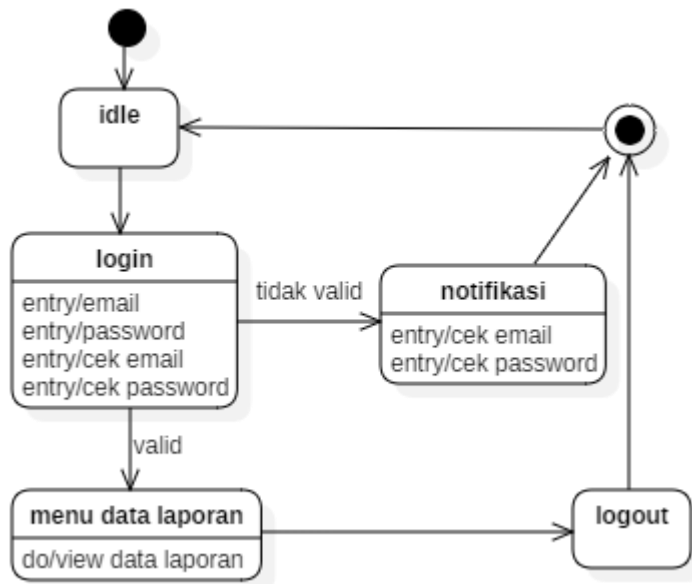


Gambar 5.19 Statechart Diagram Penilaian

Keterangan :

Pada *statechart* diagram ini menjelaskan proses penilaian yang dilakukan oleh aktor pembimbing. Aplikasi dalam keadaan *idle* lalu status aplikasi berubah saat aktor melakukan *login*. Apabila *login* valid maka aktor di alihkan ke halaman utama dan apabila *login* tidak valid maka aktor kembali ke halaman *login*. Setelah aktor berhasil melakukan *login* dan masuk ke halaman utama maka aktor masuk ke menu data laporan serta melakukan pengelolaan berupa tambah, *view* dan *edit* data laporan. Setelah melakukan pengelolaan data laporan maka aktor dapat melakukan *logout*.

5. Statechart Diagram Monitoring

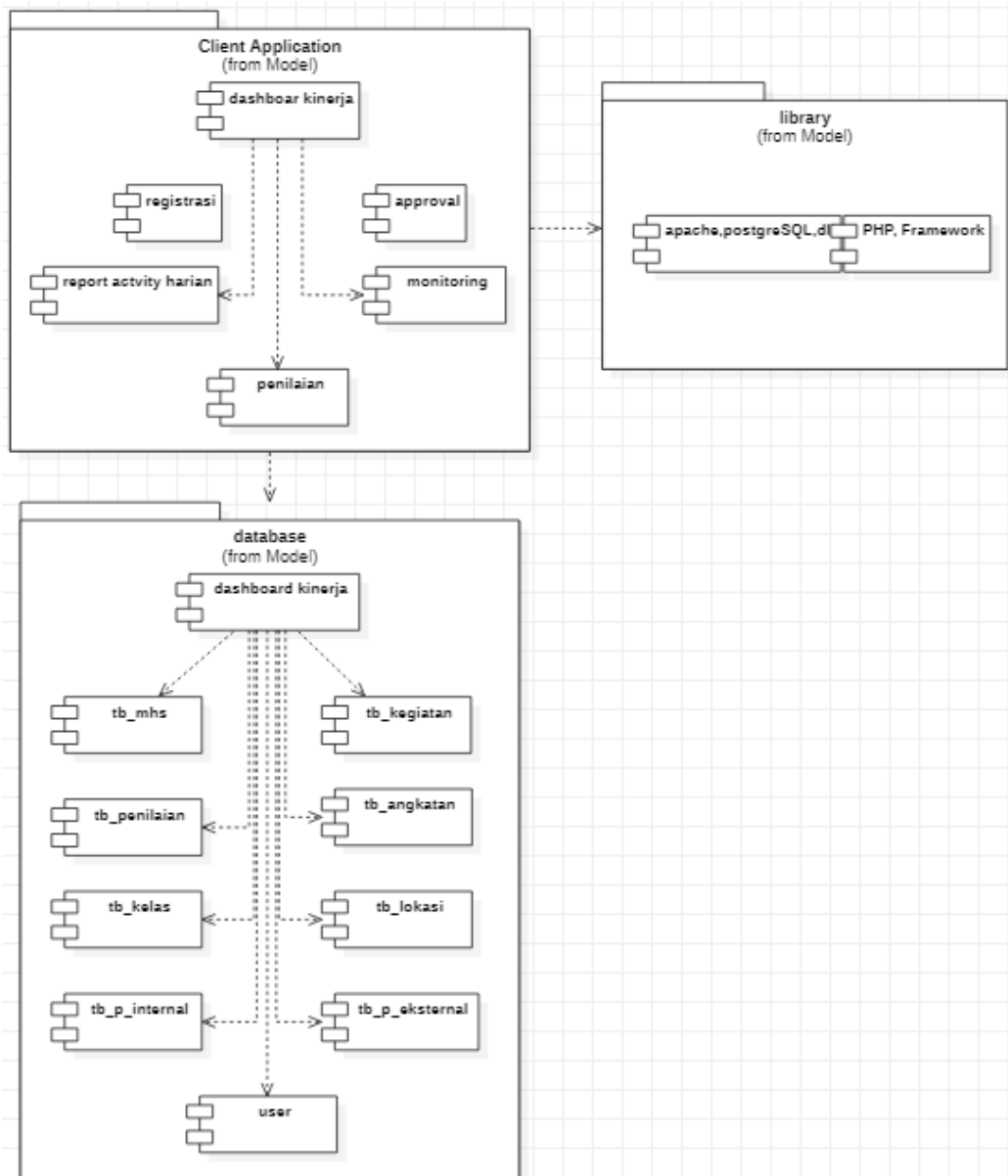


Gambar 5.20 Statechart Diagram Monitoring

Keterangan :

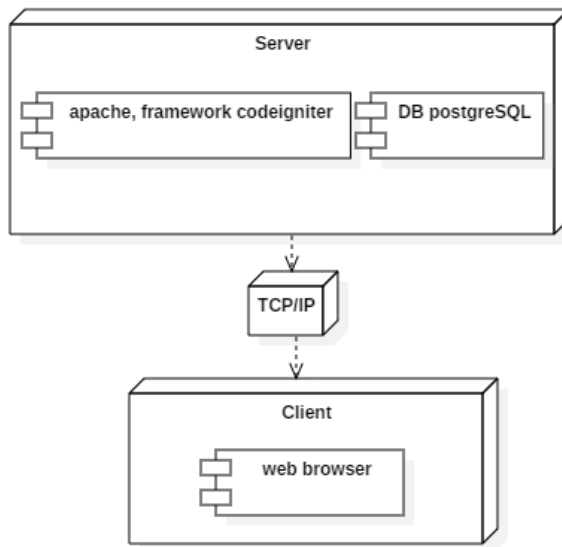
Pada *statechart* diagram ini menjelaskan proses kelola laporan yang dilakukan oleh aktor koordinator. Aplikasi dalam keadaan *idle* lalu status aplikasi berubah saat aktor melakukan *login*. Apabila login valid maka aktor di alihkan ke halaman utama dan apabila *login* tidak valid maka aktor kembali ke halaman *login*. Setelah admin berhasil melakukan *login* dan masuk ke menu data laporan maka aktor dapat melakukan aksi seperti *check-ing* data dengan menekan pada *button* detail dan sistem akan menampilkan data yang lengkap. Setelah melakukan *check-ing* data laporan maka aktor dapat melakukan *logout*.

5.3.5 Component Diagram



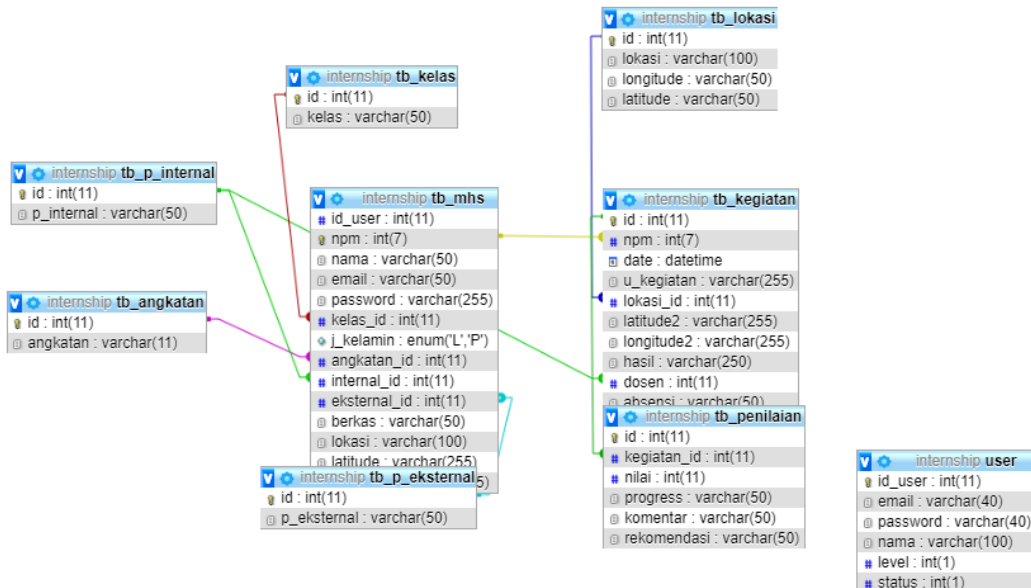
Gambar 5.21 Component Diagram

5.3.6 Deployment Diagram



Gambar 5.22 Deployment Diagram

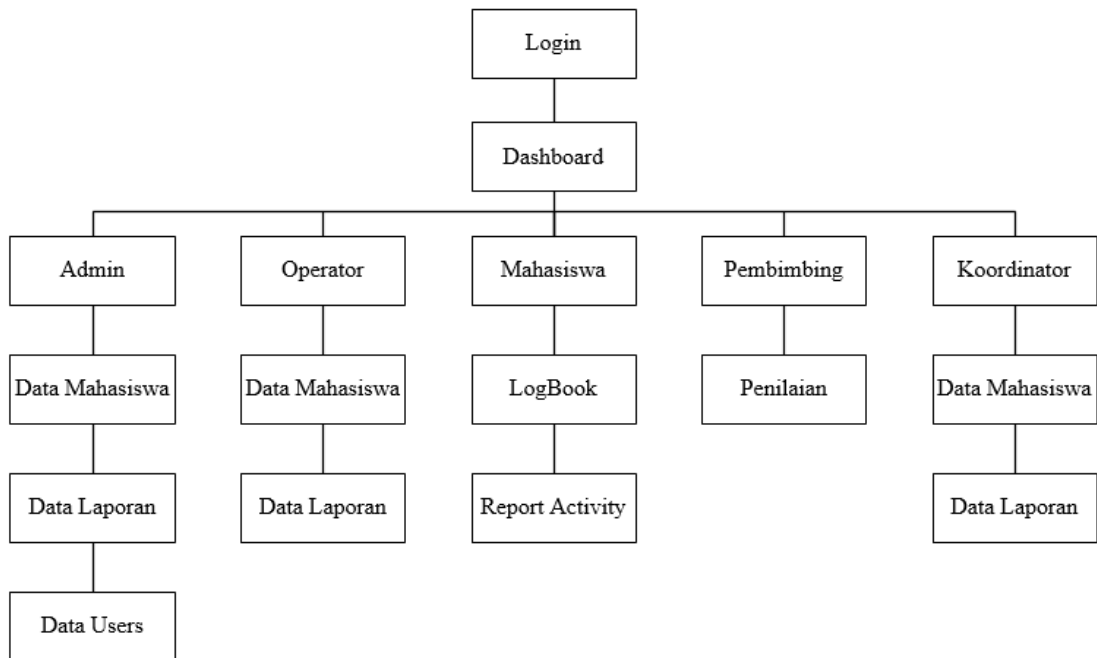
5.3.7 Perancangan Database



Gambar 5.23 Perancangan Database

5.3.8 Struktur Menu


Fungsi-fungsi yang dirancang pada tahap perancangan ini dibagi ke dalam beberapa menu yang bertujuan untuk memudahkan pengoperasian program. Menu yang digunakan pada program saat ini dapat dilihat pada struktur berikut :



Gambar 5.24 Struktur Menu

5.3.9 Perancangan *Interface*

1. Form *Login*



Sign in to start your session

email


password

Register new a membership


Sign in

Gambar 5.25 Login

2. Form Registrasi



REGISTRASI



Nama Mahasiswa

Email

Password

NPM

Program Studi/Kelas

☐ Laki-laki ☐ perempuan

Angkatan

Pembimbing Internal

Pembimbing Eksternal

Nama Lokasi

Latitude

Longitude

Pilih File

Surat Keterangan

Daftar


Reset

Gambar 5.26 Form Registrasi

3. Halaman *Approval* Data Mahasiswa

myINTERNSHIP

admin@gmail.com


Admin

Dashboard

Data Mahasiswa

Data Register Internship

> Mahasiswa

No.	Nama Mahasiswa	NPM	Email	Program Studi	Status	Aksi
1.	Alip Suprpto M	1164063	fulan01@gmail.com	D4 Teknik Informatika 4C	1 - Aktif	<div>DetailEditDelete</div>
2.	Fathi Rabbani	1164074	fulan02@gmail.com	D4 Teknik Informatika 4C	1 - aktif	<div>DetailEditDelete</div>

Gambar 5.27 Halaman Approval Data Mahasiswa

4. Form Report Activity Harian

myINTERNSHIP

mahasiswa@gmail.com

> Mahasiswa

Fulan 01

Dashboard

Report Harian

Report Harian

kegiatan

Uraian Kegiatan

NPM

Posisi kamu disini!

Latitude

Longitude

hasil

Klik hasil

Pilih File

Upload File

Simpan

Reset

Gambar 5.28 Form Report Activity Harian

5. Form Penilaian

myINTERNSHIP

pembimbing@gmail.com

> Mahasiswa

Dosen A

Dashboard

Data Mahasiswa

Data Laporan

Data Laporan

internship

No.	Nama Mahasiswa	Kegiatan	Email	Program Studi	File	Aksi
1.	Aip Suprpto M	Internship 1	fulan01@gmail.com	D4 Teknik Informatika 4C	<div></div> <div>Kegiatan.docx</div>	<div>Edit</div> <div>Hapus</div>
2.	Aip Suprpto M	Internship 1	fulan02@gmail.com	D4 Teknik Informatika 4C	<div></div> <div>Kegiatan.docx</div>	<div>Edit</div> <div>Hapus</div>

Gambar 5.29 Form Penilaian

6. Form Monitoring

myINTERNSHIP koordinator@gmail.com

Koordinator

Dashboard

Data Mahasiswa

Data Laporan

Data Laporan internship > Mahasiswa

No.	Nama Mahasiswa	Kegiatan	Pembimbing	Absensi	File	Aksi
1.	Aip Suprpto M	Internship 1	fulan01@gmail.com	Hadir	Kegiatan.docx	Detail
2.	Aip Suprpto M	Internship 1	fulan02@gmail.com	Hadir	Kegiatan.docx	Detail

Gambar 5.30 Form Monitoring

5.4 Perancangan Arsitektur Perangkat Lunak dan Perangkat Keras Sistem

Dalam perancangan dashboard monitoring kinerja mahasiswa internship membutuhkan beberapa perangkat lunak yaitu sebagai berikut.

5.4.1 Perangkat Lunak

Perangkat lunak pendukung yang digunakan adalah sebagai berikut :

Tabel 5.8 Perangkat Lunak

No.	Jenis	Keterangan
	System Operasi	Microsoft Windows 10 Profesional 64-bit
	Bahasa Pemrograman	PHP dengan Framework CI (Codeigniter)

	Database		MySQL
	Perangkat Lunak		Visual Studio Code Microsoft Visio 2013 UML 3.1.0

Tabel 5.8 Perangkat Lunak

5.4.2 Perangkat Keras

Perangkat keras pendukung yang digunakan adalah sebagai berikut :

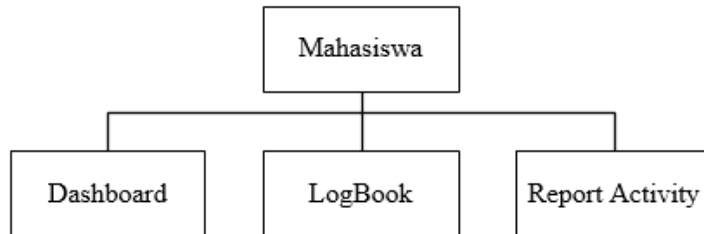
Tabel 5.8 Perangkat Keras

No.	Jenis		Keterangan
	Processor		Intel® core™i3
	Memory		8GB
	Monitor		LED 14,1 Inchi
	Mouse dan keyboard		Standard

Tabel 5.9 Perangkat Keras

5.5 Pemetaan Struktur Diagram User / Aktor Sistem

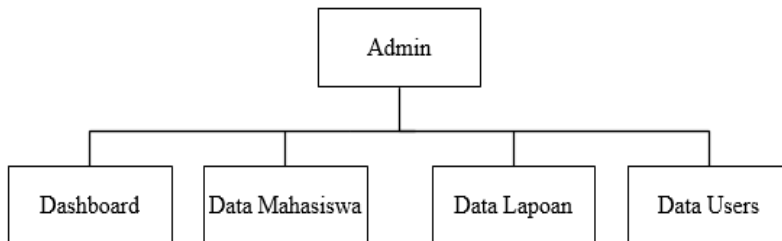
Dibawah ini merupakan pemetaan struktur diagram menu untuk



mahasiswa.

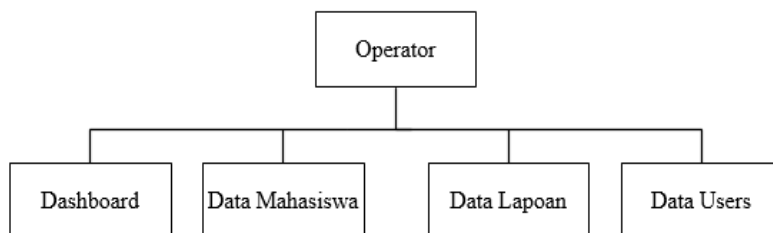
Gambar 5.31 Pemetaan Struktur Diagram Mahasiswa

Dibawah ini merupakan pemetaan struktur diagram menu untuk admin.



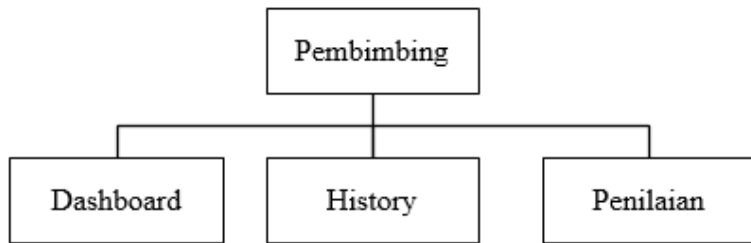
Gambar 5.32 Pemetaan Struktur Diagram Admin

Dibawah ini merupakan pemetaan struktur diagram menu untuk operator.



Gambar 5.33 Pemetaan Struktur Diagram Operator

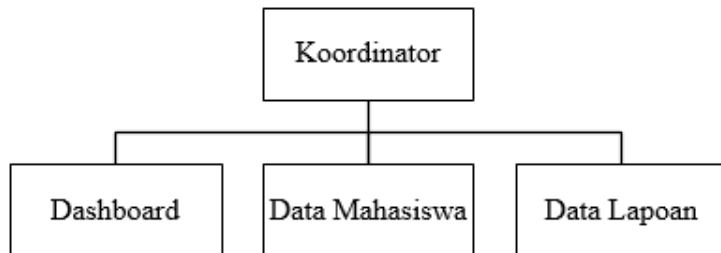
Dibawah ini merupakan pemetaan struktur diagram menu untuk pembimbing.



Gambar 5.34 Pemetaan Struktur Diagram

Pembimbing

Dibawah ini merupakan pemetaan struktur diagram



menu untuk koordinator

Gambar 5.35 Pemetaan Struktur Diagram

Koordinator

BAB VI

6.1 Implementasi Dengan Perhitungan Harvesine Formula

Metode Haversine Formula dapat digunakan untuk menghitung jarak antara dua titik, berdasarkan posisi garis lintang latitude dan posisi garis bujur longitude sebagai variabel inputan. Haversine Formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinat bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2, lat2 [25]. Metode Haversine Formula tersebut kini sudah mengalami pengembangan, yaitu dengan menggunakan rumus *spherical law of cosine* sederhana, dimana dengan penghitungan komputer dapat memberikan tingkat presisi yang sangat akurat antar dua titik. Pertama ditentukan terlebih dahulu titik awal dan titik tuju, titik awal berupa latitude1(lat1) dan longitude1(long1), titik tuju berupa latitude2(lat2) dan longitude2(long2). Titik awal dan titik tuju tersebut berbentuk desimal derajat yang kemudian dirubah menjadi nilai sudut radian, kemudian lakukan perhitungan dengan rumus Haversine Formula, yaitu:

Rumus Harvesine

$$x = (\text{lng2} - \text{lng1}) * \cos ((\text{lat1} + \text{lat2})/2);$$

$$y = (\text{lat2} - \text{lat1});$$

$$d = \text{sqrt}(x*x + y*y) * R$$

Keterangan :

x = longitude (lintang)

y = latitude (bujur)

d = jarak (km)

R = Radius Bumi = 6371 km

1 derajat = 0.0174532925 radian

6.2 Hasil dan Pembahasan

Sistem informasi monitoring kinerja mahasiswa *internship* menerapkan metode formula haversine dan dikembangkan berbasis *website* agar memudahkan pengguna mengakses sistem. Sistem ini menggunakan 8 data jumlah mahasiswa/i yang *internship* di prodi DIV Teknik Informatika Politeknik Pos Indonesia.

Pencarian lokasi pengguna yang memanfaatkan *Global Positioning System Geolocation* dari *GoogleMaps*. Langkah selanjutnya adalah mencari dari titik koordinat mahasiswa registrasi sebagai titik koordinat acuan dan titik koordinat setiap melakukan *report* harian yang telah diinput menggunakan formula haversine, sebagaimana jarak yang didapatkan dari hasil perhitungan kedua titik koordinat tersebut yang ditampilkan oleh sistem dengan visualisasi *alert*.

Google Maps API dimanfaatkan untuk menampilkan peta digital beserta markers dari titik koordinat acuan ke titik koordinat mahasiswa. Dalam perhitungan jarak, sistem menggunakan koordinat *default* yaitu prodi DIV Teknik Informatika, Politeknik Pos Indonesia dan dapat menggunakan koordinat pengguna dengan memanfaatkan fungsi *Geolocation*. Setelah mendapatkan koordinat pengguna, kemudian sistem mulai menghitung jarak menggunakan formula haversine. Berikut contoh analisis cara kerja metode haversine formula dalam perhitungan jarak antara dua titik :

1. Koordinat Patokan (acuan)

lat1, lng1
-6.873776, 107.575639

Gambar 6.1 Koordinat Acuan

Keterangan :

Data diatas diambil dari koordinat data registrasi untuk sebagai titik acuan pada saat menggunakan perhitungan formula harvesine. Penulis mengambil satu sample data lat1, lng1 yaitu -6.873776, 107.575639.

2. Koordinat *Report* Harian

lat2, lng2
-6.873529, 107.576098
-6.873770, 107.576701
-6.873529, 107.576098
-6.873333, 107.576141
-6.873399, 107.575530
-6.873399, 107.575530
-6.873399, 107.575530
-6.873383, 107.575380

Gambar 6.2 Report Harian

Keterangan :

Data diatas mengambil dari data koordinat kegiatan pada saat mahasiswa melakauk *report* harian. Selanjutnya penulis akan mem-praktekan salah satu data diatas, contoh data lat2, lng2 yang digaris merahkan (-6.873529, 107.576098).

Percobaan 1

1. Titik koordinat pertama

(Prodi DIV Teknik Informatika, Politeknik Pos Indonesia)

$$\begin{aligned}\text{Lat 1} &= -6.873776 \times 0.0174532925 \\ &= -0.1199700231 \text{ Radian}\end{aligned}$$

$$\begin{aligned}\text{Lng 1} &= 107.575639 \times 0.0174532925 \\ &= 1.87754909334 \text{ Radian}\end{aligned}$$

2. Titik koordinat kedua

(mahasiswa *report* harian.Auditorium)

$$\begin{aligned}\text{Lat 2} &= -6.873529 \times 0.0174532925 \\ &= -0.11996571214 \text{ Radian}\end{aligned}$$

$$\begin{aligned}\text{Lng 2} &= 107.576098 \times 0.0174532925 \\ &= 1.8775571044 \text{ Radian}\end{aligned}$$

$$\begin{aligned}3. \quad x &= (\text{lng2}-\text{lng1}) * \cos ((\text{lat1}+\text{lat2})/2) \\ &= (1.8775571044-1.87754909334) * \cos ((-0.1199700231 + -0.11996571214)/2) \\ &= 0.00000795348\end{aligned}$$

$$\begin{aligned}4. \quad y &= (\text{lat2}-\text{lat1}) \\ &= (-0.11996571214-(-0.1199700231)) \\ &= 0.00000431096 \text{ d} \\ &= \text{sqrt} (x*x+y*y)*R \\ &= \text{sqrt} ((0.00000795348 \times 0.00000795348)+(0.00000431096 \times 0.00000431096)) \times 6371 \\ &= \text{sqrt} (0.00000000008184222) \times 6371 \\ &= 0.00000904666900024534 \times 6371\end{aligned}$$

= 0.0576363282 KM to Meter
= 0.0576363282 x 1000
= 57.6363282006 Meter.

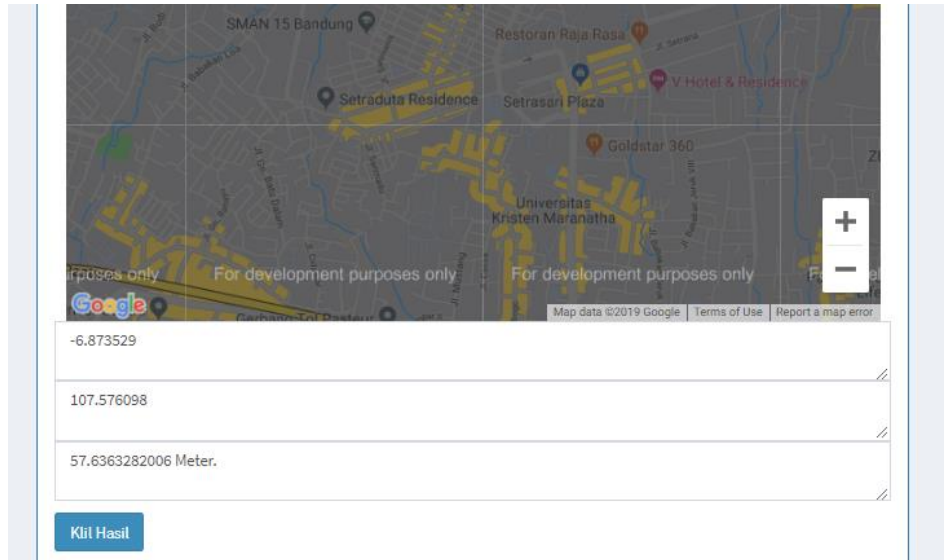
5. Implementasi *source code* dan di jalankan di *browser*

```
function haversine($id)
{
    $explodeVal = explode('~', $id);
    // dari prodi
    $lat1 = -6.873776;
    $lon1 = 107.575639;
    // target
    $lat2 = $explodeVal[0];
    $lon2 = $explodeVal[1];
    // distance between latitudes
    // and longitudes
    $dLat = ($lat2 - $lat1) *
    M_PI / 180.0;
    $dLon = ($lon2 - $lon1) *
    M_PI / 180.0;

    // convert to radians
    $lat1 = ($lat1) * M_PI / 180.0;
    $lat2 = ($lat2) * M_PI / 180.0;

    // apply formulae
    $a = pow(sin($dLat / 2), 2) +
    pow(sin($dLon / 2), 2) *
    cos($lat1) * cos($lat2);
    $rad = 6371;
    $c = 2 * asin(sqrt($a));
    $SimpanRad = $rad * $c;
    // Driver code
    $penjumlahan = $SimpanRad*1000;
    $data['jsonarray'] = $penjumlahan.' METER';
    echo json_encode($data);
}
```

Gambar 6.3 Implementasi Source Code



Gambar 6.4 implementasi browser

Percobaan 2

6. Titik koordinat pertama

(kamar 1)

$$\text{Lat 1} = -6.8741201 \times 0.0174532925$$

$$= -0.11997602878 \text{ Radian}$$

$$\text{Lng 1} = 107.57703269999999 \times 0.0174532925$$

$$= 1.877573418 \text{ Radian}$$

7. Titik koordinat kedua

(kamar 5)

$$\text{Lat 2} = -6.8741534 \times 0.0174532925$$

$$= -0.11997660998 \text{ Radian}$$

$$\text{Lng 2} = 107.5770213 \times 0.0174532925$$

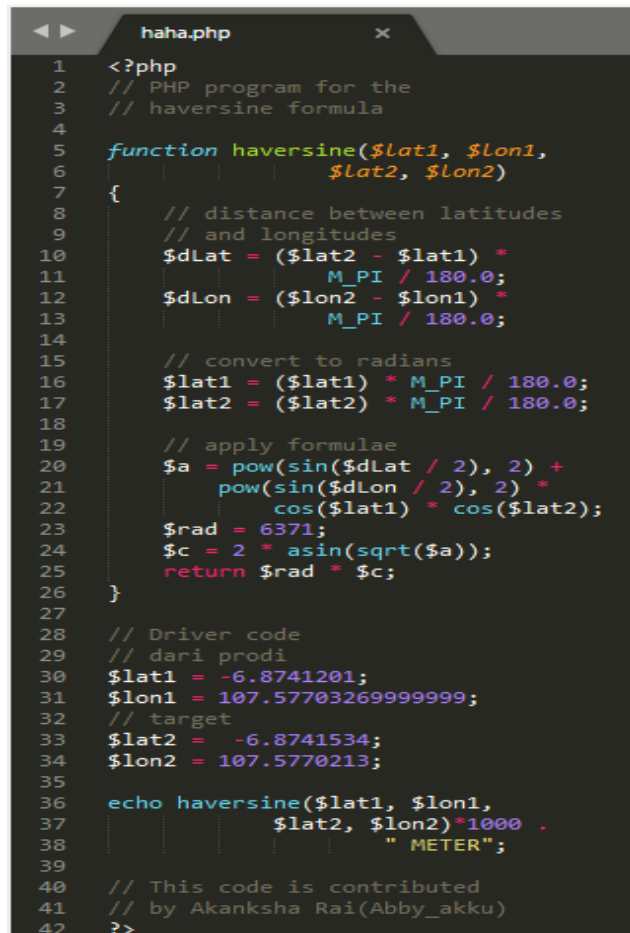
$$= 1.87757321903 \text{ Radian}$$

$$\begin{aligned} 8. \quad x &= (\text{lng2} - \text{lng1}) * \cos ((\text{lat1} + \text{lat2})/2) \\ &= (1.87757321903 - 1.877573418) * \cos ((-0.11997602878 + -0.11997660998)/2) \end{aligned}$$

$$= 0.00000795348$$

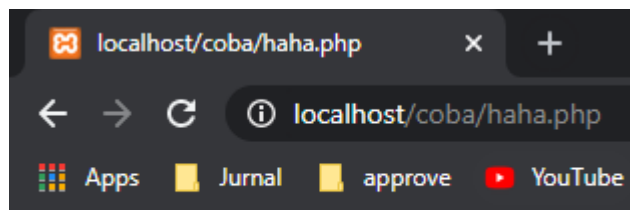
$$\begin{aligned}
 9. \quad y &= (\text{lat2-lat1}) \\
 &= (-0.11997660998 - (-0.11997602878)) \\
 &= 0.000006242096 \text{ d} \\
 &= \sqrt{x^2 + y^2} * R \\
 &= \sqrt{((0.000002395348^2 + 0.00000435348^2) + (0.000003131096^2 + 0.00000431096^2))} * 6371 \\
 &= \sqrt{0.000000000081876422} * 6371 \\
 &= 0.00006897975669024534 * 6371 \\
 &= 0.039108034268386 \text{ KM to Meter} \\
 &= 0.03910803488 * 1000 \\
 &= 3.91080348 \text{ Meter.}
 \end{aligned}$$

10. Implementasi *source code* dan dijalankan dibrowser



```
1 <?php
2 // PHP program for the
3 // haversine formula
4
5 function haversine($lat1, $lon1,
6                   $lat2, $lon2)
7 {
8     // distance between latitudes
9     // and longitudes
10    $dLat = ($lat2 - $lat1) *
11            M_PI / 180.0;
12    $dLon = ($lon2 - $lon1) *
13            M_PI / 180.0;
14
15    // convert to radians
16    $lat1 = ($lat1) * M_PI / 180.0;
17    $lat2 = ($lat2) * M_PI / 180.0;
18
19    // apply formulae
20    $a = pow(sin($dLat / 2), 2) +
21          pow(sin($dLon / 2), 2) *
22          cos($lat1) * cos($lat2);
23    $rad = 6371;
24    $c = 2 * asin(sqrt($a));
25    return $rad * $c;
26 }
27
28 // Driver code
29 // dari prodi
30 $lat1 = -6.8741201;
31 $lon1 = 107.57703269999999;
32 // target
33 $lat2 = -6.8741534;
34 $lon2 = 107.5770213;
35
36 echo haversine($lat1, $lon1,
37               $lat2, $lon2)*1000 .
38        " METER";
39
40 // This code is contributed
41 // by Akanksha Rai(Abby_akku)
42 ?>
```

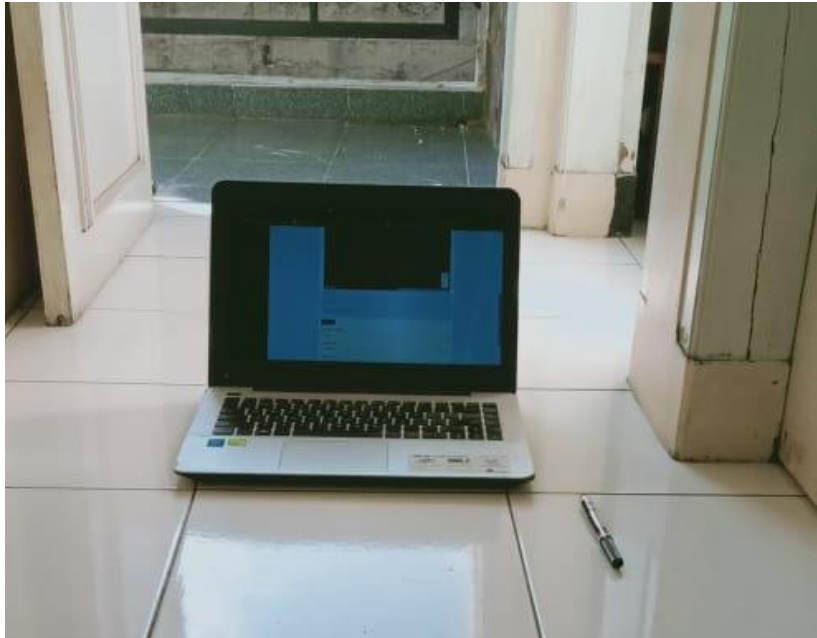
Gambar 6.5 implementasi source code 2



3.9108194268349 METER

Gambar 6.6 Hasil Pada Browser

11. Perhitungan manual dengan alat meter gulung



Gambar 6.7 Koordinat latitude, longitude 1



Gambar 6.8 Koordinat latitude, longitude 1



Gambar 6.9 Koordinat latitude, longitude 2



Gambar 6.10 Koordinat latitude, longitude 2



Gambar 6.11 Hasil Jarak

Hasil akhir dari koordinat latitude, longitude 1 dan latitude, longitude 2 adalah sebesar 391 cm, apabila di rubah ke meter menjadi 3,91 Meter.

12. Data koordinat hasil dari pengujian sistem

Data-data koordinat hasil perhitungan metode harvesine formula dengan sistem yang dibangun dan nama lokasi yang diambil oleh peneliti adalah parkir belakang gedung pendidikan :

	Perpindahan	Arah Mata Angin				
No.	Jarak	Utara	Selatan	Barat	Timur	Range
	0					
1.	2 Meter	0	13 Meter	14 Meter	14 Meter	14 Meter
2.	4 Meter	3 Meter	13 Meter	6 Meter	6 Meter	10 Meter
3.	6 Meter	13 Meter	15 Meter	6 Meter	114 Meter	108 Meter
4.	8 Meter	13 Meter	8 Meter	8 Meter	6 Meter	7 Meter
5.	10 Meter	15 Meter	11 Meter	12 Meter	12 Meter	4 Meter
6.	12 Meter	66 Meter	8 Meter	14 Meter	14 Meter	58 Meter
7.	14 Meter	11 Meter	14 Meter	13 Meter	13 Meter	3 Meter
8.	16 Meter	14 Meter	10 Meter	19 Meter	18 Meter	9 Meter
9.	18 Meter	14 Meter	71 Meter	8 Meter	7 Meter	64 Meter
10.	20 Meter	14 Meter	57 Meter	59 Meter	3 Meter	56 Meter
	Rata-rata					33,3

Gambar 6.12 Hasil Pengujian Sistem

Keterangan :

Pada gambar 6.12 adalah hasil pengujian dari sistem yang dilakukan oleh peneliti dengan mengambil lokasi di parkir belakang gedung pendidikan serta melakukan pergerakan dari 0 sampai 20 meter yang menghasilkan rata-rata 33,3 dari penjumlahan *range* setiap mata arah angin. Hasil dari data diatas maka peneliti menambahkan rumus angka pada sistem (33,3) agar perhitungan jarak lebih mendekati akurat.

```
// Driver code
// dari acuan
$lat1 = -6.8734373;
$lon1 = 107.57479509999999;
// target
$lat2 = -6.8734351;
$lon2 = 107.5754055;

echo haversine($lat1, $lon1,
               $lat2, $lon2)*33,3 .
               "METER";

?>
```

Gambar 6.13 Source Code Sistem

Apabila dilakukan perhitungan maka hasil dari sistem yang dibangun adalah sebagai berikut :

	Perpindahan	Arah Mata Angin			
No.	Jarak	Utara	Selatan	Barat	Timur
	0				
1.	2 Meter	0	4 Meter	4 Meter	4 Meter
2.	4 Meter	1 Meter	4 Meter	2 Meter	2 Meter
3.	6 Meter	4 Meter	5 Meter	2 Meter	38 Meter
4.	8 Meter	5 Meter	2 Meter	2 Meter	2 Meter
5.	10 Meter	5 Meter	3 Meter	4 Meter	4 Meter
6.	12 Meter	2 Meter	3 Meter	4 Meter	4 Meter
7.	14 Meter	3 Meter	4 Meter	4 Meter	4 Meter
8.	16 Meter	3 Meter	4 Meter	6 Meter	6 Meter
9.	18 Meter	4 Meter	23 Meter	2 Meter	2 Meter
10.	20 Meter	4 Meter	19 Meter	19 Meter	1 Meter

Gambar 6.14 Hasil Pengujian Sistem

6.3 Evaluasi Metode

Hasil dari perhitungan yang dilakukan oleh peneliti secara manual serta diperbandingkan dengan perhitungan sistem hasil yang diperoleh sama, jadi untuk diterapkan pada sistem yang dibangun sangat membantu untuk menghitung jarak mahasiswa/i pada pergerakan saat melakukan laporan kegiatan.

