

**PANDUAN LENGKAP**  
**MEMBANGUN SISTEM MONITORING KINERJA**  
**MAHASISWA INTERNSHIP BERBASIS WEB DAN**  
**GLOBAL POSITIONING SYSTEM**

**PANDUAN LENGKAP MEMBANGUN SISTEM  
MONITORING KINERJA MAHASISWA INTERNSHIP  
BERBASIS WEB DAN 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 membangun sistem monitoring kinerja mahasiswa *internship* berbasis *web* dan *global positioning system*.

## ***ACKNOWLEDGMENTS***

## ***ACRONYMS***

## ***SYMBOLS***

## ***INTRODUCTION***

## **BAB I**

### **INTERNSHIP**

#### **1.1 Pengertian Magang**

*Training* merupakan suatu istilah yang memiliki konotasi tertentu tergantung pada pengalaman seseorang dan latar belakangnya. Definisi *training* menurut Robinson adalah pengajaran atau pemberian pengalaman kepada seseorang untuk mengembangkan tingkah laku (pengetahuan, skill, sikap) agar mencapai sesuatu yang diinginkan. Dalam *dictionary of Education*, pelatihan (*training*) diartikan sebagai suatu pengajaran tertentu yang tujuannya telah ditentukan secara jelas dan dapat diragakan, yang menghendaki peserta dan penilaian terhadap perbaikan unjuk kerja peserta didik (Marzuki, 2010: 174-175).

Menurut pendapat *Good*, *training* juga diartikan sebagai pengetahuan. Definisi yang terakhir ini jika dikaitkan dengan andragogi, *training* umumnya ditujukan kepada orang dewasa, sesuai sekali karena andragogi adalah seni/ilmu membantu orang lain dalam belajar (Brundage). Ini merupakan konsep *training* yang luas dan luwes karena dengan konsep ini terwadahi segala macam kegiatan *training* seperti *coaching*, *on the job training*, magang dan *job rotation* (Marzuki, 2010: 175).

Masalah magang telah diatur dalam Undang-undang No. 13 tahun 2003 tentang Ketenagakerjaan khususnya pasal 21-30. Dan lebih spesifiknya diatur dalam Peraturan menteri Tenaga



Kerja dan transmigrasi no.Per22/Men/IX/2009 tentang Penyelenggaraan Permagangan di Dalam Negeri.

Dalam peraturan Menteri tersebut, permagangan diartikan sebagai bagian dari sistem pelatihan kerja yang diselenggarakan secara terpadu antara pelatihan di lembaga pelatihan dengan bekerja secara langsung di bawah bimbingan dan pengawasan instruktur atau pekerja yang lebih berpengalaman dalam proses produksi barang dan/atau jasa di perusahaan dalam rangka menguasai keterampilan atau keahlian.

Menurut *Greene*, magang memiliki pengertian seorang pemula yang mempelajari suatu keahlian. Magang regular yang disponsori oleh Departemen Tenaga Kerja, bekerja sama dengan pemerintah daerah dan berbagai agen kerja terdekat, biasanya mengharuskan peserta yang berminat untuk mengikuti magang ialah mereka yang usianya diatas 18 tahun namun ada juga magang untuk pemula yang disponsori juga oleh Departemen Tenaga Kerja dengan syarat yang berbeda sesuai dengan hukumperlindungan anak yaitu mereka yang telah duduk di kelas 11 sekolah menengah (*Greene 2006: 155*).

Magang menawarkan kesempatan untuk mencoba sesuatu di luar pakem, dan biasanya diakhir program magang, peserta didik akan mendapatkan sertifikat Kecakapan Bekerja yang menunjukan bahwa peserta didik ikut serta dalam program magang serta telah mendapatkan keahlian tertentu.

Program pembelajaran magang atau biasa disebut dengan *apprenticeship, learning by doing, on-the-job-training/off-the-job-training* dan *built in learning*, dimana program ini dirancang untuk level keahlian yang lebih tinggi. Oleh karenanya program pembelajaran magang (*learning by doing*) cenderung lebih mengarah pada pendidikan pada pelatihan dalam hal pengetahuan dan dalam melakukan suatu keahlian atau suatu rangkaian pekerjaan yang saling berhubungan. Program pembelajaran magang adalah menggabungkan pelatihan dan pengalaman pada pekerjaan dengan instruksi yang didapatkan di dalam tempat tertentu untuk subjek-subjek tertentu. Magang juga mirip dengan *internship*, namun demikian *internship* bersifat sementara. *Internship* dilakukan biasanya untuk pelajar atau mahasiswa dengan waktu dan program yang sangat terbatas. Program yang dikembangkan dalam *internship* bisa sama dengan magang dimana memberikan individu-individu dengan pengalaman pada pekerjaan tertentu, atau pengenalan terhadap pekerjaan, organisasi, atau industri (Kamil, 2002: 10).

Magang merupakan suatu proses pembelajaran yang mengandung unsur belajar sambil bekerja. Biasanya warga belajar sebagai pemegang akan membiasakan diri mengikuti proses pekerjaan yang dilakukan oleh pendidik atau orang yang bekerja disitu.

Pembelajaran magang dikatakan berhasil apabila permagang (pendidik) mampu menyadarkan pemagang (peserta

didik) untuk melakukan kegiatan belajar secara mandiri. Sudjana , D (2000) mengemukakan lebih lanjut bahwa melalui magang seseorang yang memiliki pengalaman tertentu menyampaikan pengetahuan dan keterampilan yang telah ia miliki kepada orang lain yang belum berpengalaman dan yang lebih dahulu memiliki pengalaman dan keahlian tertentu sehingga pemagang memiliki keahlian atau pengalaman itu kemudian setelah terjadi penerimaan pengalaman atau keahlian, pemagang mampu melakukannya sendiri.

Pada konteks menurut Dirjen Diklusepora magang memiliki pengertian sebagai suatu proses belajar dimana seseorang memperoleh dan menguasai keterampilan dengan jalan melibatkan diri dalam proses pekerjaan tanpa atau dengan petunjuk orang yang sudah terampil dalam pekerjaannya (Kamil,2002: 11).

Kajian teoritis tersebut menggambarkan bahwa program pembelajaran magang adalah suatu pembelajaran dimana dalam proses pembelajarannya pemagang (peserta didik) turut membantu langsung dalam pekerjaan pemagang (pendidik) yang mana diharapkan pemagang (peserta didik) dapat memiliki keterampilan dan perubahan dalam pengetahuan dan sikap selama menjalani pekerjaan tersebut dan yang pada akhirnya pemagang (peserta didik) dapat melakukannya sendiri setelah proses pembelajaran magang selesai.

Inti dari proses pembelajaran magang sendiri adalah adanya interaksi edukatif melalui belajar sambil bekerja atau

bekerja sambil belajar dimana unsur peniruan memegang peranan penting dalam keberhasilan program pembelajaran magang.

## **1.2 Program *Internship* Mahasiswa**

Program *internship* mahasiswa dapat dijadikan salah satu alternatif bagi mahasiswa mengasah kemampuan sebelum benar-benar terjun ke dunia kerja. Program *internship* bisa menjadi salah satu ajang bagi mahasiswa untuk mendapatkan pemasukan tambahan. Terdapat penyelenggara program *internship* yang akan memberikan gaji pada mahasiswanya, namun tidak jarang pula mahasiswa tidak mendapatkan upah karena termasuk program *volunteer*.

Program *internship* sebenarnya tidak terlalu jauh berbeda dengan program magang. *Internship* juga diperlukan bagi seorang mahasiswa untuk bisa mendapatkan banyak pengalaman sebelum terjun langsung ke dunia kerja. Ketika belajar di bangku kuliah, mahasiswa belum mendapatkan jaminan pengalaman terjun ke dunia kerja, sehingga program *internship* sangat diperlukan untuk mengatasi kesenjangan tersebut.

Mahasiswa atau pekerja yang memiliki status *internship* disebut dengan *intern*. Meskipun demikian, ruang kerja atau ranah kerja seorang *intern* dibatasi dan tidak disejajarkan dengan karyawan biasa. Sebenarnya program *internship* tidak hanya membawa keuntungan sebagai sarana untuk mendapatkan pengalaman tambahan saja, tetapi juga keuntungan bagi pihak pemberi lowongan *internship*. Perusahaan bisa memperoleh tenaga kerja yang kompeten di bidangnya. Hasil magang atau program

*internship* tidak hanya berwujud untuk menambah pengalaman saja. Kamu juga dapat menjadikan sertifikat hasil magang untuk dicantumkan pada *curriculum vitae*, sehingga pihak yang membaca CV milikmu bisa mempertimbangkan hal tersebut sebagai alasan untuk menerimamu. Hal tersebut dikarenakan saat ini banyak pemberi lowongan kerja yang mensyaratkan kandidat atau pelamarnya sudah memiliki pengalaman bekerja dalam waktu tertentu. Pengalaman kerja sangat diperlukan, sehingga adanya program *internship* mampu menjadi salah satu jembatan untuk mengatasi kesenjangan tersebut.

### **1.2.1 Hak dan Kewajiban**

Selama masa *internship* peserta mempunyai hak dan kewajiban yang harus dipatuhi, antara lain :

#### **A. Hak Peserta Internship Selama Internship :**

1. Peserta dapat diberikan uang saku dan tunjangan lain yang berlaku pada perusahaan yang bertalian. (Sesuai dengan kemampuan Perusahaan yang bertalian).
2. Memperoleh keterangan yang berkaitan dengan :
  - a. Ruang lingkup program *internship*.
  - b. Profil perusahaan.
  - c. Peraturan dan tata tertib perusahaan yang berlaku bagi mahasiswa *internship*.
  - d. Perlengkapan kerja (bila ada dan diharuskan) sesuai dengan peraturan perusahaan.

3. Monitor dan bimbingan selama masa *internship* oleh seorang mentor yang ditunjuk oleh manajemen perusahaan dimana peserta *internship*.

B. Kewajiban Peserta Selama *Internship* :

1. Mentaati semua peraturan yang berlaku diperusahaan tersebut.
2. Mengerjakan tugas-tugas yang telah disampaikan oleh mentor.
3. Mengikuti *review* yang diselenggarakan oleh manajemen perusahaan penerima *internship*, berupa :
  - a. *review activity plan*, peserta mempresentasikan *activity plan* yang telah disepakati bersama mentor.
  - b. *Progress Review*, peserta mempresentasikan laporan aktivitas yang telah dijalankan kepada pihak manajemen perusahaan dan manajemen Politeknik Pos Indonesia di tempat yang disepakati oleh perusahaan tempat diman peserta *internship* berada.
  - c. *Final report*, peserta membuat laporan tugas selama periode tertentu sesuai dengan yang diminta oleh perusahaan, misalnya disesuaikan dengan *activity plan* yang ditandatangani oleh mentor.

### C. Peran dan Tanggung Jawab Peserta

Peran dan Tanggung jawab peserta antara lain :

1. Melakukan tugas-tugas yang menjadi tanggung jawabnya.
2. Mempelajari hal-hal teknis di unit kerjanya dalam rangka memahami kompetensi teknis sesuai dengan ruang lingkup kerjanya.
3. Menerima *feed back* secara konstruktif.
4. Membuat laporan dan mempresentasikan temuan masalah yang dihadapi berikut dengan pemecahan masalahnya atau perbaikan (bila ada).

#### 1.3 Tujuan dan Ciri Magang

Tujuan adanya proses pembelajaran magang adalah untuk memiliki pengetahuan, pengalaman atau keahlian dan perubahan sikap mental pada bidang pekerjaan tertentu yang diharapkan setelah mengikuti proses pembelajaran magang, peserta dapat melakukannya sendiri sebagai bukti bahwa ia memiliki kemampuan mengerjakan sesuatu relative sama dengan permagangnya.

Latihan kerja merupakan suatu jalur yang harus dilalui oleh setiap tenaga kerja, baik yang sudah bekerja maupun yang siap bekerja, baik dalam bentuk klasikal maupun dalam bentuk permagangan, bekerja sambil berlatih atau belajar agar siap untuk dipakai bekerja.

Melalui suatu proses pengalaman kerja atau magang, diharapkan seseorang tidak hanya memiliki kemampuan

mengembangkan diri dengan memiliki keterampilan teknis, tetapi juga mampu mengembangkan diri untuk memiliki kemampuan managerial, *managerial know how* untuk dapat memimpin atau mengatur orang lain (Soeharsono, 1989: 61).

Kajian teoritis mengenai magang menggambarkan bahwa program pembelajaran magang diasumsikan memberi pengaruh kuat terhadap sikap kemandirian dan penghasilan kerja. Asumsi tersebut didukung oleh tujuan magang seperti diuraikan di bawah ini (Kamil, 2002: 12) :

1. Untuk memantapkan penguasaan keterampilan yang diinginkan dan ditekuni untuk dijadikan mata pencaharian.
2. Memperluas dan mempercepat jangkauan pengadaan tenaga-tenagan terampil yang cukup mampu untuk segera berpartisipasi dalam proses pembangunan.

Mengacu pada tujuan tersebut, magang dalam arti proses memiliki ciri-ciri sebagai berikut :

1. Proses magang adalah permagangan dan pemagang (sumber magang atau orang yang dimagangi) berada dalam tempat permagang bekerja. Pemagang melihat dan mencoba menggunakan alat yang diperlukan sehingga tahu, bisa dan terbiasa bagaimana mempergunakannya, bagaimana memperbaikinya kalau rusak, bagaimana merawatnya, dimana menyimpannya, dimana dibelinya serat dibuatnya.



2. Proses magang adalah para pemegang sebaiknya bekerja dan belajar, belajar-bekerja sesuai dengan urutan pekerjaan yang dikerjakan pemegang. Pemegang dapat memulai belajar-bekerja dan bekerja belajar dari mana saja, dari awal, di tengah atau diujung proses pekerjaan itu.
3. Bahwa pemegang belajar-bekerja dan bekerja-belajar tidak diawali teori, melainkan langsung praktek, langsung bekerja.
4. Dilihat dari sudut sumber magang (permagang). Sumber magang tidak perlu orang yang mengetahui teori. Sumber magang atau permagang adalah orang yang pintar dan biasa melaksanakan pekerjaan yang dimagangi.
5. Dilihat dari sudut pandang pemegang, pemegang bukan hanya memperoleh pengetahuan, keterampilan, kemahiran dan sikap mental saja, melainkan dapat terampil melaksanakan pekerjaan (Dirjen Diklusepora, 1990:5).

Tujuan magang disini yaitu memberikan pendidikan dan keterampilan kepada pemegang sambil bekerja dengan waktu yang tidak terbatas. Disini keterampilan diberikan oleh permagang (orang yang memperkerjakan) bersamaan dengan bekerja (sambil bekerja). Sedangkan media yang digunakan dalam pembelajaran yaitu barang yang akan di buat kerja oleh pemegang.

#### **1.4 Manfaat Magang**

Magang merupakan syarat utama untuk melalui proses pendidikan. Magang merupakan bagian dari pelatihan kerja, biasanya magang dilakukan oleh mahasiswa tingkat akhir atau

siswa kelas 3 SMK sebagai salah satu syarat utama untuk menyelesaikan proses pendidikan. Sedangkan pelatihan kerja biasanya diikuti oleh pekerja yang sudah menandatangani kontrak dengan perusahaan dalam rangka untuk mengembangkan kompetensi kerja dan produktifitas sang karyawan.

Kegiatan magang dapat memiliki kesempatan untuk mengaplikasikan semua ilmu yang telah dipelajari di bangku kuliah dan mempelajari detail tentang seluk beluk standar kerja yang profesional. Pengalaman ini kemudian menjadi bekal dalam menjalani jenjang karir yang sesungguhnya. Kegiatan magang juga dapat menambah wawasan mengenai dunia industri dan perkantoran juga meningkatkan keterampilan serta keahlian praktik kerja.

Pemagangan menjadi peran yang penting karena di dalam pemagangan terdapat tujuan yaitu mempromosikan formasi pembelajaran dan keterampilan, serta memfasilitasi tenaga kerja dengan menjembatani antara dunia pendidikan dan dunia kerja juga pemagangan dapat membantu perusahaan dalam pemenuhan kebutuhan tenaga kerja dan menyediakan pelatihan keterampilan bagi kaum muda untuk mempersiapkan mereka dalam menghadapi dunia kerja. Manfaat yang didapatkan dalam pelaksanaan pemagangan juga tidak hanya dirasakan pihak perusahaan yang dapat menghasilkan tenaga kerja sesuai dengan standar industri dan kebutuhan perusahaan, namun juga pihak dari pemangang itu sendiri mendapatkan kesempatan untuk menerima pelatihan,

bukan hanya untuk mengasah keterampilan yang sesuai dengan standar industri/perusahaan, namun juga untuk mendapatkan secara langsung pelatihan secara teknis dan keterampilan kerja inti yang dapat meningkatkan kinerja mereka.

Selain menguasai keterampilan teknis, manfaat pemagangan juga membentuk keterampilan non-teknis (*soft-skill*) peserta pemagangan. Dan menumbuhkan suasana kerja yang mendorong terciptanya inovasi dari peserta magang atau pekerja di perusahaan yang bersangkutan.

Berikut merupakan beberapa hal yang perlu dilakukan dalam pelaksanaan program pemagangan di perusahaan :

1. Pemahaman peraturan perundang-undangan tentang pemagangan.
2. Kebutuhan perusahaan akan tenaga kerja yang memenuhi kualifikasi.
3. Menyusun program pemagangan.
4. Kesepakatan antara perusahaan dengan pemagang yang dituangkan dalam perjanjian pemagangan.
5. Berkoordinasi dengan pihak pemerintah yang membidangi ketenagakerjaan.
6. Memanfaatkan sumber pengetahuan dan informasi yang ada, diantaranya di dapat dari forum pemagangan lainnya.

Pemagangan terdapat beberapa kelebihan dan juga kelemahan, kelebihan magang antara lain :

1. Biaya murah, ditinjau dari segi pembiayaan, magang merupakan cara melatih dengan biaya yang sangat murah bahkan mungkin tanpa biaya. Peserta magang yang mengikuti program pemagangan ini mau tidak dibayar atau dibayar sangat rendah karena tujuan utamanya untuk belajar.
2. Memerlukan manajemen sederhana, dari segi pengelolaan, magang menggunakan manajemen sederhana sehingga sangat membantu dan tidak merepotkan pengelola.
3. Lebih matang, para peserta melalui pengalaman magang ini akan lebih matang dalam menjalankan tugasnya. Hal ini disebabkan mereka langsung menghadapi pekerjaan yang ditangani sehingga lebih dapat menghayati dan menekuni pekerjaan tersebut.
4. Loyalitas, bila perusahaan pada akhirnya ingin menggunakan peserta sebagai karyawan tetap perusahaan, para peserta akan memiliki loyalitas yang tinggi karena sudah banyak mengenal lebih banyak perusahaan tempat mereka magang tersebut.

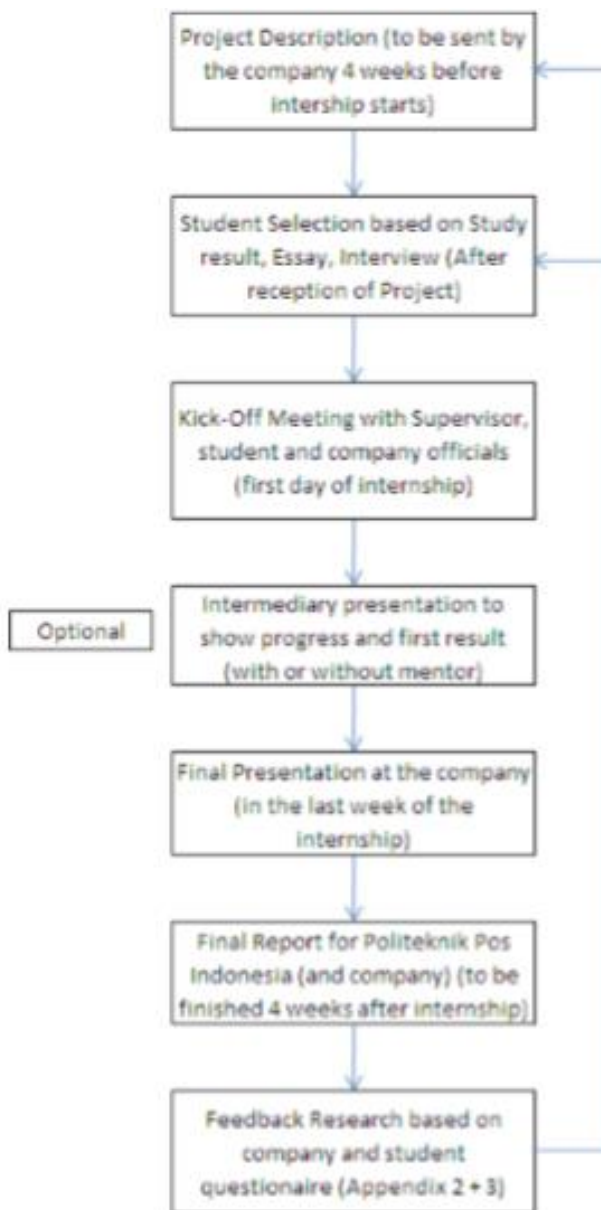
Kelemahan magang sebagai berikut :

1. Terlalu lambat, untuk menjadi ahli melalui proses magang memerlukan waktu cukup lama apalagi bila peserta magang ingin segera memperoleh pekerjaan yang diinginkan dengan segera.

2. Statis dan pengaruh lingkungan, tuntutan zaman yang lebih cepat menuntut para peserta magang untuk mengikuti perkembangan zaman. Bila dalam mengikuti kegiatan magang aspek lingkungan kurang kondusif, sikap pemagang akan memperoleh pengalaman belajar dan bekerja yang kurang baik.

### **1.5 Prosedur Program *Internship***

Secara umum prosedur *internship* dilakukan melalui beberapa tahap. Tahap pertama mahasiswa mempersiapkan ketentuan (aplikasi, persyaratan) yang ditujukan ke Kepala Program Studi atau Pembantu Bidang Kemahasiswaan, tahap kedua dilakukan seleksi administrasi meliputi : IPK, Kosentrasi Subjek *Internship* dan penilaian aspek individu. Tahap ketiga pengiriman aplikasi ke perusahaan-perusahaan rekanan. Tahap keempat penempatan sesuai dengan kriteria dan kebutuhan perusahaan (dilakukan melalui test, wawancara atau secara langsung). Dalam pembuatan *internship* ini, isi laporan berupa analisis, perancangan dan pembuatan aplikasi/alat. Untuk itu dalam pembuatan laporan *internship* ditekankan pada permasalahan yang mahasiswa/i dapatkan di perusahaan.



Gambar 1.1 Prosedur Program *Internship*

## 1. Mentoring

Agar tercapai tujuan serta manfaat bagi kedua belah pihak, maka diperlukan adanya sistem bimbingan atau mentor yang dilakukan oleh karyawan yang ditunjuk oleh pihak manajemen perusahaan, sehingga terjadi sinergitas antara mahasiswa dan perusahaan.

### a. Definisi

Mentoring adalah suatu proses hubungan yang dibangun antara mentor dengan peserta *internship* (mahasiswa) yang di fokuskan pada upaya pengembangan *soft skill* mahasiswa, baik dari segi *attitude* maupun *skill* demikian juga dengan dengan *competency* dalam rangka meningkatkan produktivitas kerja peserta pada saat ini maupun masa yang akan datang.

### b. Tujuan

Mentoring bertujuan untuk memberikan bantuan bagi para mahasiswa yang *internship* agar mereka lebih cepat beradaptasi dengan lingkungan kerja, karyawan lain serta dengan tugas-tugas yang diberikan kepadanya. Sedangkan target mentoring adalah memberikan dorongan, motivasi serta transfer *of knowledge* yang dibutuhkan bagi para mahasiswa yang *internship*, agar mereka lebih peduli dan lebih

cepat beradaptasi sehingga mereka dapat memberikan kontribusi yang diharapkan oleh perusahaan.

c. Sasaran Pengembangan

Sasaran yang ingin dicapai dalam mentoring ini adalah pengembangan sikap kerja para mahasiswa yang ikut *internship*, pengembangan kompetensi, baik yang bersifat *generic* maupun yang bersifat *technical*, sehingga terjadi *transfer of knowledge* yang didasarkan pada proses pembelajaran ditempat kerja (*learning by doing*).

d. Personel Mentoring

Terdiri dari dua mentor : 1). Mentor dari program studi sebanyak maksimum 2 orang dengan penempatan dari program studi atau jurusan, 2). Mentor dari perusahaan maksimum 2 orang dengan penetapan dari perusahaan yang bertalian.

e. Peran dan Tanggung Jawab

Agar program dan internship ini dapat berjalan dengan baik sesuai dengan pola kemitraan maka harus ada komitmen yang harus ditaati dan dihormati oleh para peserta internship. Adapun peran dan tanggung jawab baik mentor maupun peserta internship dapat diuraikan pada *table* ini.



No.	Mentor	Peserta <i>Internship</i>
1.	Memberikan tugas-tugas selama masa <i>internship</i> yang dituangkan dalam <i>Activity Plan</i> peserta. Dalam menyusun <i>Activity Plan</i> ini mentor diharapkan dapat memberikan informasi dan ide-ide untuk menarik inisiatif dan kreativitas peserta <i>internship</i> .	Melakukan tugas-tugas yang menjadi tanggung jawabnya.
2.	Memberikan bimbingan dalam bentuk arahan maupun <i>feed back</i> sesuai dengan target kompetensi yang harus dikuasai peserta <i>internship</i> .	Mempelajari hal-hal yang bersifat teknis di Unit kerjanya dalam rangka menguasai kompetensi teknik sesuai dengan standar yang diharapkan.
3.	Membantu apabila peserta mengalami kesulitan, baik dalam masalah teknis pekerjaan maupun koordinasi dengan pihak-pihak terkait. Apabila ada masalah dengan sikap peserta, mentor diharapkan bisa mengkoordinasikan masalah tersebut dengan <i>Human Resources Dept</i> yang bertalian.	Mengkomunikasikan setiap ada masalah kerja yang dihadapinya secara jelas kepada mentor.
4.	Melakukan evaluasi terhadap kinerja dan kompetensi peserta selama <i>internship</i> sesuai dengan kriteria evaluasi yang disediakan oleh pihak perusahaan	Menerima <i>feed back</i> secara konstruktif.

5.	Pemberian arahan masukan tentang pembuatan pelaporan <i>internship</i> sesuai dengan <i>project paper</i> yang akan diminati peserta <i>internship</i> .	Mengumpulkan data, temuan penyimpangan dan masalah yang kemudian dipresentasikan serta usulan perbaikannya (bila ada).
----	--	--

## 1.6 Unsur Pembelajaran Magang

Proses pembelajaran magang memiliki beberapa unsur yang perlu mendapatkan perhatian agar proses pembelajaran tersebut dapat meningkatkan kemandirian warga belajar. Unsur-unsur tersebut adalah sebagai berikut (Kamil, 2002:51) :

1. Pemagang (orang yang belajar bekerja) pada konteks pemagang ada beberapa factor yang perlu diperhatikan diantaranya adalah :
  - a. Bakat dan minat, hal ini perlu diperhitungkan karena mempengaruhi keberhasilan pelaksanaan magang, sebab seseorang yang mengikuti magang tetapi tidak sesuai dengan bakat yang dimiliki serta minat yang dikehendaki, kemungkinan besar akan mengalami kesulitan baik dalam proses belajar bekerja maupun pencapaian tujuan.
  - b. Kebutuhan, kebutuhan ini perlu diperhitungkan baik yang berkaitan dengan kebutuhan individu pemagang (*need assessment*) maupun kebutuhan pasar kerja, hal ini perlu mendapat perhatian agar tidak mengalami kesulitan setelah selesai mengikuti magang.

- c. Kemampuan, kemampuan yang dimaksudkan disini adalah kemampuan mengikuti magang untuk menyadap pengetahuan, keterampilan dan sikap mental yang diberikan sumber magang maupun kemampuan untuk membiayai dirinya dalam mengikuti magang.
  - d. Faktor lain yang perlu dipertimbangkan adalah kesediaan untuk mandiri setelah selesai magang bila belum mempunyai pekerjaan tetap dan usia yang masih produktif dalam bekerja.
2. Sumber magang (orang yang dimagangi atau permagang).  
Pada komponen ini ada tiga hal yang perlu diperhatikan diantaranya adalah :
- a. **Kesediaan**  
Kesediaan sumber magang untuk dimagangi, menularkan pengetahuan dan keterampilan yang dimiliki kepada pemagang, baik secara sukarela ataupun dengan imbalan. Apabila sumber magang tidak bersedia dimagangi, tidak bersedia menularkan pengetahuan dan keterampilannya dalam hal ini tidak mungkin terjadi proses magang.
  - b. **Kemampuan**  
Kemampuan dalam arti sumber magang (permagang) harus mahir dalam menguasai keterampilan serta mahir dalam menularkan

keterampilan yang dimilikinya, serta memiliki kemauan untuk menularkan apa yang dimilikinya.

### 3. Pola Magang

Proses pembelajaran dalam pendidikan luar sekolah, baik dalam bentuk magang atau dalam bentuk lainnya, harus dilakukan melalui berbagai pola yang mendukung terhadap proses dan keberhasilan dari proses tersebut. Untuk lebih jelasnya pola tersebut bias dari kebutuhan/tujuan, materi/bidang, mata pencaharian, sumber, persyaratan, dll.

Magang sebagai alternatif pendidikan luar sekolah memiliki unsur yang sama dengan program pendidikan lainnya. Magang dalam penelitian ini berada pada ruang lingkup kelompok produksi. Menurut Sudjana (dalam Nugroho, 2008) pelaku-pelaku kegiatan belajar dalam kelompok produksi yang memiliki keahlian dalam peningkatan kualitas produksi, pengolahan bahan baku dan penggunaan alat-alat produksi. Kedua, orang-orang atau anggota kelompok yang belum memiliki kemampuan dalam peningkatan produksi. Mereka inilah yang belajar dari orang-orang yang disebut pertama melalui magang. Ketiga adalah orang-orang yang telah memiliki kemampuan dari kelompok pertama namun masih dalam tingkatan yang lebih rendah.

### **1.7 Fase Magang**

Magang mencakup serangkaian fase yang membantu mengartikulasikan peran para pelajar dan guru selama proses mengamati dan memberlakukan konsep magang. Fase magang meliputi :

1. Tahap I : Modeling – Tindakan lengkap meliputi pengamatan dan perenungan. Berarti bahwa bagian-bagian yang lebih kecil yang membentuk keseluruhan belum diteliti secara mendalam. "Modeling terjadi dalam dua bagian: pemodelan perilaku memungkinkan peserta didik untuk mengamati kinerja dari suatu kegiatan oleh anggota yang berpengalaman untuk berbagi" trik perdagangan "dengan anggota baru" (Hansman, 2001: 47). Pelajar menggunakan artikulasi dan domain-spesifik heuristik dalam fase ini (Brandt et al.,1993).
2. Tahap II : Approximating – Dalam swasta atau non-kritis skenario, pengamat mulai meniru tindakan guru. Melalui bimbingan dekat, pelajar mulai mengartikulasikan lebih jelas tindakan guru. Fase ini memungkinkan pelajar untuk mencoba kegiatan dan memungkinkan mereka berpikir tentang apa yang mereka rencanakan untuk dilakukan dan mengapa mereka berencana untuk melakukannya. Kemudian setelah kegiatan berakhir, pelajar meniru kegiatan ini. Mereka meneliti apa yang mereka lakukan dibandingkan dengan apa yang pemagang ahli lakukan.

3. Tahap III: Memudar – Peserta didik, masih dalam pengawasan, mulai beroperasi secara lebih rinci, bermain dalam struktur yang telah diajarkan. Kemampuan pelajar meningkat sebagai bantuan ahli (Hansman, 2001).
4. Tahap IV: Self-directed Learning Peserta didik mencoba tindakan nyata dalam masyarakat, tetap membatasi dirinya untuk lingkup tindakan di lapangan yang dipahami. Pelajar melakukan tugas yang sebenarnya dan hanya mencari bantuan bila diperlukan dari ahli (Hansman, 2001).
5. Tahap V: Generalisasi – Pelajar merealisasikan apa yang telah dipelajari, mencoba untuk menerapkan keterampilan dan terus tumbuh dalam kemampuan di lapangan. Pelajar menggunakan diskusi dalam fase ini untuk menghubungkan bahwa mereka telah belajar untuk situasi lain yang relevan (Hansman, 2001).

### **1.8 Strategi Pembelajaran Magang**

Strategi pembelajaran magang ditujukan agar pelaksanaan program magang dapat mendapatkan hasil yang sesuai dengan tujuan yang ingin dicapai oleh peserta didik. Dalam pelaksanaan pembelajaran magang perlu diperhatikan beberapa strategi pembelajarannya, seperti perumusan tujuan pelaksanaan magang, kemampuan peserta didik, penggunaan berbagai metode dan media pembelajaran dan didukung oleh sarana peralatan kerja serta unsur-unsur pembelajaran magang yang meliputi :

1. Identifikasi kebutuhan. Hal ini berhubungan erat dengan kebutuhan individu pemagang maupun kebutuhan pasar

kerja. Dengan mengidentifikasi kebutuhan, pemagang akan mengerti kebutuhan pendidikan apa yang dia butuhkan selama proses magang berlangsung. Dan dengan mengidentifikasi kebutuhan ini pemagang tidak mengalami kesulitan setelah selesai mengikuti magang (Kamil,2002: 52).

2. Perumusan tujuan pembelajaran magang. Perumusan tujuan pembelajaran diperlukan sebelum pelaksanaan pembelajaran magang. Tujuan pembelajaran magang hendaknya bertujuan untuk memperbaiki pengetahuan lama yang telah dimiliki dengan pengetahuan baru dalam bidang manajemen dan pelatihan kepekaan pemagang. Memberikan tambahan keterampilan dan pengetahuan baru supaya dapat memperoleh pengalaman yang lebih baik (Nur Halim, 2011: 77). Perumusan tujuan magang hendaknya bertujuan agar pemagang mampu memiliki keterampilan tambahan dan mengasah keterampilan tersebut sehingga nantinya ia dapat menjadi mandiri dan dapat meningkatkan taraf hidupnya.
3. Penentuan sasaran pembelajaran magang. Dalam menentukan sasaran magang, perlu diperhatikan unsur bakat dan minat. Karena bakat dan minat memiliki pengaruh dalam keberhasilan pelaksanaan magang. Sebab seseorang yang mengikuti magang tetapi tidak sesuai dengan minat yang besar, kemungkinan besar akan

mengalami kesulitan dalam kegiatan belajar maupun pencapaian tujuan (Kamil, 2002: 51).

4. Kegiatan magang.

Kegiatan magang terdiri atas beberapa tahap, diantaranya adalah tahap modeling, approximating, tahap memudar, tahap *self directed learning* dan tahap generalisasi (internet).

5. Materi magang.

Pemberian pekerjaan dalam pembelajaran magang menurut Putut Hargiyarto (2007) sebagai berikut :

- a. Pemberian materi/pekerjaan kepada peserta didik dimulai dengan pekerjaan yang ringan, mudah, dan tidak berisiko tinggi menyebabkan kerusakan.
- b. Secara bertahap peserta diberikan keterampilan dengan pekerjaan yang makin berat, sulit dan berisiko kerusakan tinggi.

6. Menetapkan sumber belajar.

Dalam menetapkan sumber belajar magang harus disesuaikan dengan kesediaan dan kemampuan sumber magang untuk dimagangi, menularkan pengetahuan dan ketrampilan yang dimiliki kepada pemagang (Kamil, 2002).

7. Beberapa metode yang digunakan dalam pelaksanaan magang antara lain :



- a. Metode survey yakni dengan mencari informasi dan mengetahui tentang segala sesuatu yang berkaitan dengan materi/isi pembelajaran yang akan digunakan saat pelaksanaan magang.
- b. Merancang skema untuk rencana pelaksanaan magang.
- c. Mengaplikasikan ilmu dan ketrampilan yang telah dikuasainya ke dalam kehidupan sehari-hari.

8. Pendekatan yang digunakan.

Pendekatan dalam pembelajaran magang menggunakan pendekatan individual, sehingga dapat mudah tercapai tujuan pembelajaran yang diinginkan karena pemagang ahli mengenal secara individual masing-masing pemagang/warga belajarnya.

9. Evaluasi.

Evaluasi dari pembelajaran magang menggunakan evaluasi proses dan akhir. Karena pemagang dinyatakan berhasil dan mampu apabila ia mampu mengaplikasikan ilmu yang diperolehnya selama pembelajaran magang ke dalam kehidupan sehari-hari yang mana dapat meningkatkan taraf hidup pemagang tersebut.

### **1.9 Prinsip-prinsip Magang**

Prinsip magang yang dibahas dalam penelitian ini berkaitan dengan konsep magang yang dapat menghasilkan tenaga kerja terampil, kompeten dan produktif, khususnya konsep

magang secara tradisional (*Grassroot*) yang banyak terjadi pada perusahaan industri kecil atau *home* industri.

Prinsip pemagangan menurut Depnaker (1994), memiliki aspek yang menjadi cirinya yaitu pencapaian kualifikasi yang jelas, ada criteria latihan pemagangan, ada aturan main pelaksanaan, ada uji keterampilan dan sertifikasi serta kontrak pemagangan. Jenjang pemagangan terdiri atas (a) sertifikasi perusahaan untuk tingkat dasar, (b) sertifikasi Nasional tingkat II untuk tingkat menengah, (c) Sertifikasi Nasional Tingkat II untuk tingkat lanjutan, (d) Sertifikasi uji ketrampilan tingkat I untuk tingkat pengelola.

#### **1.10 Sistem Evaluasi**

Program *Internship* diharapkan mampu memberikan kontribusi positif bagi kedua belah pihak. Hal yang memerlukan proses evaluasi berkesinambungan yang memungkinkan perbaikan-perbaikan yang relevan bagi program ini. Harapan lebih besar program ini dapat diposisikan sebagai bagian dari suatu strategi *recruitment* jangka panjang (*long term recruitment strategy*) bagi perusahaan yang bertalian, sehingga selama masa *internship* itu berjalan, maka kesempatan dimaksud dapat digunakan atau *internship* dalam situasi kerja sebenarnya. Secara umum evaluasi ini dilakukan secara bertahap sebagaimana pada gambar 1, berupa presentasi awal dari peserta setelah 2 bulan melakukan kegiatan *internship/internship*, dan presentasi akhir setelah peserta melakukan kegiatan *internship* dengan jangka waktu yang ditentukan dari pihak perusahaan maupun perguruan tinggi.

## 1. Tujuan Evaluasi

Evaluasi bertujuan antara lain :

- a. Untuk memberikan masukan serta sebagai dasar penilaian kemampuan dan hasil yang diperoleh selama melakukan *internship/internship*.
- b. Melakukan penilaian atas capaian kesempatan kontribusi positif keilmuan dalam kegiatan operasional perusahaan.
- c. Lebih jauh nosa rekomendasi kepada pihak manajemen perusahaan dalam pengambilan keputusan khususnya dalam recruitment pegawai baru.

## 2. Kriteria

Evaluasi terdiri dari 2 (dua) bagian yaitu yang menyagkut dengan :

- a. *Generic Competency*, yaitu kompetensi umum yang dimiliki oleh setiap peserta untuk tingkat/level pegawai yang dipersyaratkan oleh perusahaan yang bertalian.
- b. *Technical Competency*, yaitu pengentahuan dan *technical skill* yang dibutuhkan untuk menjalani suatu tugas, jabatan karyawan pada tingkat/level tertentu di perusahaan yang bertalian.

## 3. Proses Evaluasi

Agar evaluasi ini berjalan dengan baik, disamping dapat digunakan oleh manajemen Politeknik Pos Indonesia maupun

perusahaan, maka evaluasi *internship* dapat dilakukan melalui 2 (dua) tahap yaitu :

1. *Progress review generic competency*, targetnya adalah untuk mengetahui perkembangan proyek peserta dan hambatan yang dihadapi dalam penyelesaiannya, serta untuk mengevaluasi sikap dan kompetensi peserta *internship*.
2. *Final review generic and technical review*, targetnya adalah evaluasi akhir yang berkaitan dengan penyelesaian proyek sebagai *final report* atau sebagai *internship* yang merupakan persyaratan bagi kelulusan mereka yang menyangkut dengan kemampuan daya serap peserta selama mereka *internship*.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Sistem**

Istilah sistem merupakan istilah dari bahasa Yunani “*system*” yang artinya adalah himpunan bagian atau unsur yang saling berhubungan secara teratur untuk mencapai tujuan. Berikut ini pengertian sistem dari beberapa ahli :

Sistem menurut pendapat Satzinger, Jackson, & Burd (2010, p6) adalah kumpulan komponen-komponen yang saling berkaitan yang berfungsi bersama untuk mencapai beberapa hasil.

Sedangkan sistem menurut pendapat O'Brien, & Marakas (2009, p24) adalah sekelompok komponen yang saling berkaitan dan bekerja sama ke arah tujuan bersama dengan menerima masukan-masukan dan menghasilkan keluaran dalam proses pengelolaan transformasi atau perubahan.

Menurut Romney dan Steinbart (2006, p4), sistem adalah kumpulan dari dua atau lebih komponen yang berinteraksi untuk mencapai tujuan. Sistem terdiri dari subsistem yang lebih kecil, masing-masing melaksanakan fungsi penting dan mendukung sistem yang lebih besar.

Berdasarkan pendapat-pendapat di atas, maka dapat disimpulkan bahwa definisi dari sistem adalah suatu koleksi dari komponen – komponen yang memiliki suatu kesatuan dan berhubungan dengan satu dengan yang lain dan memiliki fungsi yang sama.

### **2.1.1 Pengertian Informasi**

Menurut Rainer & Cegielski (2010, p10), informasi mengacu pada data yang telah diorganisasikan sehingga mempunyai arti dan nilai bagi penerimanya.

Menurut O'Brien & Marakas (2009, p38), informasi merupakan data yang telah diubah menjadi konteks yang berarti dan berguna untuk penerima akhir tertentu.

Berdasarkan definisi diatas dapat disimpulkan bahwa informasi adalah data yang telah diolah untuk menghasilkan suatu hasil (*output*) dimana hasil dari proses tersebut bermanfaat bagi penerimanya.

### **2.1.2 Pengertian Sistem Informasi**

Menurut O'Brien & Marakas (2009, p4), Sistem Informasi adalah suatu kombinasi yang teratur baik dari orang – orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data dimana mencakup kegiatan mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.

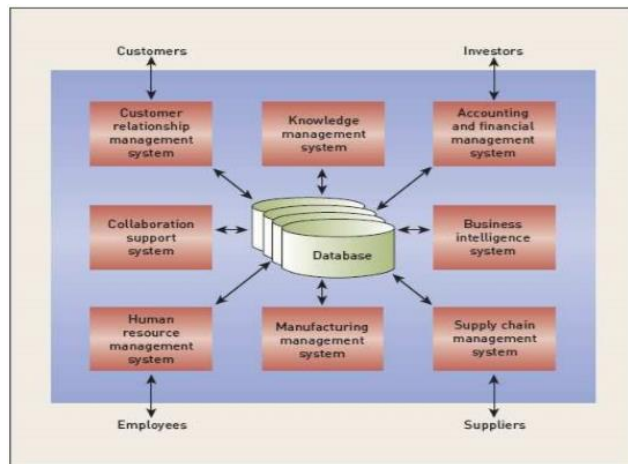
Menurut Satzinger, Jackson, & Burd (2010, p6), Sistem Informasi adalah kumpulan komponen yang saling berhubungan dimana mengumpulkan, memproses, menyimpan, dan menyajikan output dari informasi yang dibutuhkan untuk menyelesaikan tugas bisnis.

Setiap sistem mempunyai sebuah batas antara lingkungan sistem dan sistem itu sendiri. Setiap *input* maupun *output* harus melewati *system boundary*. *System boundary* adalah batas antara sebuah sistem dan lingkungannya dimana harus dilewati oleh *input*

dan *output*. Dalam sebuah sistem informasi, manusia juga merupakan sebuah kunci komponen dan manusia tersebut melakukan sebuah kerja untuk diselesaikan oleh sebuah sistem. Jadi selain terdapat *system boundary*, juga terdapat batas lainnya yang dinamakan *automation boundary* yang merupakan hal penting untuk seorang sistem analis. *Automation boundary* adalah batas antara bagian sistem yang terotomatisasi dengan bagian sistem yang bersifat manual.

Jenis-jenis Sistem Informasi menurut Satzinger, Jackson, & Burd (2010, p9) sebagai berikut :

- A. *Customer Relationship Management System*
- B. *Supply Chain Management System*
- C. *Accounting and Financial Management System*
- D. *Human Resource Management System*
- E. *Manufacturing Management System*
- F. *Knowledge Management System*
- G. *Collaboration Support System*
- H. *Business Intelligence System*



Gambar 2.1 Jenis-jenis Sistem Informasi

Sumber : (Satzinger, Jackson, Burd, 2010, P9)

### 2.1.3 Komponen – komponen Sistem Informasi

Sistem informasi terdiri dari sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama menerima input serta menghasilkan output dalam proses transformasi yang teratur (Ladjamudin, 2005, p14).

Kerangka konsep berbagai komponen dan aktivitas sistem informasi dapat dilihat pada gambar di bawah ini :





Gambar 2.2 Komponen-Komponen Sistem Informasi

Sumber : (Ladjamudin, 2005,p15)

Berdasarkan gambar di atas, komponen-komponen sistem informasi terdiri dari :

**A. Manusia/User**

manusia diperlukan dalam operasi sistem informasi. Sumber daya manusia meliputi pemakai akhir dan pakar sistem. Pemakai akhir adalah orang yang menggunakan informasi yang dihasilkan sistem informasi, misalnya pelanggan, pemasok, teknisi, mahasiswa, dosen, dan orang-orang yang berkepentingan. Sedangkan pakar sistem informasi adalah orang yang mengembangkan dan mengoperasikan sistem informasi, misalnya system, analyst, developer, operator sistem dan staf administrasi lainnya (Mulyanto, 2009,p31).

## **B. Perangkat Keras/*Hardware***

Sumber daya *hardware* adalah semua peralatan yang digunakan dalam memproses informasi, misalnya komputer dan periferalnya, lembar kertas, *disk magnetic* atau optik dan *flash disk* (Mulyanto, 2009, p31).

## **C. Perangkat Lunak/*Software***

*Software* merupakan sekumpulan perintah/fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer agar melaksanakan sesuatu (Ladjamudin, 2005, p15).

## **D. Data**

Data merupakan dasar sumber daya organisasi yang diperlukan untuk memproses informasi. Data dapat berbentuk teks, gambar, audio maupun video. Sumber daya informasi umumnya diatur, disimpan dan diakses oleh berbagai pengelolaan sumber daya data ke dalam database dan dasar pengetahuan (Ladjamudin, 2005, p15).

## **E. Jaringan/*Network***

Sumber daya jaringan merupakan media komunikasi yang menghubungkan komputer, pemroses komunikasi dan peralatan lainnya dengan kendali *software* komunikasi. Jaringan dapat berupa kabel, satelit, seluler dan pendukung jaringan seperti modem, *software* pengendali serta prosesor antar jaringan (Ladjamudin, 2005, p15).

Keseluruhan komponen sistem informasi tersebut saling terkait satu sama lain dalam sistem informasi. Sistem informasi dibangun menggunakan teknologi komunikasi dan informasi yaitu *hardware*, *software* dan jaringan. Ketiga komponen tersebut dipakai untuk mengolah data yang diperoleh untuk menghasilkan informasi yang lebih bermanfaat. Keseluruhan proses pengolahan informasi tidak lepas dari komponen manusia. Manusia adalah komponen penting sistem informasi karena sistem informasi adalah benda yang tidak bermanfaat bila tidak digunakan oleh manusia.

## **2.2 Pengertian Database**

Menurut O' Brien dan Marakas (2009, p196) database adalah kumpulan terintegrasi dari elemen data yang secara logika saling berhubungan. Berbeda dengan Connolly & Begg (2005, p15) yang mengatakan bahwa database adalah kumpulan relasi-relasi logikal dari data, dan deskripsi dari data yang dapat digunakan bersama dan dibuat untuk memperoleh informasi yang dibutuhkan oleh perusahaan.

## **2.3 Pengertian Aplikasi**

Aplikasi merupakan suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer.

Terdapat beberapa teori yang mendefinisikan aplikasi yang dikemukakan oleh beberapa para ahli, diantaranya adalah :

- a. Menurut Hengky W. Pramana (2012, p17) Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, *game*, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia.
- b. Menurut Shelly, Cashman, Verman (2009, p57) Aplikasi adalah seperangkat instruksi khusus dalam komputer yang dirancang agar kita menyelesaikan tugas-tugas tertentu.
- c. Menurut Dhanta (2009, p32), aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas – tugas tertentu.

Jadi Aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak dengan komputer untuk memudahkan pekerjaan atau tugastugas tertentu seperti penerapan, penggunaan, dan penambahan data yang dibutuhkan.

Menurut O'brien (2010, p124) software komputer terbagi menjadi 2 yaitu :

1. *Application Software*

Mengarahkan kinerja penggunaan tertentu atau aplikasi komputer untuk memenuhi kebutuhan pengolahan informasi dari pengguna

Contoh : *word processing*.

## 2. *System software*

Mengendalikan dan mendukung operasi dari sistem komputer karena melakukan berbagai tugas pengolahan informasi.

Contoh : *operating system, network management, database Management.*

### 2.4 **Pengertian Aplikasi berbasis Web**

Menurut O'Brien (2010, p157), *Web service* merupakan komponen *software* yang berbasis *framework web* dan standar *object-oriented* dan teknologi untuk penggunaan web yang secara elektronik menghubungkan aplikasi *user* yang berbeda dan *platform* yang berbeda. *Web service* dapat menghubungkan fungsi bisnis untuk pertukaran data secara *real time* dalam aplikasi berbasis web.

Banyak dari perusahaan - perusahaan berkembang yang menggunakan Aplikasi berbasis Web dalam merencanakan sumber daya mereka dan untuk mengelola perusahaan mereka. Aplikasi berbasis Web ini menggunakan protokol *HTTP*, aplikasi di sisi server berkomunikasi dengan *client* melalui *Web server*. Aplikasi di sisi *client* umumnya berupa *Web browser* jadi. Aplikasi berbasis *Web (client / server-side script)* berjalan di atas aplikasi berbasis internet.

Menurut Simarmata (2010, p185), aplikasi berbasis web adalah sistem perangkat lunak yang berdasarkan pada teknologi dan standar *World Wide Web Consortium (W3C)*. Mereka

menyediakan sumber daya web spesifik seperti konten dan layanan melalui sebuah antarmuka pengguna dan *browser web*.

#### **2.4.1 Tujuan Aplikasi Berbasis Web**

Tujuan aplikasi berbasis web, yaitu :

1. Aplikasi berbasis web dapat digunakan untuk membantu operasional perusahaan seperti membuat *invoice*, sistem informasi persediaan.
2. Memudahkan dalam penyimpanan data di *database*.
3. Aplikasi berbasis web juga dapat bekerja memonitoring sistem dalam hal tampilan, dapat didesain dan disesuaikan untuk berbagai jenis industri.

#### **2.4.2 Karakteristik Aplikasi Web**

Karakteristik aplikasi web antara lain :

##### **1. Kepadatan Jaringan**

Aplikasi web umumnya berada pada suatu jaringan komputer dan harus bisa melayani kebutuhan komunitas klien yang beragam.

##### **2. Keserempakan**

Sejumlah besar pengguna mungkin akan mengakses aplikasi secara serempak.

##### **3. Jumlah pengguna yang tidak dapat diprediksi**

Jumlah pengguna mungkin beragam dari hari ke hari. Pengguna di hari libur mungkin berada di hari kerja.

4. **Kinerja**

Pengguna aplikasi web akan pergi jika pengguna terlalu lama menunggu aplikasi web.

5. **Ketersediaan**

Menyediakan akses layanan 24 jam

6. **Digerakkan oleh data**

Beberapa aplikasi web bergantung pada layanan data, contoh : *onlineshop*.

7. **Peka terhadap isi**

Kualitas isi dan keindahan tetap merupakan faktor penting kualitas web.

8. **Evolusi yang berkesinambungan**

Merupakan hal yang biasa untuk aplikasi web jika ada pembaharuan setiap saat.

9. **Keamanan**

Terhubung ke internet, menyebabkan aplikasi web rentan serangan dari pihak lain.

10. **Estetika**

Salah satu daya tarik aplikasi web adalah tampilan dan nuansanya.

11. **Kesegaraan**

Aplikasi web sering memerlukan kedisiplinan waktu untuk merilis produk ke pasar dalam waktu beberapa hari atau beberapa minggu.

### 2.4.3 Perbandingan Web dan Aplikasi Desktop

Aspek	Aplikasi Web	Aplikasi Desktop
Kemudahan akses	Mudah diakses dari komputer atau perangkat <i>portable</i> yang memiliki akses internet.	tidak mudah diakses, karena harus mendatangi komputer yang terinstal program tersebut.
Instalasi	Aplikasi web memerlukan instalasi tunggal (yang diletakkan di <i>hosting</i> ).	Aplikasi desktop harus diinstal secara individual pada setiap komputer.
<i>Maintenance</i>	Biasanya dimaintenance oleh penyedia hosting yang bertanggung jawab terhadap web klien (tergantung penyedia hosting)	Dimaintenance secara individu pada komputer yang terinstal program.
Keamanan	Lebih rentan terhadap gangguan keamanan seperti <i>hacking</i> , <i>cracking</i> , dll.	Gangguan keamanan rendah
Biaya	Membutuhkan biaya seperti berlanggan <i>domain</i> , <i>hosting</i> . Jika tidak diperbaharui, aplikasi tidak dapat diakses.	Tidak memerlukan biaya berlangganan, dikarenakan sekali program dibuat hanya memerlukan biaya <i>updating</i> .
Koneksi	Bergantung pada <i>bandwidth</i> untuk mentransfer data ke user, jika <i>bandwidth</i> tidak terpenuhi web akan <i>blank</i> .	Tidak bergantung pada <i>bandwidth</i> yang disediakan.



#### **2.4.4 Manfaat dari pengembangan aplikasi web**

##### **1. Kenyamanan**

Manfaat yang paling penting dari aplikasi berbasis web adalah bahwa yang paling nyaman digunakan. Satu dapat menggunakan aplikasi ini setiap saat dari lokasi manapun di seluruh dunia baik dengan menggunakan komputer atau telepon untuk mengakses data yang dibutuhkan sebagai seluruh database tersedia sepanjang waktu.

##### **2. Biaya yang lebih rendah**

Aplikasi berbasis web biaya murah dari pada aplikasi dekstop karena dukungan yang berkurang dan pemeliharaan, persyaratan lebih rendah pada sistem pengguna akhir dan arsitektur sederhana. Ini juga tidak menyia-nyiakan ruang komputer.

##### **3. Beberapa *Platform***

Biasanya semua aplikasi berbasis web jauh lebih harmonis dari seluruh platform perangkat lunak yang diinstal tradisional. Umumnya ada kebutuhan dari web browser seperti Internet Explorer, Firefox, Netscape. Mereka juga kompatibel dengan sebagian besar sistem operasi komputer (Windows, Linux, atau Mac). Material dari browser atau sistem operasi, tidak ada perbedaan dalam cara aplikasi bekerja. Kualitas pekerjaan tetap sepanjang masa yang sama.

#### 4. *Download software*

Selalu *up-to-date*, aplikasi berbasis web biasanya *up-to-date* tidak perlu menjadi di jalankan selalu, sebagai gradasi. Pengguna tidak harus menggagalkan tentang apakah aplikasi *up-to-date* atau tidak. Ini adalah salah satu manfaat dari aplikasi web *open source*.

#### 2.4.5 **Kelebihan dan Kelemahan Aplikasi berbasis web**

Kelebihan kompetitif dari aplikasi berbasis web :

1. Aplikasi tersebut ringan dan dapat diakses selama ada koneksi internet atau intranet ke *server*.
2. Dapat diakses dengan menggunakan *browser* tanpa harus menginstall aplikasi tersebut.

Kekurangan menggunakan aplikasi berbasis web :

1. Antarmuka yang dapat dibuat terbatas sesuai spesifikasi standar untuk membuat dokumen *Web* dan keterbatasan kemampuan *Web browser* untuk menampilkannya.
2. Terbatasnya kecepatan internet mungkin membuat respon aplikasi menjadi lambat.
3. Tingkat keamanan yang lebih rentan untuk diakses oleh orang lain atau pihak yang tidak berhak.

Oleh karena itu dapat disimpulkan, user dapat mengakses data atau informasi perusahaan mereka melalui laptop, smartphone, atau bahkan komputer PC dengan mudah tanpa harus menginstal terlebih dahulu aplikasi tersebut.

## 2.5 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 mengolah data *form* dari *web*. Jadi, awal mula PHP digunakannya untuk menghitung jumlah pengunjung didalam *web*-nya.

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

Dengan perilisan kode sumber ini menjadi open source, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah dapat mengakses database dan dapat terintegrasi dengan *Hypertext Markup Language* (HTML). Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP / FI secara signifikan. PHP versi 2.0 ini telah menarik banyak perhatian programmer, namun bahasa ini memiliki masalah dengan kestabilan yang kurang bisa diandalkan. Hal ini lebih

dikarenakan Lerdorf hanya bekerja sendiri untuk mengembangkan PHP.

Pada saat itulah Zeev Suraski dan Andi Gutmans, ikut mengambil bagian dan membuat ulang parsing engine yang menjadi dasar dari PHP agar lebih stabil. Dengan dukungan dari banyak programmer lainnya, proyek PHP secara perlahan beralih dari proyek satu orang menjadi proyek masal yang lebih akrab kita kenal *open-source project*. PHP selanjutnya dikembangkan oleh The PHP Group yang merupakan kumpulan banyak programmer dari seluruh dunia.

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

PHP versi 4.0 keluar pada tanggal 22 Mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya Zend Engine yang dibuat oleh Zeev Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP *scripting engine*, yang lainnya adalah build in HTTP *session*, tidak lagi menggunakan library tambahan seperti pada PHP. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi web. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan *web server*.

PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Beberapa penambahan fitur meliputi PHP *Data Objects* (PDO) untuk mengakses *database*, *closures*, *trait*, dan *namespaces*.

Versi lanjutan dari PHP, yakni PHP 6.x sebenarnya telah lama dikembangkan, bahkan sejak tahun 2005. Fokus pengembangan PHP 6 terutama dalam mendukung *unicode*, agar PHP bisa mendukung berbagai jenis karakter bahasa *non-latin*.

Namun dikarenakan beberapa alasan seperti kurangnya programmer, dan performa yang tidak memuaskan, pengembangan PHP 6 dihentikan dan fitur yang ada dimasukkan kedalam PHP 5.

Pada tahun 2014, sebuah proyek lanjutan PHP mulai mengemuka, yakni PHP 7 yang berkembang dari banyak eksperimen yang dinamakan PHP *Next Generation* (PHPNG), yang dikembangkan Dmitry Stogov, Xinchun Hui, dan Nikita Popov. Proyek ini menggunakan pendekatan modern agar PHP diproses lebih cepat seperti memakai teknik just-in-time (JIT) compiler. Proyek PHPNG bertujuan untuk menyusun ulang kode

PHP untuk meningkatkan performa. Selain performa yang meningkat, terdapat beberapa fitur baru pada PHP 7, seperti *combined comparison operator* atau dikenal dengan *spaceship operator* “`<`”, *anonymous classes*, dan dukungan yang lebih stabil untuk server 64-bit.

Beberapa fitur yang sudah usang (*deprecated*) juga dihapus, seperti penulisan PHP dengan ASP style `<% %>` dan tag `<script language=php></script>`. Kedua cara ini sudah tidak bisa digunakan lagi. Modul *mysql extension* juga dihapus karena sudah digantikan dengan *mysqli extension*. Penghapusan modul *mysql extension* ini sering menjadi masalah karena banyak kode program atau buku PHP lama yang masih menggunakannya.

### **2.5.1 Pengertian PHP**

Menurut Doyle (2009, p3), PHP adalah bahasa pemrograman untuk membuat situs web dinamis dan interaktif. PHP berjalan di *server web* dan melayani pengunjung dengan halaman web sesuai permintaan. Artinya adalah sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di server tetapi disertakan HTML biasa.

PHP dikenal sebagai bahasa *scripting* yang menyatu dengan tag HTML, dieksekusi di *server* dan digunakan untuk membuat halaman web yang dinamis seperti ASP (*Active Server Pages*) dan JSP (*Java Server Pages*).

Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi aplikasi yang dijalankan diatas teknologi web. Dalam hal

ini, aplikasi pada umumnya akan memberikan hasil pada web *browser*, tetapi prosesnya secara keseluruhan dijalankan *web server*.

**Kelebihan PHP** Ketika *e-commerce* semakin berkembang, situs-situs yang statis pun semakin ditinggalkan, karena dianggap sudah tidak memenuhi keinginan pasar, padahal situs tersebut harus tetap dinamis. Pada saat ini bahasa PERL dan CGI sudah jauh ketinggalan jaman sehingga sebagian besar *designer web* banyak beralih ke bahasa *server-side scripting* yang lebih dinamis seperti PHP. Seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan system database didalam web.

### **2.5.2 Fungsi PHP**

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

#### **a. Pembatas**

PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP. Apapun di luar pembatas tidak diproses oleh PHP. Pembatas paling umum adalah “<?php” untuk membuka dan “?>” untuk menutup kode PHP. Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML, dan Javascript.

b. Variabel

Variabel dalam PHP diawali dengan simbol dolar “\$”. Pada versi PHP 5 diperkenalkan jenis isyarat yang memungkinkan fungsi untuk memaksa mereka menjadi parameter objek dari *class* tertentu, *array*, atau fungsi. Namun, jenis petunjuk tidak dapat digunakan dengan jenis skalar seperti angka atau *string*. Contoh variabel dapat ditulis sebagai *\$nama\_variabel*.

c. Komentar

PHP memiliki 3 jenis sintaks sebagai komentar pada kode yaitu blok “/ \* \* /”, komentar 2 baris “//”, serta tanda “#” digunakan untuk komentar 1 baris. Komentar bertujuan untuk meninggalkan catatan pada kode PHP dan tidak akan diterjemahkan ke program.

### 2.5.3 Keunggulan & Kekurangan

a. Keunggulan bahasa pemrograman PHP, antara lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web server* yang mendukung PHP dapat ditemukan di mana-mana dari mulai *apache*, *IIS*, *Lightpad*, hingga *xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya *developer* yang siap membantu dalam pengembangan.



4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
  5. PHP adalah bahasa *open source* yang digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta dapat menjalankan perintahperintah sistem.
- b. Kekurangan bahasa pemrograman PHP, antara lain :
1. Tidak ideal jika untuk pengembangan skala besar.
  2. Tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya.
  3. Tidak dapat memisahkan antara tampilan dengan logik dengan baik.
  4. PHP mempunyai kelemahan *security* tertentu yang mana jika *programmer* tidak jeli dalam melakukan pemrograman dan kurang memperhatikan isu dan konfigurasi PHP.

## **2.6 Pengenalan OOP**

Konsep *Object Oriented Programming* (OOP) pertama kali muncul di MIT (*Massachusetts Institute of Technology*) pada era 1960-an. Sekitar beberapa tahun kemudian antara 1962-1965, sebuah bahasa pemrograman yang mendasari konsep OOP diperkenalkan dengan nama bahasa pemrograman SIMULA 1, dikembangkan oleh Kristen Nygaard dan OleJohan yang merupakan warga negara Norwegia. Setelah itu pada tahun 1967 keluarlah SIMULA 67.

Bahasa SIMULA dengan OOP menginspirasi banyak pengembang, sehingga pada tahun 70-an tercipta bahasa SMALTALK, bahasa yang pertama kali disebut oleh dunia sebagai *object oriented programming*, dengan membawa fitur barunya yaitu *inheritance*.

Semenjak itu konsep OOP semakin dikenal, pada tahun 1980-an banyak bahasa pemrograman menggunakan konsep OOP, diperkenalkan kepada dunia, tetapi hanya empat yang cukup populer, diantaranya ADA (US Department of Defense), PROLOG (The Japanese “Fifth Generation Computer Project), Eiffel dan C++. Bahasa pemrograman ADA dan PROLOG dipercaya akan bersaing ketat sebagai bahasa pemrograman yang paling dominan.

Namun pada tahun 1980-an bahasa pemrograman C++ mematahkan kepercayaan tersebut. Bahasa pemrograman C++ menjadi bahasa pemrograman yang populer dan mendominasi hingga sekarang. Bahasa pemrograman C++ yang merupakan gabungan dari 2 konsep bahasa pemrograman, yakni C dan SIMULA.

Semenjak C++ terkenal, banyak sekali pengembang yang terinspirasi oleh C++ dan pada tahun 1990-an, bahasa pemrograman Java diperkenalkan yang mengaku terinspirasi oleh C++, dan tahun 2002 perusahaan Microsoft juga mengeluarkan bahasa turunan dari C++ yaitu C# (C-Sharp), disusul dengan

VB.Net dengan fitur OOP yang merupakan penyempurnaan dari bahasa VB 0.6 yang tidak mendukung fitur OOP.

### **2.6.1 Pengertian OOP**

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

Masing-masing objek dapat memiliki sifat dan tugasnya. Pada paradigma ini, objek-objek tersebut dapat bekerja sendiri dan juga dapat saling bekerja sama dengan kemungkinan untuk saling berhubungan, seperti menerima, mengirim data kepada objek lainnya dan memproses data.

Paradigma OOP dapat dilihat sebagai interaksi dari objek yang saling berhubungan satu sama lain untuk melakukan tugasnya. Pengembangan program OOP dilakukan dengan cara mengikuti model yang telah ada di dalam dunia nyata.

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

### **2.6.2 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 – Eiffel, Smalltalk, Ruby, Jade, dan lain-lain.
- b. Bahasa OOP *hybrid*, merupakan bahasa yang dirancang untuk pemrograman objek dengan beberapa elemen prosedural.
- c. Bahasa OOP *hybrid* dalam *web*, salah seperti bahasa OOP *hybrid*, yang membedakan hanya konsep yang digunakan dalam pemrograman *web*.

### **2.6.3 Konsep Object Oriented Programming**

- a. *Class*, adalah sebuah rancangan untuk mendefinisikan karakter dan perilaku dari objek, yang merupakan kumpulan atas definisi dan fungsi-fungsi dalam suatu unit, untuk suatu tujuan tertentu.
- b. *Object*, adalah dasar dari modularitas dan struktur pada OOP, dan merupakan representasi dari class, objek akan memiliki sifat dan perilaku dari class yang digunakan.
- c. *Encapsulation*, adalah konsep dalam implementasi untuk membungkus data dan fungsi menjadi satu entitas, dan membatasi akses dari luar *class*.
- d. *Inheritance*, adalah konsep pewarisan *class*. *Class* juga dapat menurun dan memiliki apa yang dimiliki oleh *class* lainnya.
- e. *Abstraction*, adalah konsep untuk mendesain sebuah objek, teknik dalam menyembunyikan detail suatu proses dalam

objek tersebut, dengan tujuan untuk memfokuskan pengguna pada fungsi inti objek.

- f. *Polymorphism*, adalah kemampuan dalam menyampaikan pesan tertentu keluar dari hierarki objeknya, dimana objek yang berbeda memberikan tanggapan atau respon terhadap pesan yang sama sesuai dengan sifat masing-masing objek.

#### **2.6.4 Keunggulan & Kekurangan OOP**

a. Keunggulan OOP, diantaranya :

1. OOP menyediakan struktur modular yang jelas untuk program sehingga OOP sangat bagus digunakan untuk mendefinisikan tipe data abstrak dimana detail implementasinya tersembunyi.
2. OOP akan mempermudah dalam maintaining dan memodifikasi kode yang sudah ada. Objek yang baru dapat dibuat tanpa mengubah kode yang sudah ada.
3. OOP menyediakan *framework* untuk *library* kode dimana komponen *software* yang tersedia dapat dengan mudah diadaptasi dan dimodifikasi oleh *programmer*. Hal ini sangat berguna untuk mengembangkan *Graphical User Interfaces* (GUI).

b. Kekurangan OOP, diantaranya :

1. Tidak memperbolehkan implementasi yang kuat pada *reuse*.

2. *Property software* tidak terikat dalam satu unit fungsional sehingga harus *crosscut* di antara komponennya.
3. *Crosscut* tersebut mengakibatkan sulitnya pengembangan dan pemeliharaan.

## 2.7 Pengertian *Framework*

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

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

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

*session*, *security*, kalender, bahasa, manipulasi gambar, dan lain-lain.

### **2.7.1 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 didalam sistem lebih stabil dan handal karena berbasis pada *framework* yang sudah teruji stabilitas dan keandalannya.
4. *Coding Style* konsisten, memudahkan dalam membaca kode dalam menemukan *bugs*.
5. *Security Concern* dan *framework* akan mengantisipasi dengan memasang perisai terhadap adanya berbagai masalah keamanan yang mungkin terjadi.
6. Dokumentasi, *framework* dapat mendisiplinkan kita untuk menulis dokumen apa yang kita tulis.

### **2.7.2 Kelebihan dan Kelemahan *Framework***

Kelebihan Framework antara lain :

1. Lebih cepat dan efisien  
Jika mengerjakan proyek besar, maka penggunaan *framework* dapat membantu mempercepat proses pengembangan. Pada umumnya, *framework* memiliki beragam fungsi dan plugin yang bisa dimanfaatkan.

Dalam kerangka kerja ini, maka proses pengembangan proyek jauh lebih cepat daripada harus menulis kode dari awal. Selain itu, tidak perlu menulis berulang-ulang untuk kode yang bersifat repetitif.

## 2. Menghemat biaya

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

## 3. Memperhatikan faktor keamanan

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

Kelemahan *Framework* antara lain :

### 1. Kurangnya pemahaman bahasa pemrograman

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



mengenai bahasa pemrograman menjadi tidak berkembang.

## 2. Memiliki batasan

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

## 3. Kode Publik

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

## 2.8 Pengenalan CodeIgniter

Kelahiran codeignier adalah bermula dari kegalauan Rick Ellis atas banyaknya kode PHP yang harus ditulis ketika membangun salah satu CMS kesayangannya, *expression engine*. Rick Ellis ingin mempermudah penulisan kode-kode program PHP dan membuatnya lebih singkat dan cepat. Akhirnya Rick Ellis mengambil inisiatif dengan membuat sendiri kode singkat / *shortcode* dari fungsi-fungsi yang ada di PHP.

Pembuatan *shortcode* tersebut membuatnya mampu membangun *expression engine* dengan sangat bagus, efisien dan cepat. Selain itu, performanya juga sangat bagus.

Setelah tidak berapa lama, Rick Ellis melalui situsnya [ellislab\(dot\)com](http://ellislab(dot)com) membagikan *shortcode* yang dibuat sendiri itu untuk digunakan oleh *developer* lainnya. Tujuannya yaitu membantu *developer* lain dalam menangani masalah dalam *framework* yang terkenal. Banyak *developer* turun tangan membantu perkembangan CodeIgniter dan puncaknya, CI menjadi *framework* terpopuler tahun 2006.



Gambar 2.1 Logo CodeIgniter

Sumber :

### **2.8.1 Pengertian 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 penggunaanya 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 (*strach*). 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.

### 2.8.2 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)

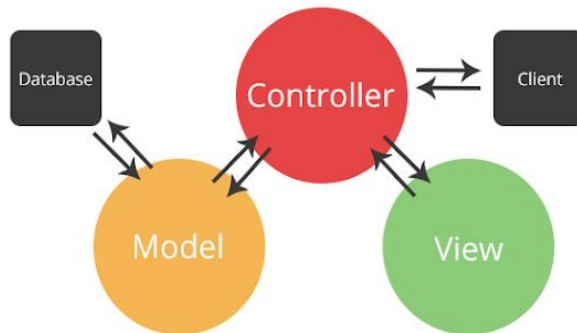
- ✓ *Image Manipulation Library* (cropping, resizing, rotation) Support GD, Image Magic, and NetPBM
- ✓ *File Uploading Class*
- ✓ *FTP Class*
- ✓ *Localization*
- ✓ *PAGINATION*
- ✓ *Data Encryption*

### **2.8.3 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 reduksi 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.



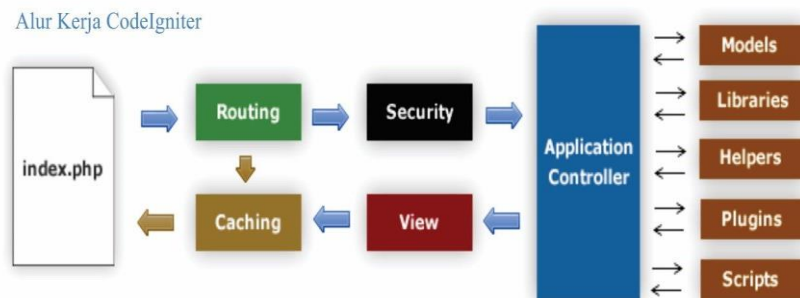
Gambar 2.2 Cara kerja MVC

Sumber :

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.

#### 2.8.4 Sistem Kerja CodeIgniter

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



Gambar 2.3 Alur kerja CodeIgniter

Sumber :

- Index.php: berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- Router : router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program.
- 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.
- 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*.
- Controller : controller akan membuka file bagian model, *core libraries*, *helper* dan semua *resources* yang dibutuhkan dalam program tersebut.

- f. *View*: bagian 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.

### **2.8.5 Fungsi CodeIgniter**

- a. Mempercepat dan mempermudah kita dalam pembuatan *website*.
- b. Menghasilkan struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file phpnya.
- c. Memberikan standar coding sehingga memudahkan kita atau orang lain untuk mempelajari kembali sistem aplikasi yang dibangun.

### **2.8.6 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 berbagi ilmu pengetahuan. Komunitas ini dapat ditemui di [codeigniter.com/forums](http://codeigniter.com/forums).

#### **2.8.7 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.

## **BAB III**

### **GPS**

#### **3.1 *Global Positioning System (GPS)***

*Global Positioning System (GPS)* merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunaanya dimana dia berada (secara global) dipermukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital.

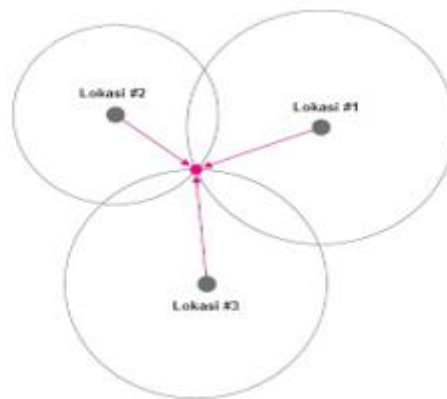
##### **3.1.1 Definisi Global Positioning System (GPS)**

GPS (*Global Positioning System*) adalah sistem navigasi yang berbasiskan satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (*Departemen of Defense*) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *reciever* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi diubah menjadi titik yang dikenal dengan nama *Way-point* nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik.

GPS adalah satu-satunya sistem satelit navigasi global untuk penentuan lokasi, kecepatan, arah, dan waktu yang telah beroperasi secara penuh didunia saat ini (*undergraduate thesis Wildan Habibi, ITS, Surabaya Januari : 2011*). GPS menggunakan

konstelasi 27 buah satelit yang mengorbit bumi, dimana sebuah GPS *receiver* menerima informasi dari tiga atau lebih satelit untuk menentukan posisi.

GPS *receiver* harus berada dalam *line-of sight* (LoS) terhadap ketiga satelit tersebut untuk menentukan posisi, sehingga GPS hanya ideal untuk digunakan dalam *outdoor positioning*.



Gambar 3.1 Trilaterasi Dalam *Global Positioning System* (GPS)

(Sumber : Ary Mazharuddin S, S.Kom., M.Kom.Sc.)

Aplikasi yang berada disisi target (*client*) setelah mendapatkan *request* dari pelacak (*server*) maka *client* akan meminta koordinat posisinya pada GPS (*Global Positioning System*), yang kemudian akan dikirimkan ke pelacak (*server*).

Sejak tahun 1980, layanan GPS yang dulunya hanya untuk keperluan militer mulai terbuka untuk publik. Meskipun satelit-satelit tersebut berharga ratusan juta dolar, namun setiap orang dapat menggunakannya dengan gratis. Satelit-satelit ini mengorbit pada ketinggian sekitar 12.000 mil dari permukaan bumi. Posisi ini

sangat ideal karena satelit dapat menjangkau *area coverage* yang lebih luas. Satelit-satelit ini akan selalu berada posisi yang bisa menjangkau semua area di atas permukaan bumi sehingga dapat meminimalkan terjadinya blank spot (area yang tidak terjangkau oleh satelit).

Setiap satelit mampu mengelilingi bumi hanya dalam waktu 12 jam. Sangat cepat, sehingga mereka selalu bisa menjangkau dimana pun posisi Anda di atas permukaan bumi. GPS *reciever* sendiri berisi beberapa *integrated circuit* (IC) sehingga murah dan teknologinya mudah untuk di gunakan oleh semua orang. GPS dapat digunakan untuk berbagai kepentingan, misalnya mobil, kapal, pesawat terbang, pertanian dan di integrasikan dengan komputer maupun laptop. (Jurnal Andi Sunyoto, STMIK AMIKOM Jogjakarta, 2013:1).

Berikut beberapa contoh perangkat GPS *reciever*



Gambar 3.2 Macam-macam Perangkat GPS

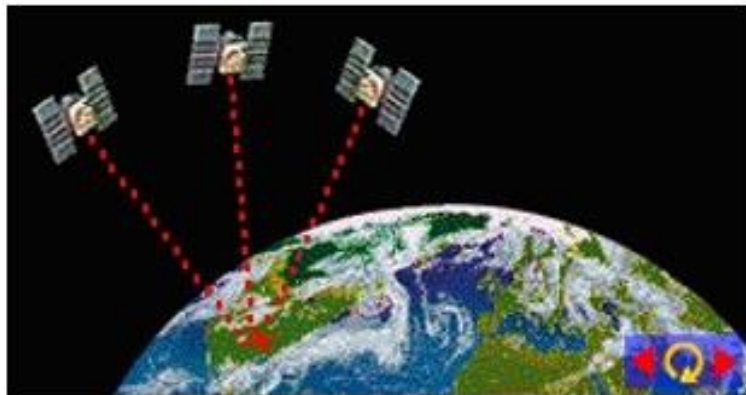
(Sumber : Jurnal Andi Sunyoto, STMIK AMIKOM)

### 3.1.2 Cara kerja Global Positioning System (GPS)

Setiap daerah di atas permukaan bumi ini minimal terjangkau oleh 3-4 satelit. Pada prakteknya, setiap GPS terbaru bisa menerima sampai dengan 12 channel satelit sekaligus. Kondisi langit yang cerah dan bebas dari halangan membuat GPS dapat dengan mudah menangkap sinyal yang dikirimkan oleh satelit. Semakin banyak satelit yang diterima oleh GPS, maka akurasi yang diberikan juga akan semakin tinggi.

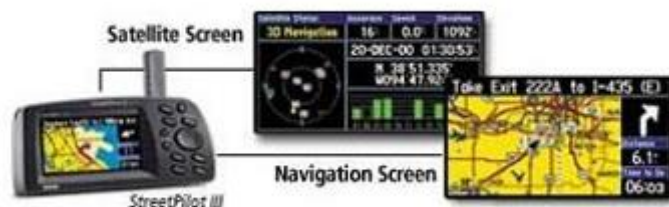
Cara kerja GPS secara sederhana ada 5 langkah, yaitu :

1. Memakai perhitungan "*triangulation*" dari satelit.
2. Untuk perhitungan "*triangulation*", GPS mengukur jarak menggunakan travel time sinyal radio.
3. Untuk mengukur *travel time*, GPS memerlukan memerlukan akurasi waktu yang tinggi.
4. Untuk perhitungan jarak, kita harus tahu dengan pasti posisi satelit dan ketinggian pada orbitnya.
5. Terakhir harus mengoreksi *delay* sinyal waktu perjalanan di atmosfer sampai diterima receiver.



Gambar 3.3 Cara Satelit menentukan posisi  
(Sumber : Jurnal Andi Sunyoto, STMIL AMIKOM)

Satelit GPS berputar mengelilingi bumi selama 12 jam di dalam orbit yang akurat dia dan mengirimkan sinyal informasi ke bumi. GPS *reciever* mengambil informasi itu dan dengan menggunakan perhitungan “*triangulation*” menghitung lokasi *user* dengan tepat. GPS *reciever* membandingkan waktu sinyal di kirim dengan waktu sinyal tersebut di terima. Dari informasi itu didapat diketahui berapa jarak satelit. Dengan perhitungan jarak GPS *reciever* dapat melakukan perhitungan dan menentukan posisi user dan menampilkan dalam peta elektronik.



Gambar 3.4 Tampilan GPS Reciever

Sebuah GPS receiver harus mengunci sinyal minimal tiga satelit untuk menghitung posisi 2D (*latitude* dan *longitude*) dan *track* pergerakan. Jika GPS *receiver* dapat menerima empat atau lebih satelit, maka dapat menghitung posisi 3D (*latitude*, *longitude* dan *altitude*). Jika sudah dapat menentukan posisi *user*, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan masih banyak lagi.

Satelit GPS dalam mengirim informasi waktu sangat presisi karena Satelit tersebut memakai jam atom. Jam atom yang ada pada satelit ialah dengan partikel atom yang di isolasi, sehingga dapat menghasilkan jam yang akurat dibandingkan dengan jam biasa. Perhitungan waktu yang akurat sangat menentukan akurasi perhitungan untuk menentukan informasi lokasi kita. Selain itu semakin banyak sinyal satelit yang dapat diterima maka akan semakin presisi data yang diterima karena ketiga satelit mengirim *pseudo-random code* dan waktu yang sama. Ketinggian itu menimbulkan keuntungan dalam mendukung proses kerja GPS, bagi kita karena semakin tinggi maka semakin bersih atmosfer, sehingga gangguan semakin sedikit dan orbit yang cocok dan perhitungan matematika yang cocok. Satelit harus tetap pada posisi yang tepat sehingga stasiun di bumi harus terus memonitor setiap pergerakan satelit, dengan bantuan radar yang presisi selalu di cek tentang altitude, position dan kecepatannya.



### **3.1.3 Cara Satelit Menentukan Posisi Lokasi**

Sinyal yang dikirimkan oleh satelit ke GPS akan digunakan untuk menghitung waktu perjalanan (*travel time*). Waktu perjalanan ini sering juga disebut sebagai *Time of Arrival* (TOA). Sesuai dengan prinsip fisika, bahwa untuk mengukur jarak dapat diperoleh dari waktu dikalikan dengan cepat rambat sinyal. Maka, jarak antara satelit dengan GPS juga dapat diperoleh dari prinsip fisika tersebut. Setiap sinyal yang dikirimkan oleh satelit akan juga berisi informasi yang sangat detail, seperti orbit satelit, waktu, dan hambatan di atmosfer. Satelit menggunakan jam atom yang merupakan satuan waktu paling presisi. Untuk dapat menentukan posisi dari sebuah GPS secara dua dimensi (jarak).

Dibutuhkan minimal tiga buah satelit. Empat buah satelit akan dibutuhkan agar didapatkan lokasi ketinggian (secara tiga dimensi). Setiap satelit akan memancarkan sinyal yang akan diterima oleh GPS *receiver*. Sinyal ini akan dibutuhkan untuk menghitung jarak dari masing-masing satelit ke GPS. Dari jarak tersebut, akan diperoleh jari-jari lingkaran jangkauan setiap satelit. Lewat perhitungan matematika yang cukup rumit, interseksi (perpotongan) setiap lingkaran jangkauan satelit tadi akan dapat digunakan untuk menentukan lokasi dari GPS di permukaan bumi. (Jurnal Andi Sunyoto, STMIK AMIKOM Jogjakarta, 2013:1).

### **3.1.4 Manfaat Penggunaan Global Positioning System (GPS)**

Dengan menggunakan GPS, Anda dapat menandai semua lokasi yang pernah Anda kunjungi. Misalnya, Lokasi Politeknik

Pos Indonesia kita beri *waypoint* dan tempat-tempat lainnya. Sebenarnya, ada banyak manfaat yang bisa diambil jika Anda mengetahui *waypoint* dari suatu tempat. Pertama, Anda dapat memperkirakan jarak lokasi yang Anda tuju dengan lokasi asal Anda. GPS keluaran terakhir dapat memperkirakan jarak Anda ke tujuan, sampai estimasi lamanya perjalanan dengan kecepatan aktual yang sedang Anda tempuh. Kedua, lokasi di daratan memang cukup mudah untuk dikenali dan diidentifikasi. Namun, jika Anda kebetulan berada ditempat memancing yang terletak di tengah lautan ataupun tempat melihat matahari terbenam yang berada di puncak gunung. Di saat seperti inilah sebuah GPS akan menunjukkan manfaatnya.

Dengan teknologi GPS dapat digunakan untuk beberapa keperluan sesuai dengan tujuannya. GPS dapat digunakan oleh peneliti, olahragawan, petani, tentara, pilot, petualang, pendaki, pengantar barang, pelaut, kurir, penebang pohon, pemadam kebakaran dan orang dengan berbagai kepentingan untuk meningkatkan produktivitas, keamanan, dan untuk kemudahan. Dari beberapa pemakai di atas dikategorikan menjadi :

1. Lokasi

Digunakan untuk menentukan dimana lokasi suatu titik dipermukaan bumi berada.

2. Navigasi

Membantu mencari lokasi suatu titik di bumi.

### 3. *Tracking*

Membantu untuk memonitoring pergerakan obyek dan membantu memetakan posisi tertentu, dan perhitungan jaringan terdekat.

### 4. *Timing*

Dapat dijadikan dasar penentuan jam seluruh dunia, karena memakai jam atom yang jauh lebih presisi di banding dengan jam bisa.

Tidak peduli posisi Anda, di tengah laut, di tengah hutan, di atas gunung, ataupun di pusat kota. Selama GPS dapat menerima sinyal dari satelit secara langsung tanpa halangan, maka GPS akan selalu memberikan informasi koordinat posisi Anda. GPS membutuhkan area pandang yang bebas langsung ke langit. Halangan-halangan seperti pohon, gedung, bahkan kaca film kelas *V-Kool*, bisa mengurangi akurasi sinyal yang diterima oleh GPS. Bahkan bukan tidak mungkin GPS tidak bisa menerima sinyal sama sekali dari satelit.

GPS juga memiliki *feature* tambahan yang mampu memberikan informasi selama anda di perjalanan, seperti kecepatan, lama perjalanan, jarak yang telah ditempuh, waktu, dan masih banyak. (Jurnal Andi Sunyoto, STMIK AMIKOM Jogjakarta, 2013:1).

## 3.2 **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.

### **3.2.1 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/androidapi/>*

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 Embeded 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 Embeded 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/webservices/>*.

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.

### **3.2.2 Kelebihan dan Kekurangan Google Maps API**

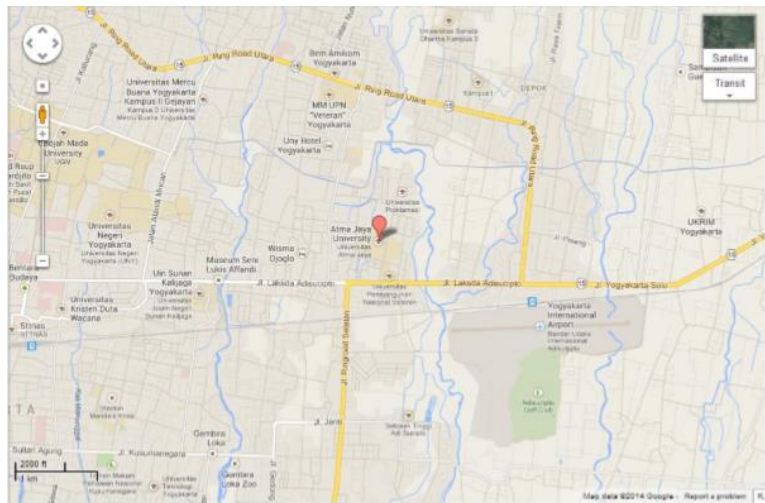
- a. Kelebihan Google Maps API
  - Dukungan penuh yang dilakukan google sehingga terjamin dan bervariasi fitur yang ada pada Google Maps API.



- Banyak pengembang yang menggunakan Google Maps API sehingga mudah dalam mencari referensi dalam pengembangan aplikasi.
- b. Kekurangan Google Maps API
  - Jika ingin mengakses maka harus terkoneksi ke internet pada perangkat yang digunakan.

### **3.3 Peta**

Peta adalah gambar atau lukisan pada kertas dan sebagainya yang menunjukkan letak tanah (Anonim, 2013). Peta bisa disajikan dalam berbagai cara yang berbeda, mulai dari peta konvensional yang tercetak hingga peta digital yang tampil di layar komputer. Istilah peta berasal dari bahasa Yunani mappa yang berarti taplak atau kain penutup meja. Namun secara umum pengertian peta adalah lembaran seluruh atau sebagian permukaan bumi pada bidang datar yang diperkecil dengan menggunakan skala tertentu. Sebuah peta adalah representasi dua dimensi dari suatu ruang tiga dimensi. Banyak peta mempunyai skala, yang menentukan seberapa besar objek pada peta dalam keadaan yang sebenarnya.



Gambar 3.5 Ilustrasi Peta Google Map

## BAB IV

### IMPLEMENTASI SISTEM

#### 4.1 Instalasi Aplikasi yang akan digunakan

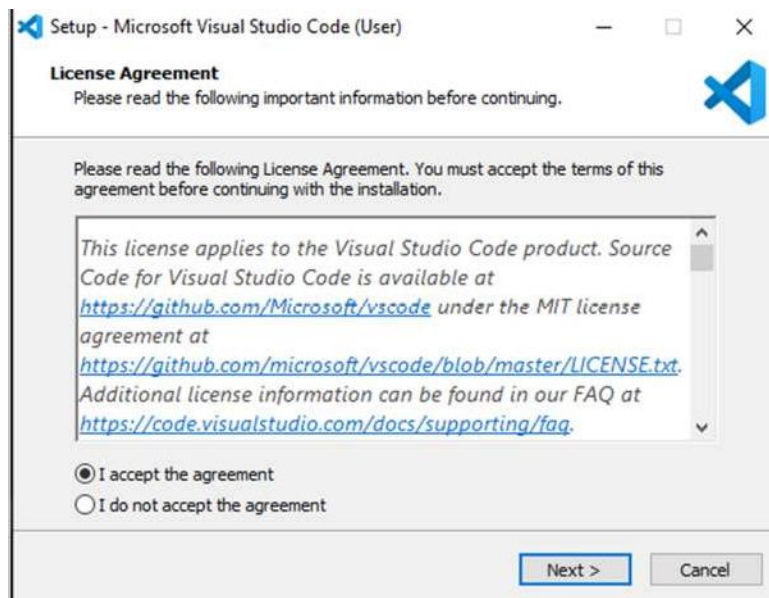
Visual studio code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node.js, serta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

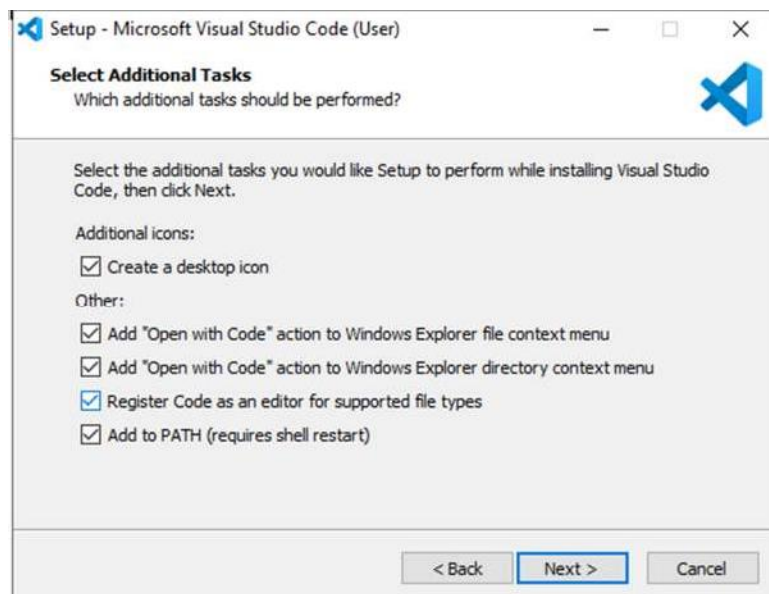
- a. Pertama, *download* terlebih dahulu file installer visual studio code melalui situs resminya pada link berikut ini <https://code.visualstudio.com/download>.



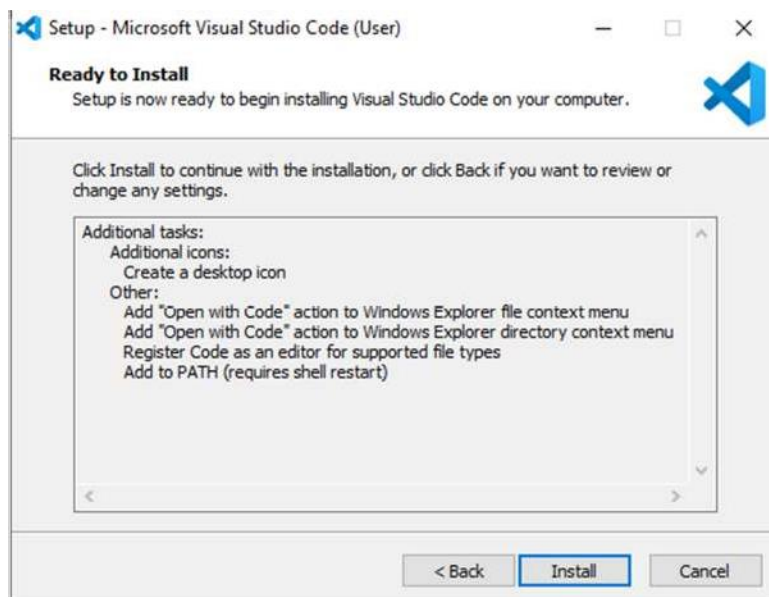
- b. Terdapat 2 pilihan untuk sistem operasi windows, diantaranya ada yang 64 bit dan 32 bit. Pilihlah installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Jika, sudah berhasil di *download*, lanjut pada proses instalasi, pertama klik pada file installer vs code.
- d. Pilih “I accept the agreement”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



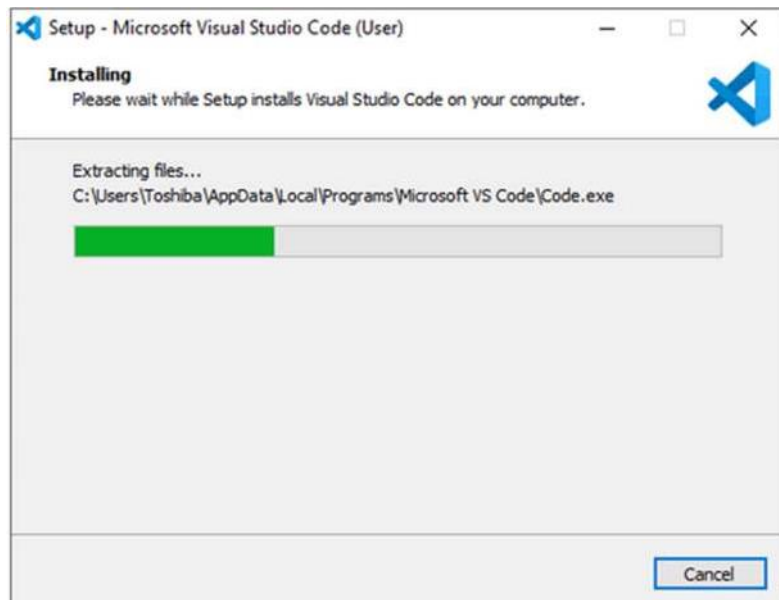
- e. Pada bagian select additional tasks, centang semuanya kemudian klik next.



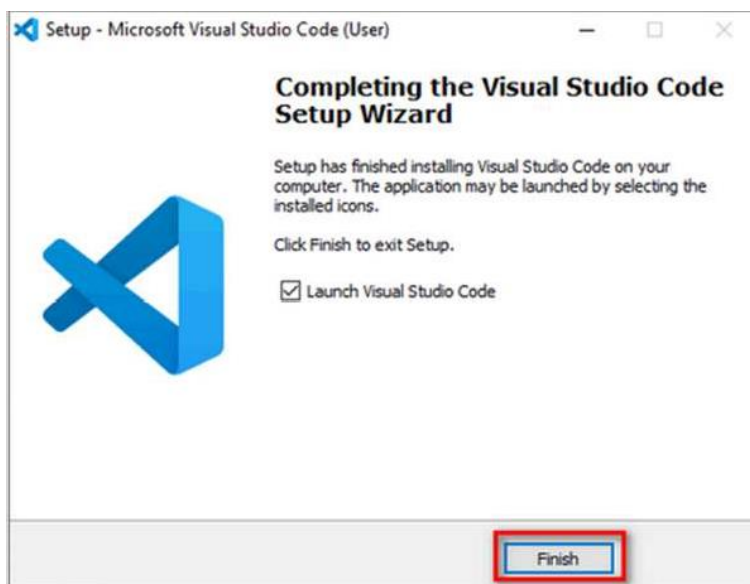
f. Kemudian klik install untuk melanjutkan proses instalasi.



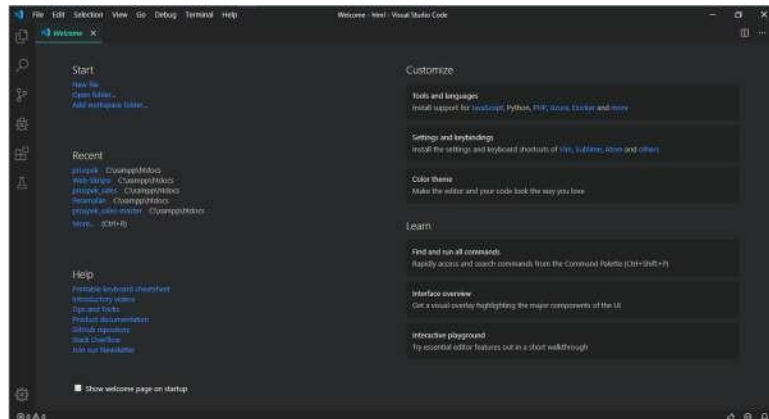
g. Tunggu proses instalasi hingga selesai.



h. Setelah selesai klik Finish.



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



## 4.2 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 ada di dalam nama XAMPP memiliki arti sebagai berikut.

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

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

**M = MySQL / MariaDB**, MySQL atau MariaDB merupakan aplikasi *database server* yang dikembangkan oleh orang yang

sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui *database*.

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

**P = Perl**, untuk huruf “P” selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi, sehingga sangat fleksibel dan banyak digunakan. Berikut merupakan langkah-langkah untuk melakukan instalasi xampp pada sistem operasi windows.

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

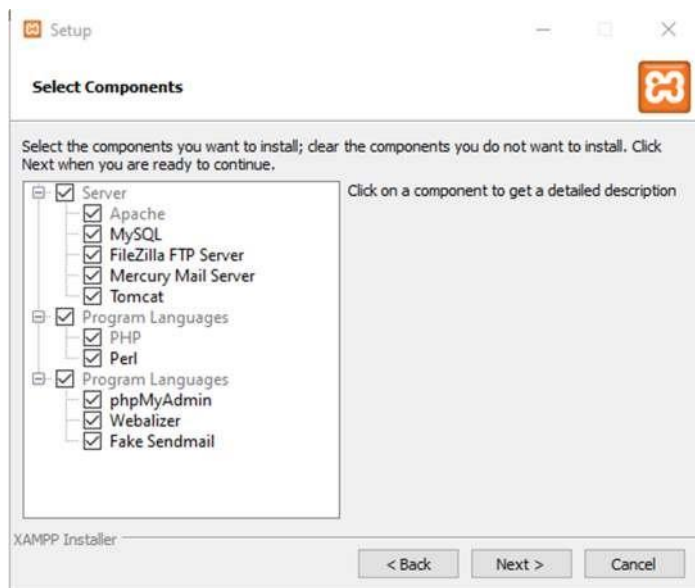


- b. Pilih file installer sesuai dengan spesifikasi laptop yang digunakan.
- c. Klik file xampp yang sudah berhasil di *download*, maka akan muncul tampilan seperti ini.

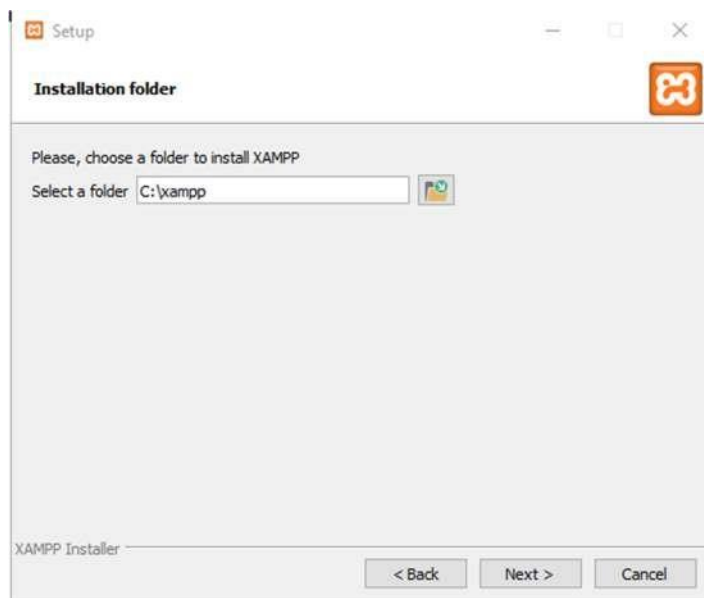




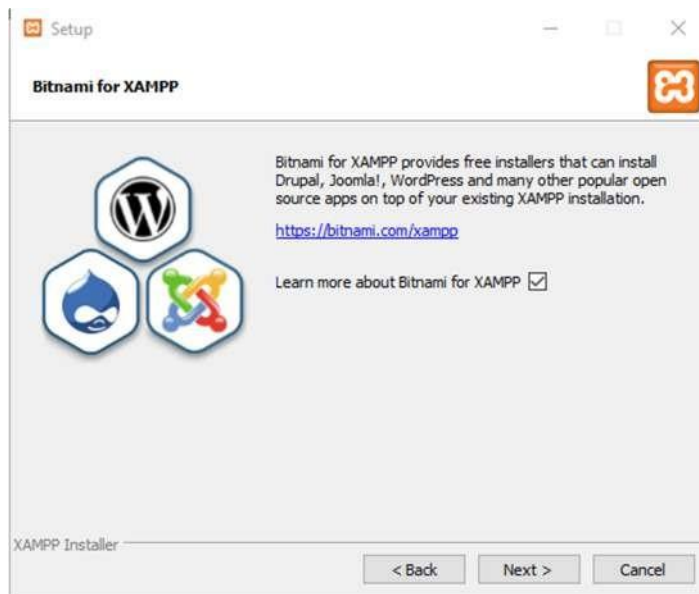
- d. Kemudian, pilih next untuk melanjutkan pada proses instalasi.



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



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



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



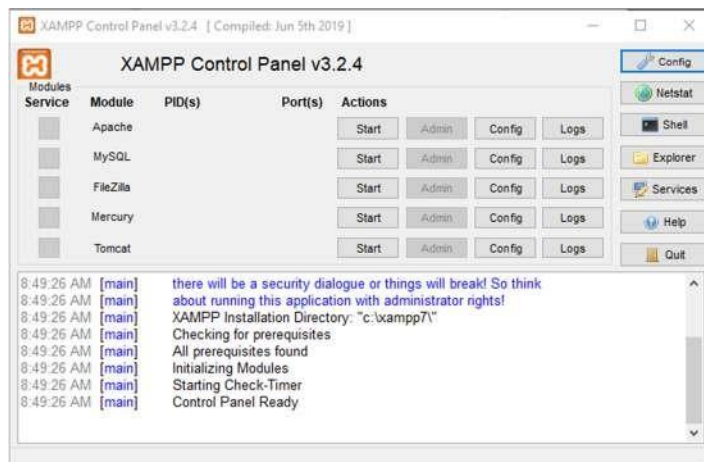
h. Kemudian tunggu hingga proses instalasi selesai.



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



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



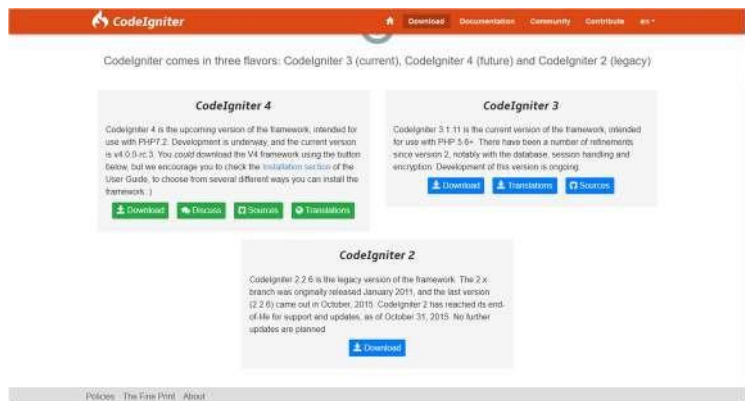
- k. Jika berhasil, maka tampilan utama *control panel* dari xampp seperti berikut.

### 4.3 CodeIgniter

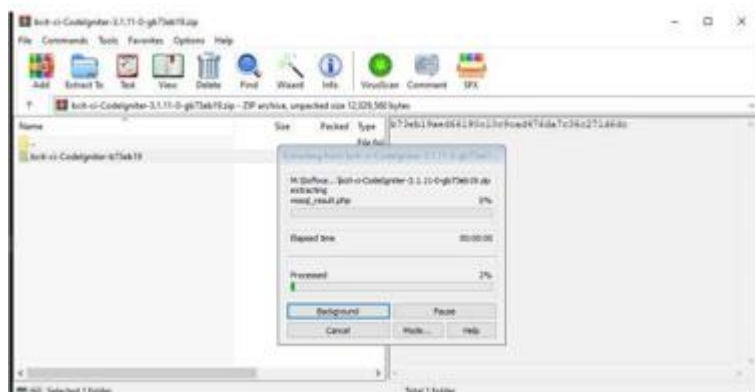
Codeigniter (CI) merupakan sebuah *web application framework* yang bersifat *open source* dimana digunakan untuk membangun aplikasi php dinamis. *Framework* itu sendiri merupakan abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh kode yang dibuat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

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

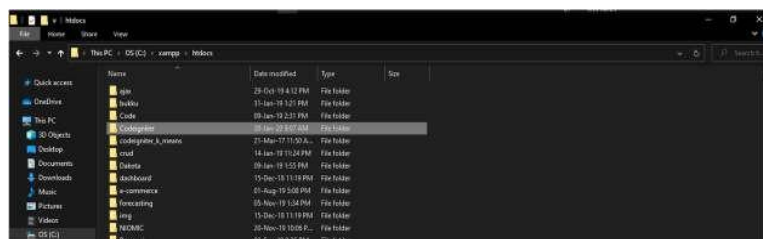
- a. Pertama, *download* terlebih dahulu codeigniter pada *website* resminya pada link di bawah ini <https://codeigniter.com/download>.



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



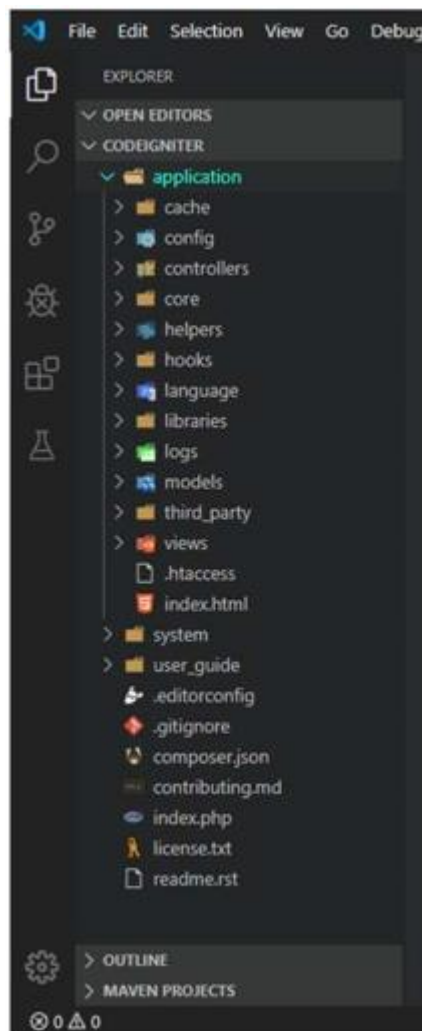
- c. Kemudian, *copy* file yang telah di ekstrak ke dalam direktori httdocs, seperti gambar berikut.



- 
- XAMPP Control Panel v3.2.2 [ Compiled: Nov 12th 2015 ]
- Modules**
- | Service                  | Module    | PID(s)       | Port(s) | Actions                 |
|--------------------------|-----------|--------------|---------|-------------------------|
| <input type="checkbox"/> | Apache    | 8352<br>2072 | 80, 443 | Stop Admin Config Logs  |
| <input type="checkbox"/> | MySQL     | 2616         |         | Stop Admin Config Logs  |
| <input type="checkbox"/> | FileZilla |              |         | Start Admin Config Logs |
| <input type="checkbox"/> | Mercury   |              |         | Start Admin Config Logs |
| <input type="checkbox"/> | Tomcat    |              |         | Start Admin Config Logs |
- Config** **Netstat** **Shell** **Explorer** **Services** **Help** **Quit**
- 9:15:13 AM [main] All prerequisites found  
 9:15:13 AM [main] Initializing Modules  
 9:15:13 AM [main] Starting Check-Timer  
 9:15:13 AM [main] Control Panel Ready  
 9:15:16 AM [Apache] Attempting to start Apache app...  
 9:15:16 AM [Apache] Status change detected: running  
 9:15:23 AM [mysql] Attempting to start MySQL app...  
 9:15:24 AM [mysql] Status change detected: running

- 
- Welcome to CodeIgniter!  
 The page you are looking at is being generated dynamically by CodeIgniter.  
 If you would like to edit this page you will find it located at  
  
 The corresponding controller for this page is found at  
  
 If you are exploring CodeIgniter for the very first time, you should start by reading the [User Guide](#).
- Your installation is 3.0.0 RC3. CodeIgniter Version: 3.0.0 RC3

- 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 dan 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.

Bila anda perhatikan pada struktur folder 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.

#### **4.3.1 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 memanage 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 <br>, 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');`

#### **4.3.2 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.

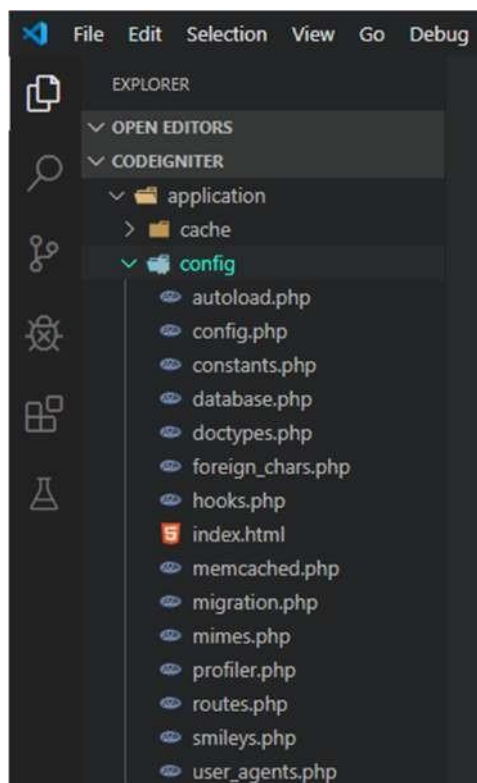
- ✓ Image Manipulation, digunakan untuk mereseize, membuat thumbnail, crop atau rotate gambar, serta watermark. Beberapa fungsi membutuhkan support PHP yang lebih baik, misalnya GD/GD2.
- ✓ Input and Security, digunakan untuk pre-process input data dari form dan URL serta untuk menghandle beberapa fungsi security (misalnya untuk menghadapi serangan XSS attacks).
- ✓ Language, library ini digunakan untuk meload beberapa bahasa yang berbeda.
- ✓ Loader, library ini secara otomatis diloat oleh system. Anda akan sering menggunakan library ini untuk meload view di dalam controller, selain itu juga untuk meload libraries.
- ✓ Output, otomatis diloat 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 diloat 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-RPX 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\_guidenya.

#### 4.4 Konfigurasi dasa 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 .....
7 AUTO-LOADER
8 .....
9 This file specifies which systems should be loaded by default.
10
11 In order to keep the framework as light-weight as possible only the
12 minimum system resources are loaded by default. For example
13 the database is not connected by default because you may not
14 need it. This file lets you to define which systems you would like
15 to load with every 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');
```

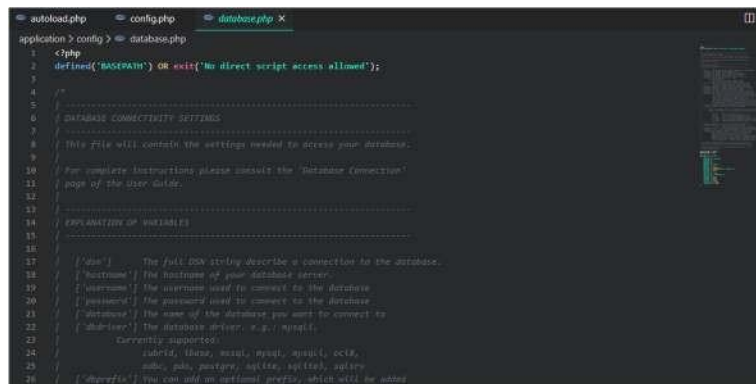
Kode tersebut dapat diartikan kita dapat meload library 'database' secara otomatis.

- 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.



- 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.

```

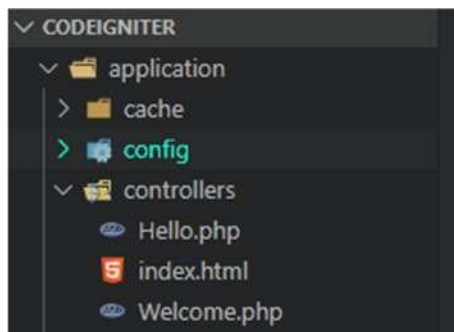
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost', //hostname
79     'username' => 'root', //username
80     'password' => '', //password
81     'database' => 'database_name', //nama database
82     'dbdriver' =>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



## 4.5 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.

### 4.5.1. Analisis Sistem Berjalan (*Current System*)

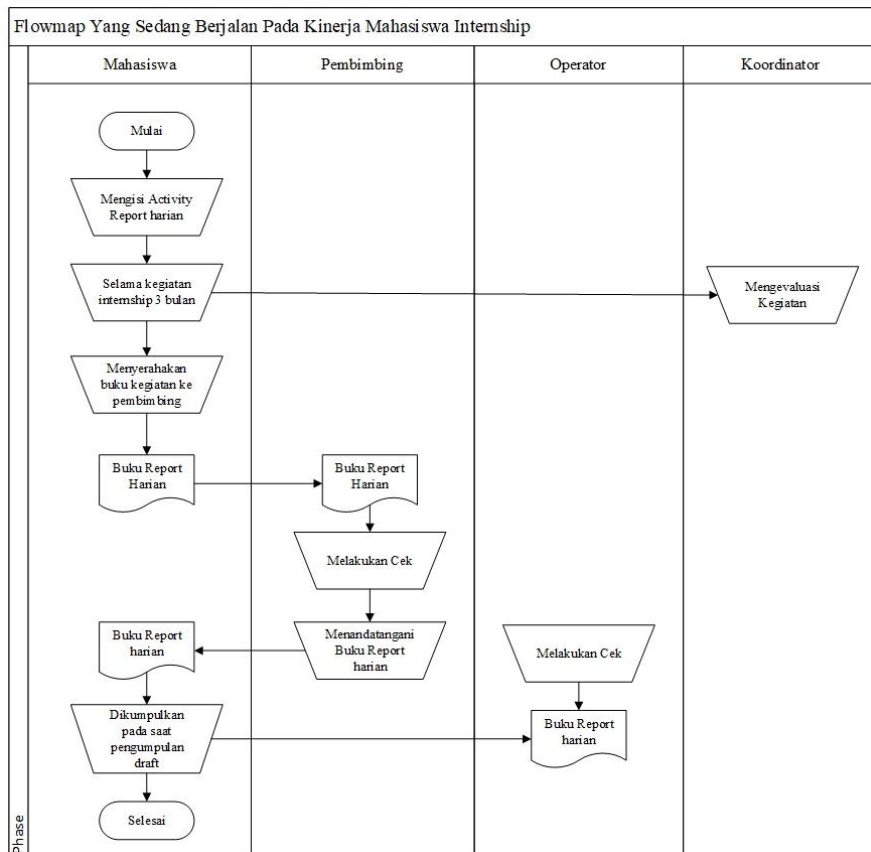
Sistem kegiatan mahasiswa *internship* di Politeknik Pos Indonesia saat ini sedang berjalan masih menggunakan cara manual lebih khususnya dibagian pengawasan kinerja atau laporan



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.

#### **4.5.1.1. 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 4.1 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*.

#### **4.5.1.2. Analisis Dokumen Yang Digunakan**

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

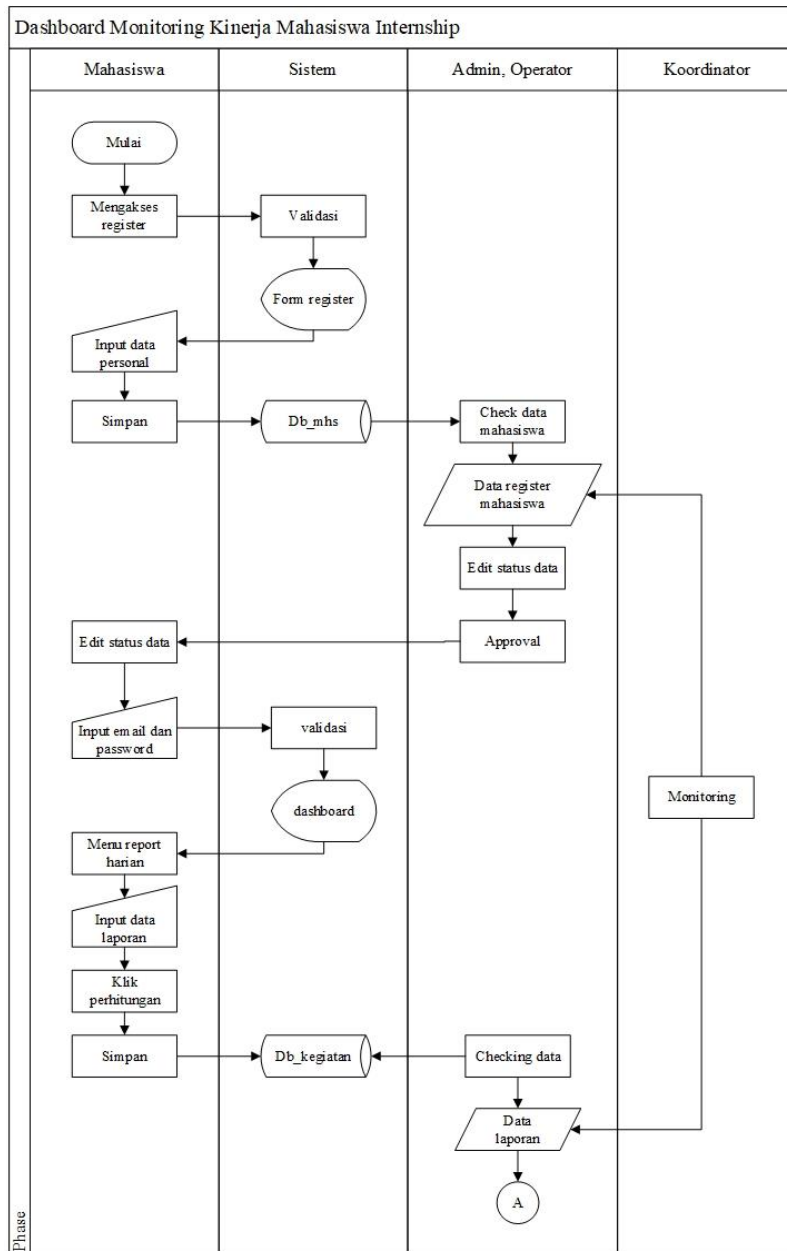
1. Formulir *report* harian , berupa form catatan aktivitas yang dilakukan dan menjadi sebagai buku *logbook*.
2. Dokumen data laporan, menjelaskan mengenai data-data laporan yang dikirimkan oleh mahasiswa berupa laporan atau materi ke pembimbing eksternal/mentor.

#### **4.5.2. Analisis Sistem yang akan dibangun**

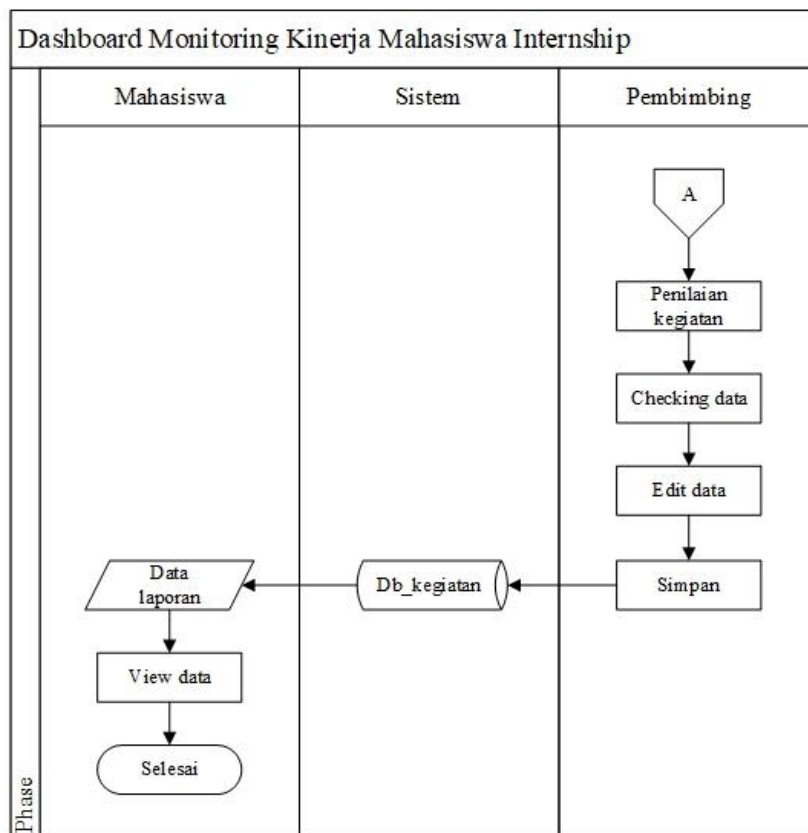
Dari analisis yang telah dilakukan maka akan dibangun sistem yang dapat memenuhi kebutuhan perusahaan/ perguruan tinggi dalam pengolahan data kinerja mahasiswa *internship*. Sistem ini sebelumnya dilakukan secara manual dan akan dibuat terkomputerisasi sehingga proses bisnis yang terjadi dapat lebih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

#### 4.5.2.1 Analisis Prosedur yang akan dibangun (*flowmap*)



Gambar 4.2 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.

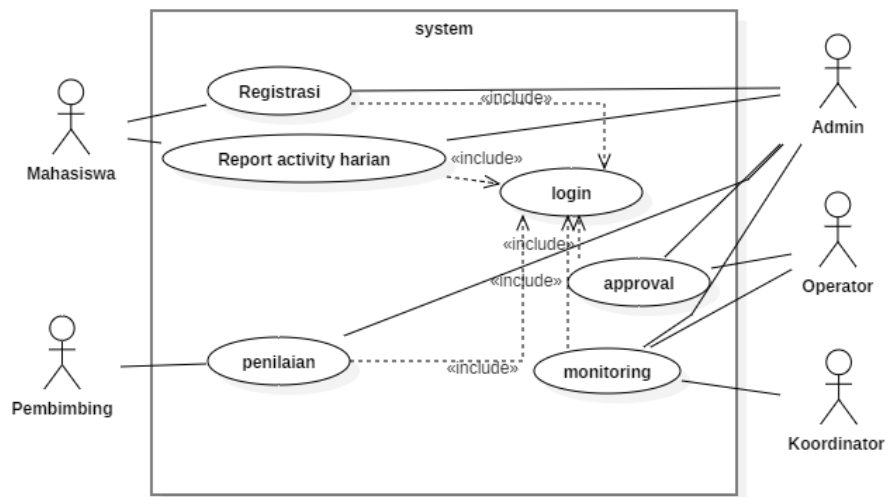
#### **4.5.2.2. Analisis Dokumen yang dibangun**

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. Definisi Aktor

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

*Tabel 5.1 Definisi Aktor*

No	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*

No	Usecase	Deskripsi
1.	Registrasi	Merupakan aktivitas untuk melakukan pendaftaran data diri sehingga terhubung dengan sistem.
2.	<i>Report activity</i> harian	Merupakan aktivitas untuk mencatat kegiatan yang dilakukan dalam bentuk sebagai laporan.



3.	Approval	Merupakan aktivitas untuk melakukan penerimaan data apabila sesuai dengan kebutuhan.
4.	Penilaian	Merupakan aktivitas untuk melakukan pengolahan informasi dan mengukur terkait pencapaian hasil kinerja yang telah dilakukan.
5.	Monitoring	Merupakan suatu kegiatan mengamati secara seksama suatu keadaan atau kondisi, termasuk juga perilaku atau kegiatan tertentu.
6.	Login	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	
Nomor	UC1
Nama	<i>Registrasi</i>
Tujuan	Merupakan aktivitas untuk melakukan pendaftaran data

	diri sehingga terhubung dengan sistem.
<b>Deskripsi</b>	
<b>Aktor</b>	Mahasiswa/i
<b>Skenario Utama</b>	
<b>Kondisi Awal</b>	Form sudah tersedia
<b>Kondisi Akhir</b>	Data registrasi mahasiswa telah berhasil ditambahkan.
<b>Main Flow Event</b>	
<b>Aksi aktor</b>	<b>Reaksi sistem</b>
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>
<b>Exceptional Flow of Event</b>	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	
Nomor	UC2
Nama	<i>Report activity harian</i>
Tujuan	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.
<i>Main Flow Event</i>	
Aksi aktor	Reaksi 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>

<b><i>Exceptional Flow of Event</i></b>	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*

<b>Identifikasi</b>	
<b>Nomor</b>	UC3
<b>Nama</b>	<i>Approval</i>
<b>Tujuan</b>	Merupakan aktivitas untuk melakukan penerimaan data apabila sesuai dengan kebutuhan.
<b>Deskripsi</b>	
<b>Aktor</b>	Admin, Operator
<b>Skenario Utama</b>	
<b>Kondisi Awal</b>	Form sudah tersedia
<b>Kondisi Akhir</b>	Data mahasiswa telah berhasil dikelola <i>(insert, update, delete, view)</i> .
<b><i>Main Flow Event</i></b>	
<b>Aksi aktor</b>	<b>Reaksi sistem</b>

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>
<b><i>Exceptional Flow of Event</i></b>	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*

<b>Identifikasi</b>	
<b>Nomor</b>	UC4
<b>Nama</b>	<i>Penilaian</i>
<b>Tujuan</b>	Merupakan aktivitas untuk melakukan pengolahan informasi dan mengukur terkait pencapaian hasil kinerja yang telah dilakukan.
<b>Deskripsi</b>	

<b>Aktor</b>	Pembimbing
<b>Skenario Utama</b>	
<b>Kondisi Awal</b>	Form sudah tersedia
<b>Kondisi Akhir</b>	Data laporan telah berhasil dikelola ( <i>insert, update, delete, view</i> ).
<b><i>Main Flow Event</i></b>	
<b>Aksi aktor</b>	<b>Reaksi sistem</b>
1. Aktor memilih menu penilaian, selanjutnya mengklik <i>detail</i> data.	2. Sistem akan menampilkan form isian data laporan mahasiswa per-progress
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>
<b><i>Exceptional Flow of Event</i></b>	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

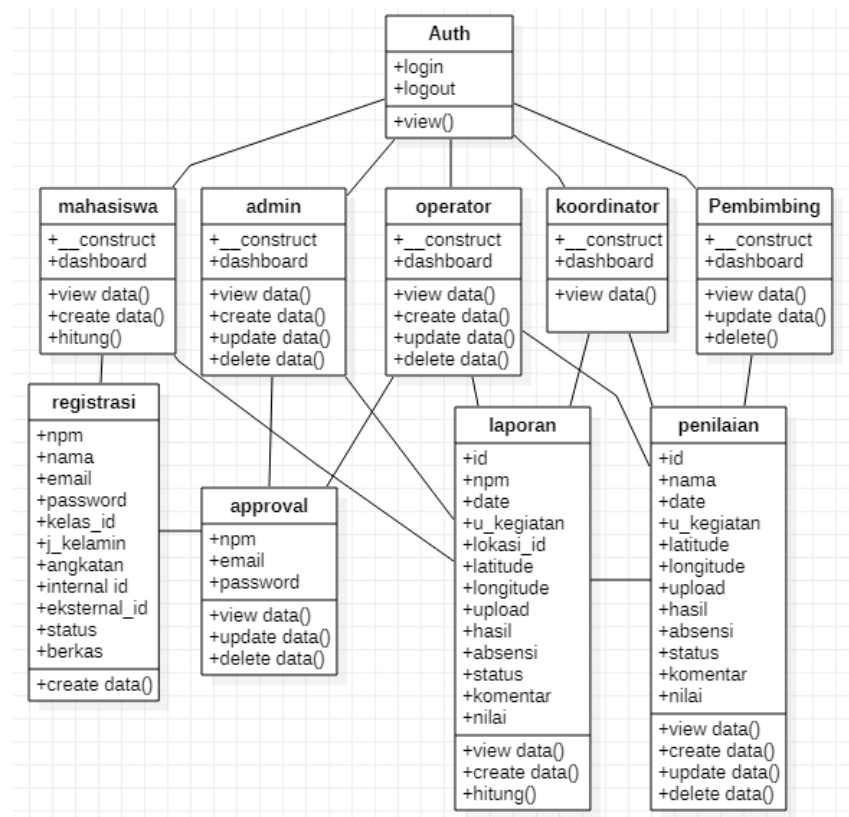
Identifikasi	
Nomor	UC5
Nama	<i>Monitoring</i>
Tujuan	Merupakan aktivitas untuk melakukan pengolahan informasi dan mengukur terkait pencapaian hasil kinerja yang telah dilakukan.
Deskripsi	
Aktor	Koordinator
Skenario Utama	
Kondisi Awal	Form sudah tersedia
Kondisi Akhir	Semua data yang dikelola. ( <i>view</i> ).
<i>Main Flow Event</i>	
Aksi aktor	Reaksi 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>

<b><i>Exceptional Flow of Event</i></b>	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

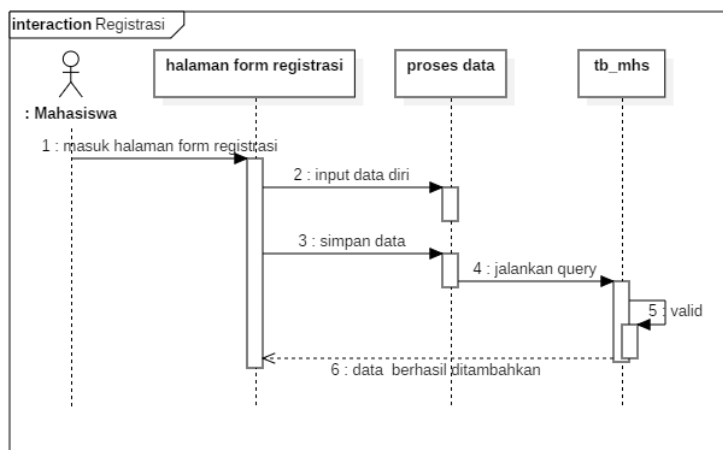


Gambar 5.5 Class Diagram

### 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



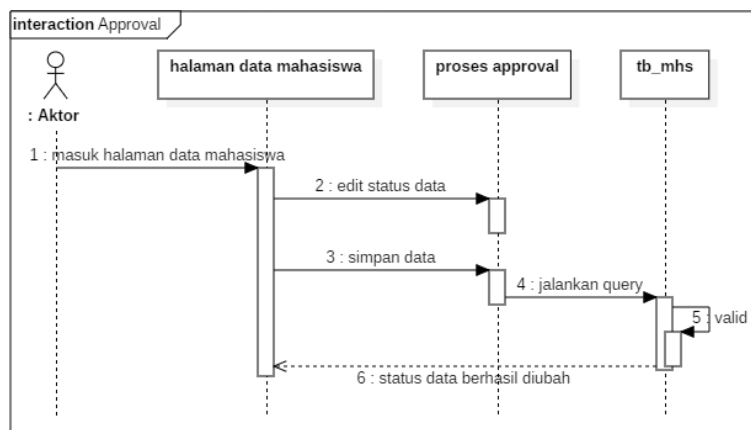
Gambar 5.6 Sequence Diagram Registrasi

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 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



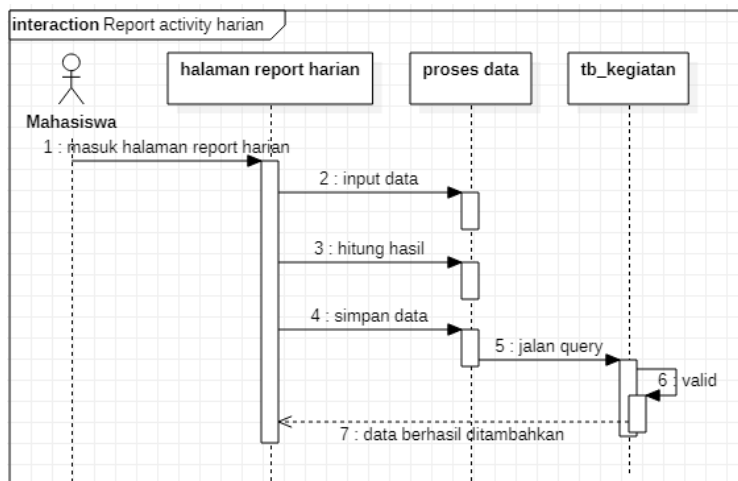
Gambar 5.7 Sequence Diagram Approval

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*



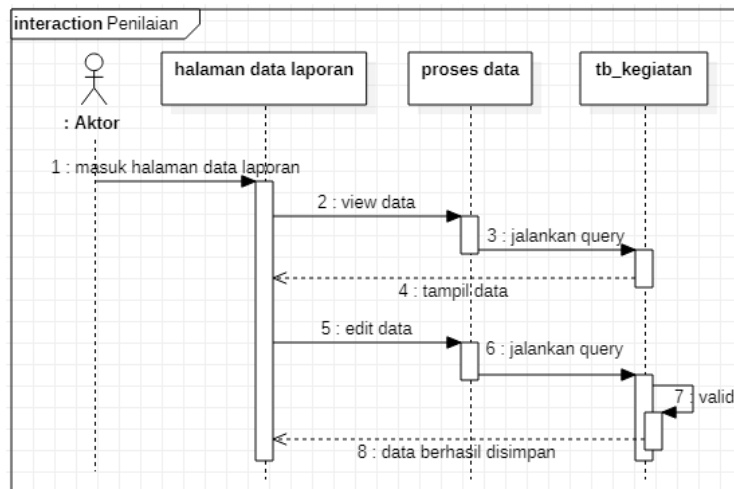
Gambar 5.8 *Sequence Diagram Report Activity Harian*

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



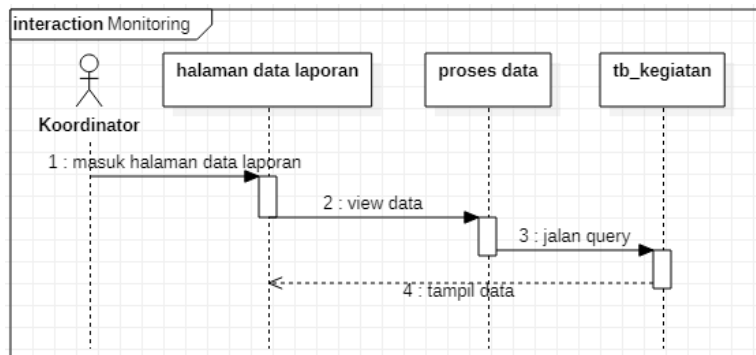
Gambar 5.9 Sequence Diagram Penilaian

Keterangan :

Pada *sequence* diagram ini menjelaskan proses penilaian terhadap kinerja mahasiswa dalam konteks 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

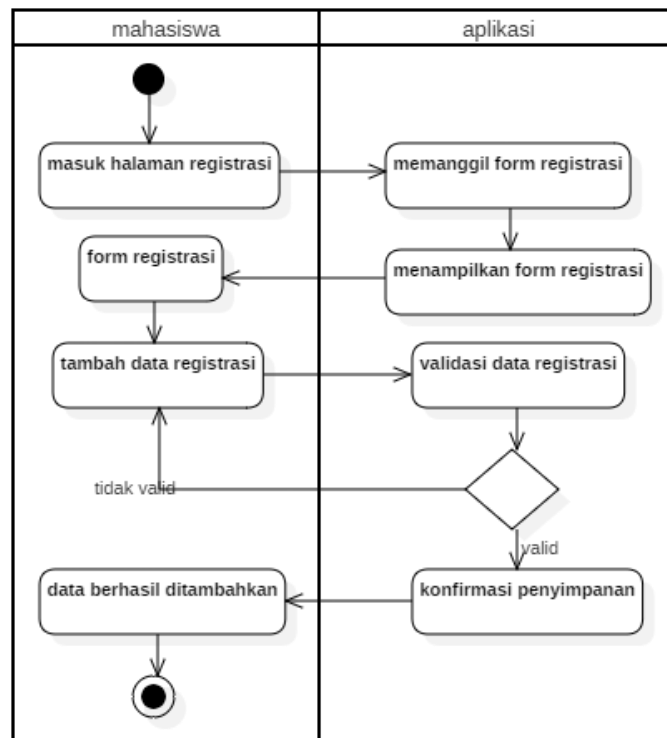
Keterangan :

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



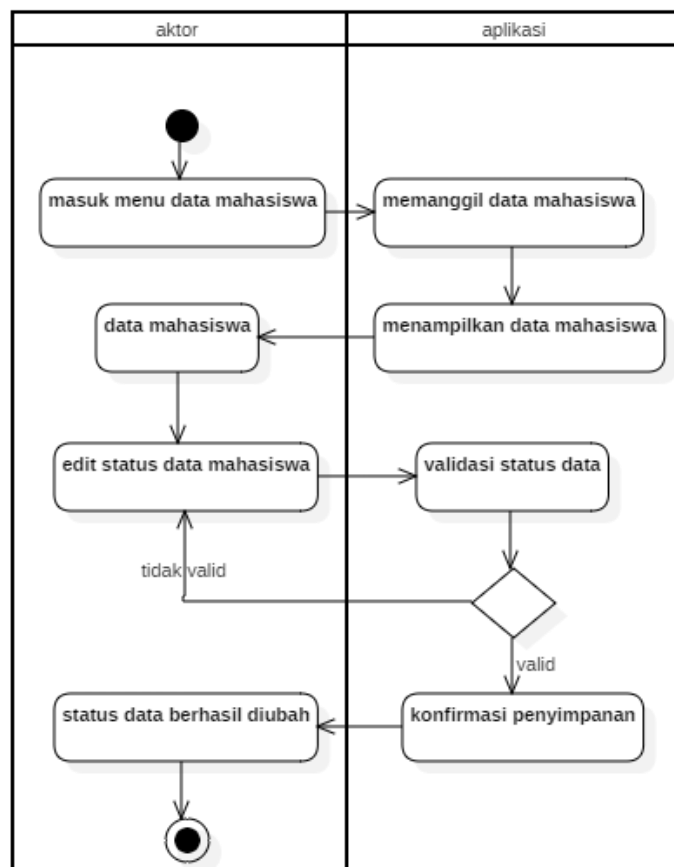
Gambar 5.11 Activity Diagram Registrasi

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



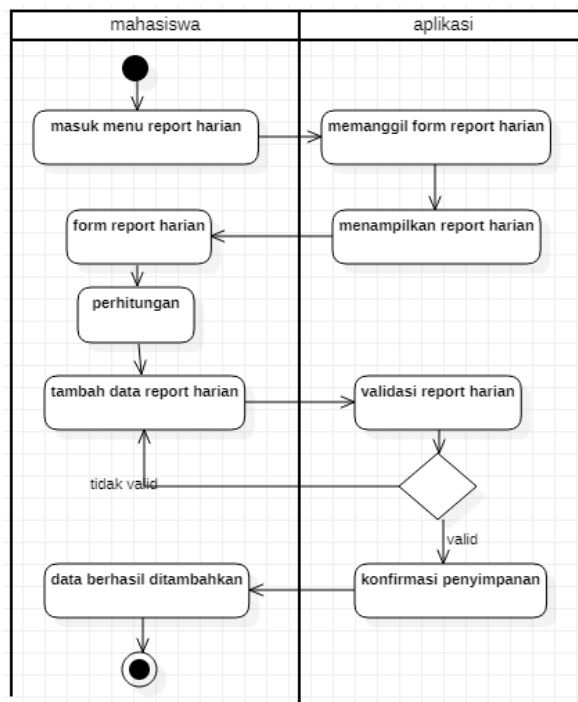
Gambar 5.12 Activity Diagram Approval



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

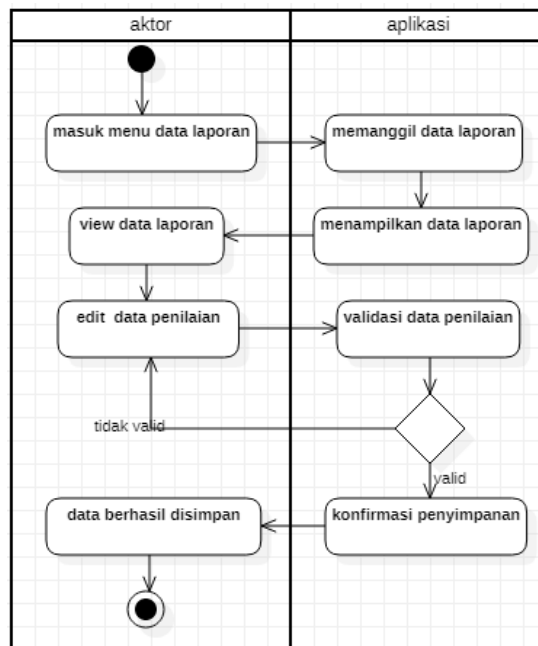


Gambar 5.13 Activity Diagram Report Activity Harian

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

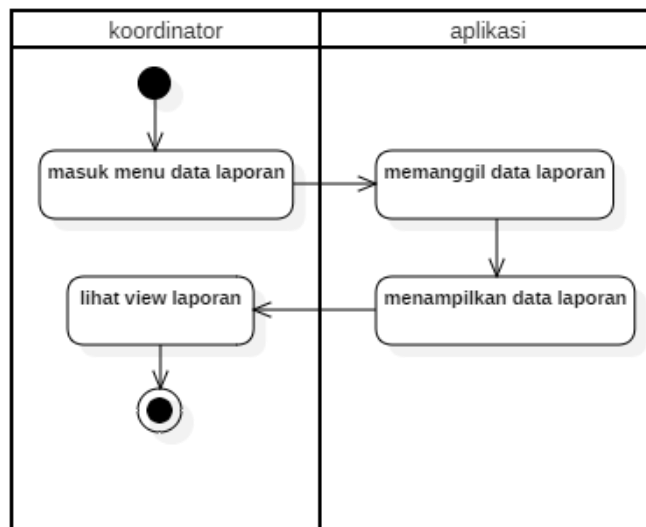


Gambar 5.14 Activity Diagram Penilaian

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

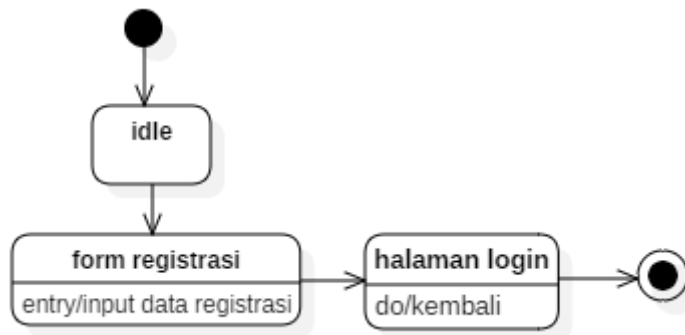
Keterangan :

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

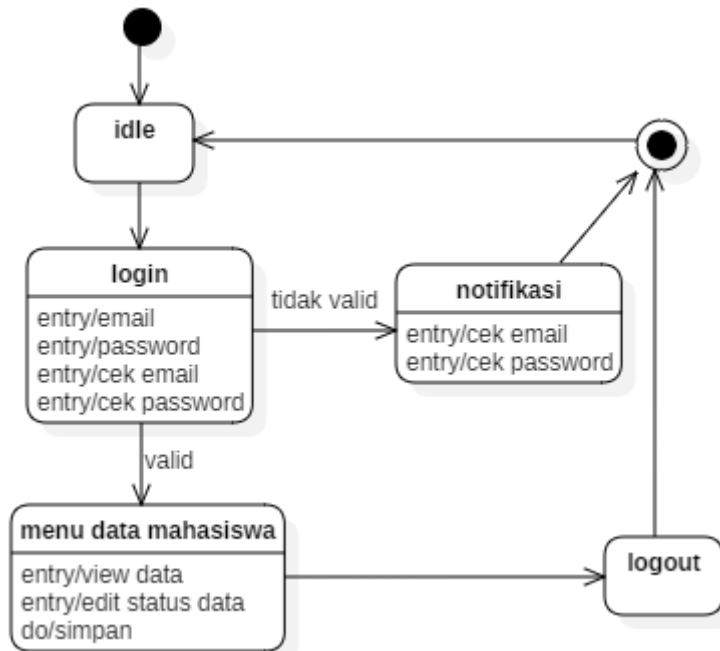


*Gambar 5.16 Statechart Diagram Registrasi*

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



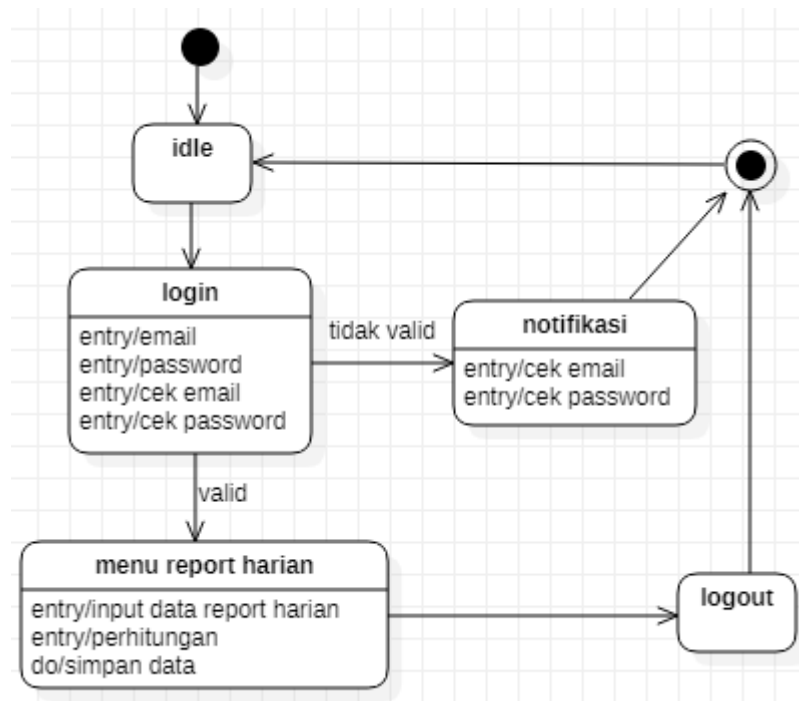
Gambar 5.17 Statechart Diagram Approval

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*



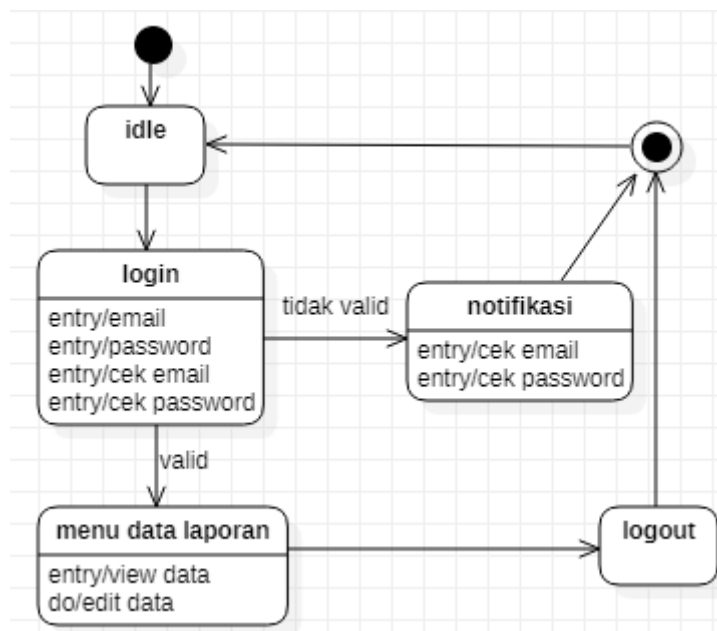
Gambar 5.18 *Statechart Diagram Report Activity Harian*

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

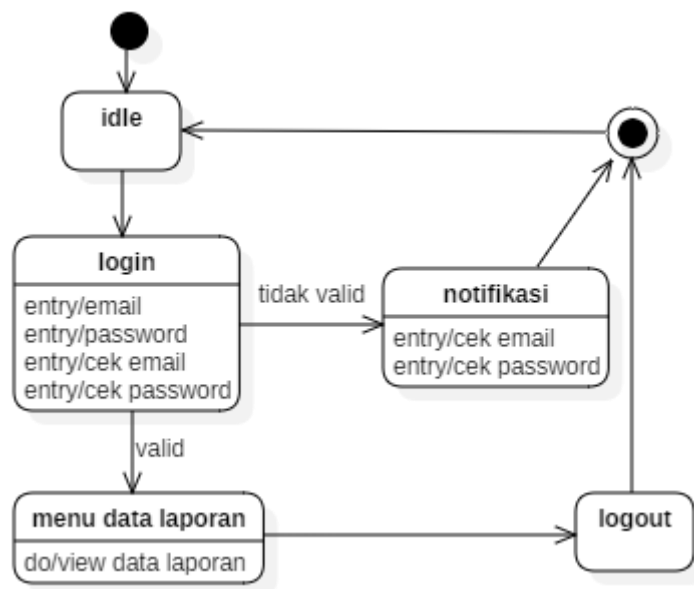
Keterangan :

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



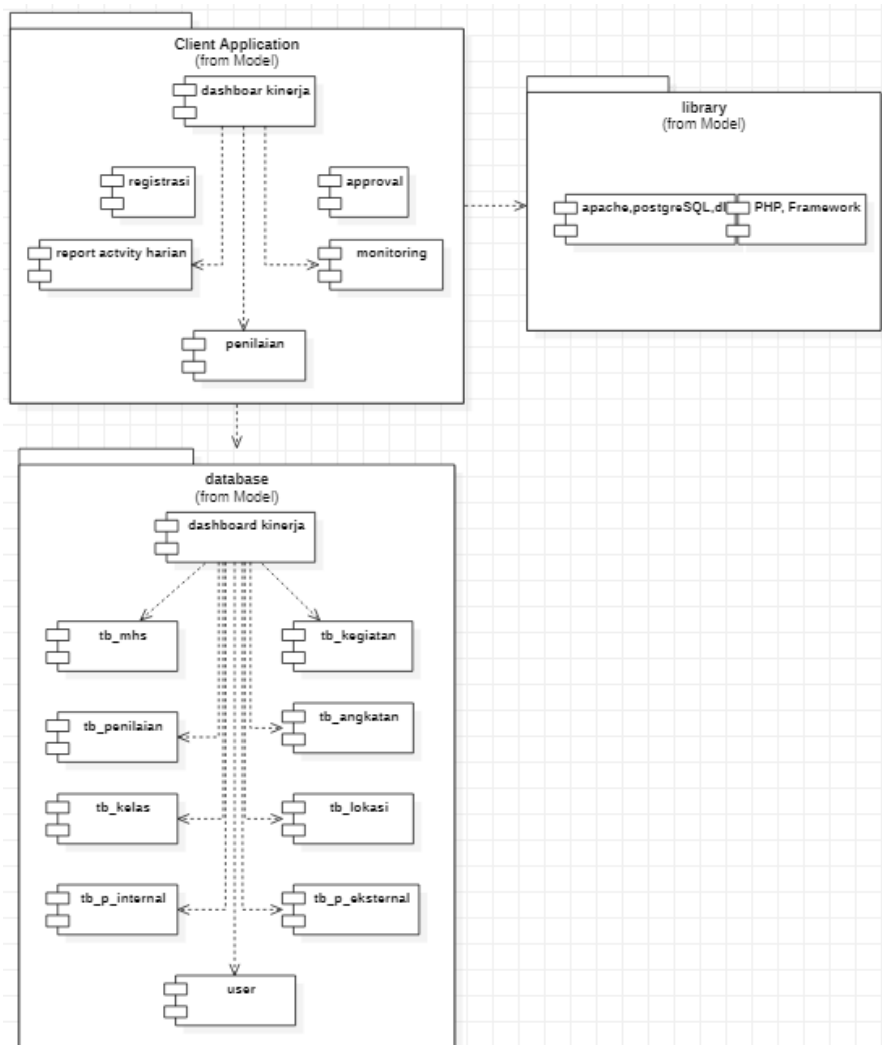
Gambar 5.20 Statechart Diagram Monitoring

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

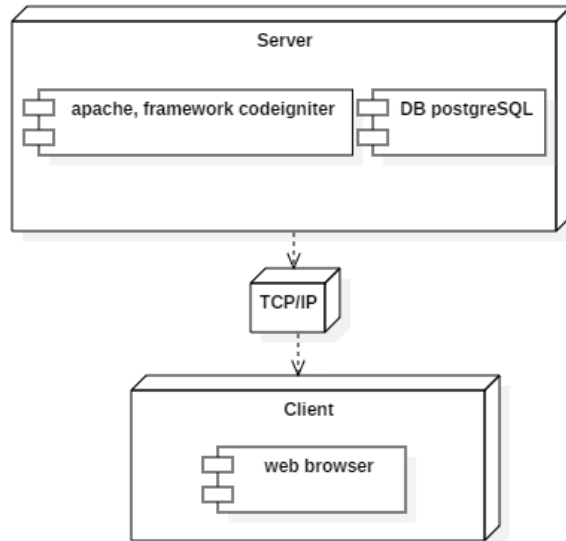
maka aktor dapat melakukan aski seperti *check*-ing data dengan menekan pada *button* detail dan sistem akan menampilkan data yang lengkap. Setelah melakukan *check*-ing data laporan maka aktor dapat melakukan *logout*.

### 5.3.5 Component Diagram



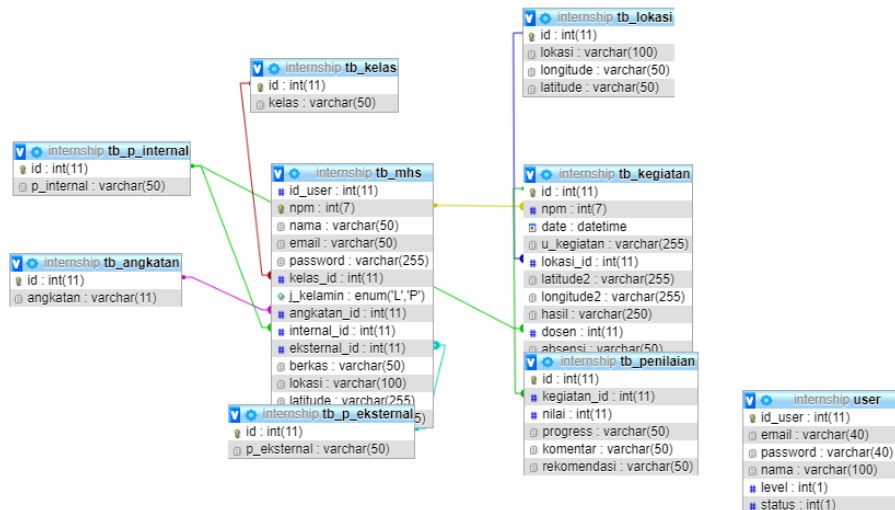
Gambar 5.21 Component Diagram

### 5.3.6 Deployment Diagram



Gambar 5.22 Deployment Diagram

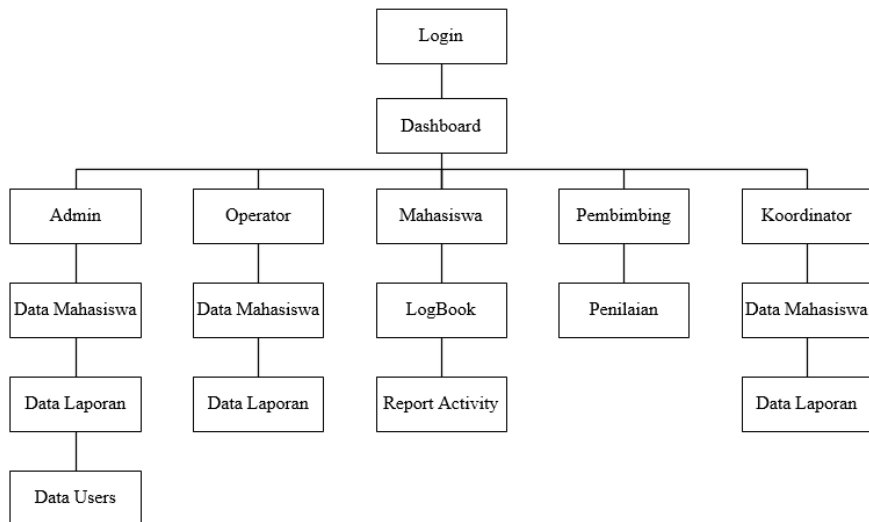
### 5.3.7 Perancangan Database



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*




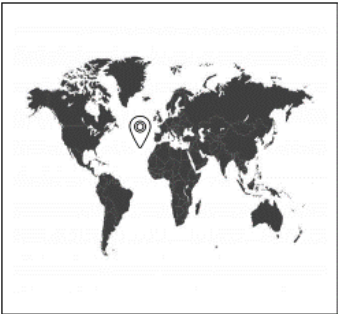
Sign in to start your session

[Register new a membership](#)

*Gambar 5.25 Login*

#### 2. Form *Registrasi*

 REGISTRASI




☐ Laki-laki ☐ perempuan

*Gambar 5.26 Form Registrasi*

### 3. Halaman *Approval* Data Mahasiswa

myINTERNSHIP

admin@gmail.com

  
Admin

Dashboard

Data Mahasiswa

Data Register internship

2 > Mahasiswa


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

Gambar 5.27 Halaman *Approval* Data Mahasiswa

### 4. Form Report Activity Harian

myINTERNSHIP

mahasiswa@gmail.com

  
Fulan 01

Dashboard


Report Harian

Report Harian kegiatan

2 > Mahasiswa

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](#)

  
Dosen A

Dashboard



Data Mahasiswa

Data Laporan

Data Laporan

internship


> Mahasiswa


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

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
1.	Sistem Operasi	:	<i>Microsoft Windows 10 Profesional 64-bit</i>
2.	Bahasa Pemrograman	:	<i>PHP dengan Framework CI (Codeigniter)</i>
3.	Database	:	<i>PostgreSQL</i>
4.	Perangkat Lunak	:	<i>Visual Studio Code Microsoft Visio 2013 StarUML 3.1.0</i>

*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
1.	<i>Processor</i>	:	Intel® core™i3

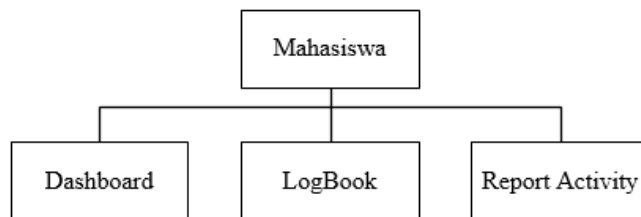


2.	<i>Memory</i>	:	4 GB
3.	<i>Monitor</i>	:	LCD 14,1 Inchi
4.	<i>Mouse dan keyboard</i>	:	<i>Standard</i>

*Tabel 5.9 Perangkat Keras*

### 5.5 Pemetaan Struktur Diagram User / Aktor Sistem

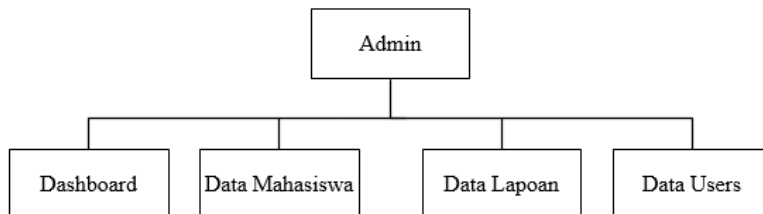
Dibawah ini merupakan pemetaan struktur diagram menu



untuk mahasiswa.

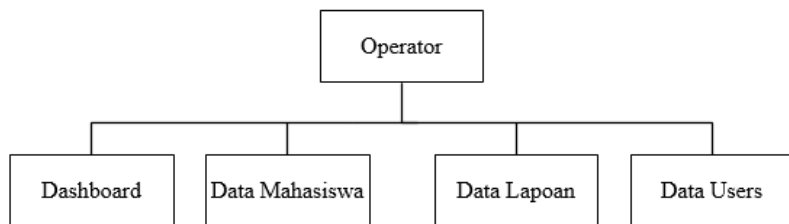
*Gambar 5.31 Pemetaan Struktur Diagram Mahasiswa*

Dibawah ini merupakan pemetaan struktur diagram menu untuk admin.



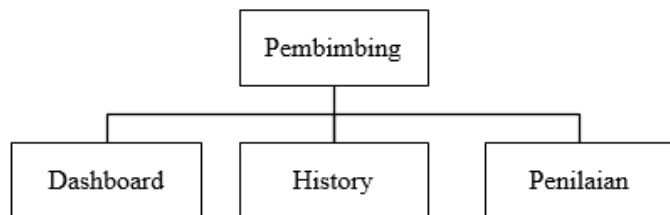
*Gambar 5.32 Pemetaan Struktur Diagram Admin*

Dibawah ini merupakan pemetaan struktur diagram menu untuk operator.



*Gambar 5.33 Pemetaan Struktur Diagram Operator*

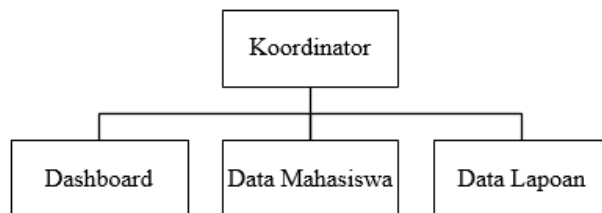
Dibawah ini merupakan pemetaan struktur diagram menu untuk pembimbing.



*Gambar 5.34 Pemetaan Struktur Diagram*

*Pembimbing*

Dibawah ini merupakan pemetaan struktur diagram menu



untuk koordinator

*Gambar 5.35 Pemetaan Struktur Diagram Koordinator*









































































