

**RANCANG BANGUN APLIKASI SISTEM INFORMASI GEOGRAFIS  
LOKASI DAN PENDAFTARAN VAKSIN COVID-19 BERBASIS WEB  
DI KOTA CIMAHI**

**TUGAS AKHIR**

Diajukan sebagai salah satu syarat untuk kelulusan  
Jenjang Strata Satu(S1)  
pada Program Studi Teknik Informatika

Oleh:

**SATIA NUR MAULIDIN**  
**362031005**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER  
INDONESIA MANDIRI  
BANDUNG  
2022**

## LEMBAR PENGESAHAN

### RANCANG BANGUN APLIKASI SISTEM INFORMASI GEOGRAFIS LOKASI DAN PENDAFTARAN VAKSIN COVID-19 BERBASIS WEB DI KOTA CIMAHI

Oleh:

**Satia Nur Maulidin**  
**362031005**

Tugas Akhir ini telah diterima dan disahkan  
untuk memenuhi persyaratan mencapai gelar

STRATA SATU TEKNIK INFORMATIKA

Pada

PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER  
INDONESIA MANDIRI

Bandung, 09 Februari 2022  
Disahkan oleh

Ketua Program Studi Teknik  
Informatika,

Chalifa Chazar, S.T., M.T.  
NIDN: 0421098704

Dosen Pembimbing,

Dr. Pahlawan Sagala  
NIDN: 0422065802

## LEMBAR PERSETUJUAN REVISI TUGAS AKHIR

### RANCANG BANGUN APLIKASI SISTEM INFORMASI GEOGRAFIS LOKASI DAN PENDAFTARAN VAKSIN COVID-19 BERBASIS WEB DI KOTA CIMAHI

Oleh:  
**Satia Nur Maulidin**  
**362031005**

Telah melakukan sidang tugas akhir pada hari Rabu, tanggal 02 Februari 2022 dan telah melakukan revisi sesuai dengan masukan pada saat sidang tugas akhir.

Bandung, 09 Februari 2022

Menyetujui

No	Nama	Penguji	Tanda Tangan
1	Dr. Pahlawan Sagala	Pembimbing	
2	Hendra Gunawan, S.T., M.Kom.	Penguji 1	
3	Dede Suryana, Ir., M.Si.	Penguji 2	

Mengetahui

Ketua Program Studi Teknik Informatika,

**Chalifa Chazar, S.T., M.T.**  
NIDN: 0421098704

## **SURAT PERNYATAAN**

Dengan ini saya menyatakan bahwa:

- (1) Tugas akhir ini adalah asli diajukan untuk mendapatkan gelar akademik.
- (2) Tugas akhir ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- (3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya .

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, 09 Februari 2022

Yang Membuat Pernyataan

SATIA NUR MAULIDIN

362031005

## ABSTRAK

Sejalan dengan perkembangan teknologi informasi berbasis georeferensi serta kebutuhan informasi mengenai lokasi disuatu area, maka pemanfaatan sistem informasi geografis sangat dibutuhkan terutama saat kondisi pandemi Covid-19 saat ini, banyak penggiat kesehatan menggunakan sistem informasi geografis untuk melihat sebaran pandemi di suatu daerah. Rancang bangun sistem informasi geografis lokasi dan pendaftaran vaksin di Kota Cimahi. Diharapkan data mengenai peta wilayah daerah, jumlah sasaran bisa ditampilkan secara bersama. Sehingga, apabila suatu wilayah mempunyai jumlah sasaran target yang kurang dengan SIG bisa diputuskan dengan wilayah yang paling terdekat untuk pelaksanaan dan pendaftaran vaksin. Untuk mengatasi permasalahan tersebut, maka dibutuhkan suatu aplikasi baru. Perancangan dan pembuatan aplikasi pengelolaan lokasi dan pendaftaran vaksin yang baru ini memakai metode *waterfall* serta menggunakan program aplikasi PHP dan My SQL untuk databasenya. Hal lain yang juga perlu dilakukan dalam merancang sistem yang baru adalah merancang *Unified Modeling Language* dan *Entity Relationship Diagram*. Hasil dari pembuatan program ini adalah memudahkan masyarakat untuk menemukan lokasi dan pendaftaran vaksinasi covid-19 tanpa harus kesulitan dengan mencari alamat fisik.

Kata Kunci : Sistem Informasi Geografis, Pemrograman Web, Rekayasa Perangkat Lunak

## ABSTRACT

*In line with the development of georeferenced-based information technology and the need for information regarding locations in an area, the use of geographic information systems is needed, especially during the current Covid-19 pandemic, many health activists use geographic information systems to see the spread of the pandemic in an area. Design and build a geographic information system for the location and registration of vaccines in Cimahi City. It is hoped that data on regional maps and the number of targets can be displayed together. So, if an area has a number of targets that are lacking with GIS, it can be decided with the closest area for vaccine implementation and registration. To overcome these problems, a new application is needed. The design and manufacture of this new vaccine site management and registration application uses the waterfall method and uses PHP and My SQL application programs for the database. Another thing that also needs to be done in designing a new system is to design a Unified Modeling Language and Entity Relationship Diagram. The result of making this program is to make it easier for the public to find information about the location and registration of vaccines.*

*Keywords: Geographic Informations System, Web Programming, Software Engineering*

## **KATA PENGANTAR**

*Assalamu'alaikum Wr.Wb.*

*Alhamdulillahirabbil'alamin* Segala Puji dan syukur penulis panjatkan kepada Allah SWT, karena atas Rahmat dan Hidayah-Nya sehingga Penulis dapat menyelesaikan pembuatan laporan tugas akhir ini dengan baik. Sholawat serta salam mudah-mudahan tercurah limpah kepada pemimpin kita, Nabi Muhammad SAW, kepada sahabatnya, keluarganya dan mudah-mudahan sampai kepada kita. Pembuatan laporan tugas akhir ini ditujukan untuk memenuhi dan melengkapi persyaratan Akademik pada jurusan Teknik Informatika di STMIK Indonesia Mandiri dengan judul : “RANCANG BANGUN APLIKASI SISTEM INFORMASI GEOGRAFIS LOKASI DAN PENDAFTARAN VAKSIN COVID-19 BERBASIS WEB DI KOTA CIMAHI”.

Penulis menyadari banyak kesulitan dan hambatan yang dihadapi dalam proses penyusunan skripsi ini, namun berkat kerja keras, do'a, dukungan dan bantuan dari berbagai pihak akhirnya laporan ini dapat diselesaikan. Oleh karena itu dengan segala kerendahan hati penulis menyadari bahwa penyusunan dan penulisan skripsi ini masih jauh dari kesempurnaan baik ditinjau dari kalimat maupun materinya. Hal ini mengingat keterbatasan kemampuan dan pengetahuan penulis. Walaupun demikian usaha yang maksimal telah dilakukan dalam penyelesaian skripsi ini agar dapat memenuhi harapan. Semoga skripsi ini dapat

bermanfaat, khususnya bagi penulis, umumnya bagi pembaca sehingga kritik dan saran demi perbaikan sangat diharapkan.

Sebagai rasa hormat, cinta dan kasih sayang, penulis persembahkan Skripsi ini untuk Ibu dan Ayah yang telah memberikan banyak Do'a dan semangat untuk penulis dan satu satunya alasan bagi penulis untuk dapat menyelesaikan Skripsi ini dengan tepat waktu, Dalam kesempatan ini penulis menyampaikan ungkapan rasa syukur dan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan segala kemudahan dan kekuatan hingga terselesaikannya skripsi ini.
2. Bapak Dr. Chairuddin., M.M., M.T selaku Ketua Sekolah Tinggi Manajemen Informatika Indonesia Mandiri (STMIK-IM).
3. Bapak Dr. Pahlawan Sagala pembimbing Tugas Akhir, yang telah memberikan dorongan dan arahan kepada penulis selama proses penyusunan laporan tugas akhir ini
4. Ibu Chalifa Chazar., ST., M.T., selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika Indonesia Mandiri (STMIK-IM).
5. Bapak/Ibu dosen dan staf yang telah mendidik serta membantu dalam proses pembelajaran selama penulis menjadi mahasiswa di Sekolah Tinggi Manajemen Informatika Indonesia Mandiri (STMIK-IM).



6. Ayah, Ibu, Adik, Istriku, dan Anakku tercinta yang telah mendo'akan, memberi dukungan dan bantuannya selama pembuatan tugas akhir ini.
7. Teman-teman dan *Tim Online Onboarding* yang telah memberikan dukungan dan semangat pada penulis selama pembuatan tugas akhir ini.
8. Dan semua pihak yang telah banyak membantu, yang namanya tidak dapat penulis sebutkan satu persatu.

Semoga Allah SWT senantiasa memberikan imbalan yang berlipat ganda bagi amal shaleh yang telah dilakukan umat-Nya. Semoga laporan tugas akhir ini dapat bermanfaat bagi semua pihak yang memerlukan. Amien.

Bandung, 09 Februari 2022  
Penulis

Satia Nur Maulidin  
362031005

## DAFTAR ISI

### LEMBAR PENGESAHAN

<b>SURAT PERNYATAAN .....</b>	<b>i</b>
<b>ABSTRAK .....</b>	<b>ii</b>
<b><i>ABSTRACT</i> .....</b>	<b>iii</b>
<b>KATA PENGANTAR .....</b>	<b>iv</b>
<b>DAFTAR ISI .....</b>	<b>vii</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>DAFTAR GAMBAR .....</b>	<b>xiii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah .....	3
1.3 Tujuan Penelitian .....	3
1.4 Batasan Masalah.....	3
1.5 Metode Penelitian.....	4
1.5.1 Metode Pengumpulan Data .....	4
1.5.2 Metode Pengembangan Sistem .....	5
1.6 Sistematika Penulisan .....	7
<b>BAB II LANDASAN TEORI .....</b>	<b>9</b>
2.1 Konsep Dasar Sistem .....	9
2.1.1 Pengertian Sistem .....	9

2.1.2 Karakteristik Sistem .....	10
2.2 Konsep Dasar Informasi.....	11
2.2.1 Pengertian Informasi .....	11
2.2.2 Kualitas Informasi .....	11
2.2.3 Jenis – jenis Informasi .....	13
2.3 Konsep Dasar Sistem Informasi.....	14
2.3.1 Pengertian Sistem Informasi .....	14
2.3.2 Komponen Sistem Informasi .....	15
2.4 Sistem Informasi Geografis .....	17
2.4.1 Jenis Data Masukan Sistem Informasi Geografis .....	18
2.5 Metode Pengembangan Sistem .....	19
2.6 <i>Database</i> (Basis Data).....	21
2.7 Pengertian Diagram <i>Flowchart</i> .....	21
2.8 <i>Entity Relationship Diagram</i> (ERD).....	22
2.9 OOP ( <i>Object Oriented Programming</i> ).....	24
2.9.1 Sejarah OOP ( <i>Object Oriented Programming</i> ).....	25
2.9.2 Istilah-istilah OOP ( <i>Object Oriented Programming</i> ) .....	26
2.10 <i>Unified Modelling Language</i> (UML).....	29
2.11 <i>Software</i> Pendukung .....	35
2.11.1 <i>Perl Hypertext Preprocessor</i> (PHP) .....	35
2.11.2 <i>Hypertext Markup Language</i> (HTML) .....	37

2.11.3 <i>Cascading Style Sheet (CSS)</i> .....	37
2.11.4 <i>Framework CodeIgniter</i> .....	38
2.11.5 <i>Mapbox</i> .....	41
2.11.5.1 <i>Mapbox Direction API</i> .....	41
2.11.5.2 <i>Mapbox Map API</i> .....	42
2.11.6 <i>Model View Controller (MVC)</i> .....	43
2.12 <i>Coronavirus</i> .....	44
2.12.1 <i>Pengertian Coronavirus Disease 2019 (Covid – 19)</i> .....	44
<b>BAB III ANALISIS MASALAH DAN PERANCANGAN PROGRAM ..</b>	<b>46</b>
3.1 <i>Analisis Sistem</i> .....	46
3.1.1 <i>Studi Literatur</i> .....	46
3.1.2 <i>Analisis Sistem yang Sedang Berjalan</i> .....	50
3.1.3 <i>Evaluasi Sistem yang Berjalan</i> .....	51
3.1.4 <i>Solusi yang Diusulkan</i> .....	52
3.1.5 <i>Gambaran Umum Sistem yang Diusulkan</i> .....	54
3.1.6 <i>Analisis Kebutuhan Sistem</i> .....	54
3.2 <i>Perancangan Sistem</i> .....	55
3.2.1 <i>Tujuan Perancangan Sistem</i> .....	56
3.2.2 <i>Perancangan Prosedur yang Diusulkan</i> .....	56
3.2.3 <i>Use Case Diagram</i> .....	57
3.2.4 <i>Activity Diagram</i> .....	63

3.2.5 <i>Sequance Diagram</i> .....	71
3.2.6 <i>Class Diagram</i> .....	77
3.2.7 ERD ( <i>Entity Relationship Diagram</i> ) .....	77
3.2.8 Perancangan Struktur <i>Database</i> .....	79
3.2.9 Perancangan Antar Muka .....	82
<b>BAB IV TESTING DAN IMPLEMENTASI SISTEM.....</b>	<b>87</b>
4.1 Implementasi Sistem .....	87
4.1.1 Tujuan Implementasi Sistem .....	87
4.1.2 Implementasi Perangkat Lunak .....	87
4.1.3 Implementasi Perangkat Keras .....	88
4.1.4 Implementasi Antar Muka .....	88
4.2 Pengujian Sistem .....	97
4.2.1 Rencana Pengujian .....	98
<b>BAB V KESIMPULAN DAN SARAN.....</b>	<b>102</b>
4.1 Kesimpulan .....	102
4.2 Saran.....	103
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	

## DAFTAR TABEL

Tabel 3.1 Studi Literatur .....	46
Tabel 3.2 <i>Usecase</i> Login.....	58
Tabel 3.3 <i>Usecase</i> Lokasi .....	59
Tabel 3.4 <i>Usecase</i> Role.....	60
Tabel 3.5 <i>Usecase</i> Faskes .....	60
Tabel 3.6 <i>Usecase</i> Laporan .....	61
Tabel 3.7 <i>Usecase</i> Berita .....	61
Tabel 3.8 <i>Usecase</i> Registrasi .....	61
Tabel 3.9 <i>Usecase</i> Kategori .....	62
Tabel 3.10 <i>Usecase</i> List Registrasi .....	62
Tabel 3.11 Login .....	79
Tabel 3.12 Lokasi .....	79
Tabel 3.13 Role .....	80
Tabel 3.14 Kategori .....	80
Tabel 3.15 Berita .....	80

Tabel 3.16 Faskes .....	81
Tabel 3.17 Registrasi .....	81
Tabel 3.18 Registrasi Detail .....	82
Tabel 3.19 Data Dummy NIK.....	82
Tabel 4.1 Rencana Pengujian .....	98
Tabel 4.2 Hasil Pengujian .....	99

## DAFTAR GAMBAR

Gambar 1.1 Metode <i>Waterfall</i> .....	5
Gambar:2.1 Siklus <i>Waterfall</i> .....	19
Gambar:2.2 Simbol ERD .....	24
Gambar:2.3 Simbol <i>Usecase Diagram</i> .....	32
Gambar 2.4 Simbol <i>Activity Diagram</i> .....	33
Gambar 2.5 Simbol <i>Sequence Diagram</i> .....	34
Gambar 2.6 Simbol <i>Class Diagram</i> .....	35
Gambar 2.7 Cara kerja PHP.....	36
Gambar 2.8 Diagram struktur <i>Framework CI</i> .....	40
Gambar 3.1 <i>Flowchart</i> yang sedang berjalan .....	51
Gambar 3.2 <i>Flowchart</i> yang diusulkan.....	53
Gambar 3.3 <i>Usecase Diagram</i> .....	58
Gambar 3.4 <i>Activity Diagram</i> Login .....	63
Gambar 3.5 <i>Activity Diagram</i> Lokasi .....	64
Gambar 3.6 <i>Activity Diagram</i> Role.....	65
Gambar 3.7 <i>Activity Diagram</i> Data Faskes.....	66
Gambar 3.8 <i>Activity Diagram</i> Laporan.....	67
Gambar 3.9 <i>Activity Diagram</i> Kategori .....	68
Gambar 3.10 <i>Activity Diagram</i> Registrasi .....	69
Gambar 3.11 <i>Activity Diagram</i> Berita .....	70



Gambar 3.12 <i>Activity Update Kehadiran</i> .....	71
Gambar 3.13 <i>Sequence Diagram Login</i> .....	72
Gambar 3.14 <i>Sequence Diagram Lokasi</i> .....	73
Gambar 3.15 <i>Sequence Diagram Role</i> .....	74
Gambar 3.16 <i>Sequence Diagram Faskes</i> .....	75
Gambar 3.17 <i>Sequence Diagram Laporan</i> .....	76
Gambar 3.18 <i>Sequence Diagram Registrasi</i> .....	76
Gambar 3.19 <i>Class Diagram</i> .....	77
Gambar 3.20 ERD ( <i>Entity Realtionship Diagram</i> ).....	78
Gambar 3.21 Halaman Login.....	83
Gambar 3.22 Halaman Home.....	83
Gambar 3.23 Halaman Lokasi.....	84
Gambar 3.24 Halaman Faskes .....	84
Gambar 3.25 Halaman Berita .....	85
Gambar 3.26 Halaman Registrasi .....	85
Gambar 4.1 Halaman Login.....	89
Gambar 4.2 Halaman Home.....	89
Gambar 4.3 Halaman Registrasi .....	90
Gambar 4.4 Halaman Pendaftaran Vaksin .....	91
Gambar 4.5 Halaman Lokasi .....	91
Gambar 4.6 Halaman Lokasi Vaksin .....	92

Gambar 4.7 Halaman Tambah Lokasi .....	92
Gambar 4.8 Halaman List Registrasi Vaksin .....	93
Gambar 4.9 Halaman Update Kehadiran Vaksin .....	93
Gambar 4.10 Halaman Faskes .....	94
Gambar 4.11 Halaman View Faskes Vaksin .....	95
Gambar 4.12 Halaman Informasi .....	96
Gambar 4.13 Halaman Tambah Informasi .....	96
Gambar 4.14 Halaman Laporan Vaksin .....	97
Gambar 4.15 Halaman Laporan Registrasi Vaksin .....	97

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Penelitian**

Saat ini pemanfaatan Sistem Informasi Geografi (SIG) berkembang pesat diberbagai bidang termasuk di bidang kesehatan. SIG mempunyai kemampuan untuk menampilkan informasi spasial dan non spasial secara berkesinambungan yang berupa keterangan tulisan atau angka. Dengan pemanfaatan SIG dalam perencanaan program lokasi dan pendaftaran vaksin covid-19, dimungkinkan dilakukannya penampilan informasi spasial dan non spasial secara bersamaan serta dilakukannya analisis spasial. Oleh karena itu merupakan hal yang sangat menarik untuk memanfaatkan SIG dengan berbagai kelebihannya.

Dengan pemanfaatan SIG dalam sistem informasi untuk perencanaan program lokasi dan pendaftaran vaksin, diharapkan data mengenai peta wilayah daerah, jumlah sasaran bisa ditampilkan secara bersama. Sehingga, apabila suatu wilayah mempunyai jumlah sasaran target yang kurang dengan SIG bisa diputuskan dengan wilayah yang paling terdekat untuk pelaksanaan dan pendaftaran vaksin.

Dengan SIG juga diharapkan dapat dilakukan penampakan tampilan sebaran geografis pelayanan kesehatan non puskesmas yang memberikan pelayanan vaksin beserta informasi jumlah layanan yang diberikan. Dengan mempertimbangkan peran pelayanan kesehatan non puskesmas tersebut perencanaan vaksinasi diharapkan dapat lebih tepat.

Pemanfaatan SIG dalam sistem informasi untuk perencanaan program lokasi dan pendaftaran vaksin covid-19 diharapkan juga dapat mempermudah perhitungan data yang dilakukan secara terkomputerisasi. Selain itu, data juga dapat tersimpan dalam satu basis data komputer yang bisa dilakukan pembaruan dan pengaksesan secara cepat dan memungkinkan penggunaan data bersama.

Pemerintah Kota Cimahi berkewajiban untuk memberikan perlindungan kepada setiap warga dari penyebaran virus covid-19, salah satu penanganan yang dilakukan adalah memberikan vaksin gratis kepada setiap warganya. Hal ini dilakukan untuk mendukung program pemerintah pusat agar setiap warga negara yang telah divaksin mempunyai kekebalan tubuh dari paparan covid-19 serta mencegah mutasi baru dari covid-19.

Pada saat ini di Kota Cimahi sendiri masih terdapat warga yang belum melakukan vaksin sekitar 20% untuk penyuntikan dosis pertama, dan 30% untuk penyuntikan dosis yang kedua, hal itu disebabkan oleh beberapa faktor seperti jarak lokasi vaksin yang cukup jauh, keterbatasan kuota disetiap puskesmas serta sudah lanjut usia dan yang memiliki riwayat penyakit bawaan.

Ditinjau dari latar belakang diatas penulis tertarik untuk melakukan penelitian dengan judul "RANCANG BANGUN APLIKASI SISTEM INFORMASI GEOGRAFIS LOKASI DAN PENDAFTARAN VAKSIN COVID-19 BERBASIS WEB DI KOTA CIMAHI".

## **1.2 Identifikasi Masalah**

Penulis dapat mengidentifikasi beberapa masalah yang terjadi sesuai latar belakang diatas, yakni sebagai berikut:

1. Bagaimana pembuatan aplikasi sistem informasi geografis lokasi dan pendaftaran vaksin covid-19 di Kota Cimahi?
2. Apakah sistem informasi geografis ini dapat memberikan informasi yang jelas serta tidak terpaut oleh waktu?

## **1.3 Tujuan Penelitian**

Tujuan dari tugas akhir rancang bangun sistem informasi geografis lokasi dan pendaftaran vaksin covid-19 di Kota Cimahi :

1. Memudahkan masyarakat untuk menemukan lokasi dan pendaftaran vaksinasi covid-19 tanpa harus kesulitan dengan mencari alamat fisik.
2. Membuat aplikasi yang bisa membantu masyarakat memperoleh informasi penyebaran lokasi vaksin di Kota Cimahi dan juga bisa memberikan informasi mengenai kuota yang tersedia.

## **1.4 Batasan Masalah Penelitian**

Batasan masalah disini digunakan agar penelitian lebih terarah. Berikut adalah batasan masalah penelitian.

1. Pembangunan sistem informasi berbasis web ini menggunakan Bahasa pemrograman PHP dan Database MySQL.
2. Visualisasi peta menggunakan Mapbox API.

- 3 Data lokasi yang digunakan data puskesmas dan non puskesmas di Kota Cimahi.

## **1.5 Metode Penelitian**

Metode penelitian disini akan menjelaskan metode-metode yang digunakan penulis dalam melakukan penelitian ini. Metode penelitian digunakan untuk memaparkan permasalahan yang ada serta mengumpulkan data sesuai dengan yang sebenarnya dan mencoba memecahkan masalah dengan menggunakan program aplikasi PHP dan MySql diharapkan dapat membantu proses pengolahan data, sehingga akurasi yang diharapkan dapat dicapai.

### **1.5.1 Metode Pengumpulan Data**

1. Observasi

Observasi adalah metode pengumpulan data yang kompleks karena melibatkan berbagai faktor dalam pelaksanaannya. Metode pengumpulan data observasi tidak hanya mengukur sikap dari responden, namun juga dapat digunakan untuk merekam berbagai fenomena yang terjadi. Teknik pengumpulan data observasi cocok digunakan untuk penelitian yang bertujuan untuk mempelajari perilaku manusia, proses kerja, dan gejala-gejala alam.

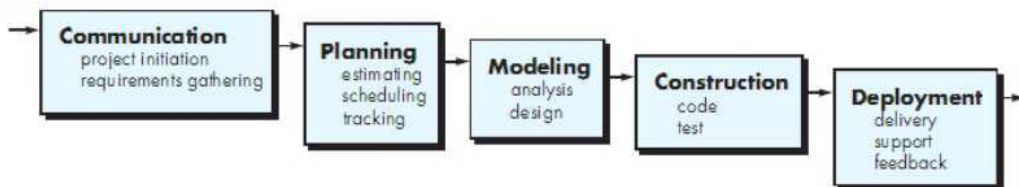
2. Studi Dokumen

Studi dokumen adalah metode pengumpulan data yang tidak ditujukan langsung kepada subjek penelitian. Studi dokumen adalah jenis

pengumpulan data yang meneliti berbagai macam dokumen yang berguna untuk bahan analisis.

### 1.5.2 Metode Pengembangan Sistem

Metode yang digunakan dalam rancang bangun aplikasi sistem informasi geografis lokasi dan pendaftaran vaksin covid-19 adalah metode *waterfall*. Alasan menggunakan metode ini adalah karena metode *waterfall* melakukan pendekatan secara sistematis dan berurutan dalam membangun suatu sistem. Proses metode *waterfall* yaitu pada pengerjaan dari suatu sistem dilakukan secara berurutan. Sistem yang dihasilkan akan berkualitas baik, dikarenakan pelaksanaannya secara bertahap sehingga tidak terfokus pada tahapan tertentu. Tahapan dari metode *waterfall* sendiri adalah sebagai berikut



Gambar 1.1. Metode *Waterfall* (Pressman, 2015)

#### 1. *Communication (Project Initiation & Requirement Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

2. *Planning (Estimating, Scheduling, & Tracking)*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko - resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem.

3. *Modelling (Analysis & Design)* adalah *prototype* yang terbatas pada antar muka pengguna (user interface).

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

4. *Construction (Code & Test)* adalah *prototype* yang meliputi perawatan file dasar dan proses-proses transaksi.

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.



## 5. *Deployment (Delivery, Support & Feedbak)*

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software*, dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015).

### **1.6 Sistematika penulisan**

Dalam penyusunan laporan skripsi ini penulis membuat sistematika penulisan yang terdiri dari 5 (lima) bab, adapun uraian masing-masing bab tersebut adalah sebagai berikut :

#### **BAB I : PENDAHULUAN**

Pada bab ini dibahas tentang beberapa hal diantaranya latar belakang penelitian, identifikasi masalah, tujuan penelitian, batasan masalah penelitian, metode penelitian, dan sistematika penulisan.

#### **BAB II : LANDASAN TEORI**

Bab Landasan Teori ini berisi penjelasan dan uraian singkat mengenai dasar teori yang mendukung dan digunakan dalam pengembangan sistem ini.

#### **BAB III : ANALISIS MASALAH DAN PERANCANGAN PROGRAM**

Bab ini menjelaskan tentang Analisis dan Perancangan Sistem ini berisi analisis dan perancangan sistem yang akan dibuat.

#### **BAB IV : IMPLEMENTASI**

Bab ini membahas mengenai Implementasi dan Pengujian Sistem ini berisi penerapan dan pembahasan hasil pengujian sistem yang telah dibuat.

#### **BAB V : PENUTUP**

Pada bab ini adalah bab terakhir yang akan berisi mengenai kesimpulan akhir dari penelitian ini serta beberapa saran dari penulis pribadi.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Konsep Dasar Sistem**

##### **2.1.1 Pengertian Sistem**

Sistem secara etimologis atau asal usul kata sistem berasal dari bahasa latin *systema* atau bahasa Yunani *sustema* yang memiliki arti suatu kesatuan dimana terdiri dari elemen atau komponen yang dihubungkan secara bersama supaya dapat memudahkan transfer materi, energi atau informasi. Sistem dikenal sebagai kesatuan bagian yang memiliki keterhubungan antara satu dengan yang lainnya dan mempunyai item-item penggerak. Semisal sistem adalah sistem tata surya, sistem pemerintahan Indonesia dan sistem eksresi di manusia serta sistem komputer.

Sistem adalah suatu rangkaian yang terdiri dari dua atau lebih komponen yang saling berhubungan dan saling berinteraksi satu sama lain untuk mencapai tujuan dimana sistem biasanya terbagi dalam sub system yang lebih kecil yang mendukung system yang lebih besar (Syaifullah, 2018).

Sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan. (Kristanto, 2018).

Terdapat dua kelompok pendekatan, yaitu menekankan pada prosedur dan komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada

prosedur pendefinisian sistem adalah suatu jaringan kerja dari prosedur-prosedur yang paling berhubungan, berkelompok dan bekerjasama untuk melakukan kegiatan pencapaian sasaran tertentu. Makna dari prosedur, yaitu urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya. Sedangkan pendekatan yang menekankan pada komponen mendefinisikan “sistem” adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Jogiyanto, 2015).

### **2.1.2 Karakteristik Sistem**

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan sasaran (*objectives*) atau tujuan (*goal*) (Jogiyanto, 2015).

Sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan atau sasaran. Unsur-unsur yang saling melengkapi tersebut terdapat di dalam sistem yang disebut dengan nama subsistem. Subsistem-subsistem tersebut harus selalu berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien.

## **2.2 Konsep Dasar Informasi**

### **2.2.1 Pengertian Informasi**

Secara Etimologi, Kata informasi ini berasal dari kata bahasa Perancis kuno *informacion* (tahun 1387) mengambil istilah dari bahasa Latin yaitu *informationem* yang berarti “Konsep, Ide atau Garis besar”. Informasi ini merupakan kata benda dari *informare* yang berarti aktivitas.

Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi. Dengan kata lain sumber informasi adalah data (Kristanto, 2018).

Informasi juga dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan (Jogiyanto, 2015). Informasi yang sudah usang tidak akan mempunyai nilai lagi. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya.

### **2.2.2 Kualitas Informasi**

Menurut Raymond mc. Leod, Adapun beberapa ciri-ciri terkait informasi yang berkualitas ciri-ciri tersebut adalah sebagai berikut (Raymond Mcleod, 2017).

1. Akurat

Informasi mencerminkan keadaan yang sebenarnya pengujian terhadap hal ini biasanya dilakukan melalui pengujian yang dilakukan oleh dua orang atau lebih yang berbeda-beda dan apabila hasil pengujian tersebut menghasilkan hasil yang sama maka dianggap data tersebut akurat.

2. Tepat Waktu

Informasi itu harus tersedia atau ada pada saat informasi tersebut diperlukan, tidak besok atau tidak beberapa jam lagi

3. Relevan

Informasi yang diberikan harus sesuai dengan yang dibutuhkan kalau kebutuhan informasi ini untuk suatu organisasi maka informasi tersebut harus sesuai dengan kebutuhan informasi diberbagai tingkatan dan bagian yang ada dalam organisasi tersebut.

4. Lengkap

Informasi harus diberikan secara lengkap karena bila informasi yang dihasilkan Sebagian-sebagian akan mempengaruhi dalam mengambil keputusan

5. *Correctness*

Berarti informasi yang dihasilkan atau dibutuhkan harus memiliki kebenaran.

6. *Security*

Berarti informasi yang dihasilkan mempunyai manfaat yang lebih besar dibandingkan dengan biaya mendapatkannya dan Sebagian besar informasi

tidak dapat ditaksir keuntungannya dan dengan satuan nilai uang tetapi dapat ditaksir nilai efektifitasnya.

### **2.2.3 Jenis-jenis Informasi**

1. Informasi berdasarkan fungsi dan kegunaan, adalah informasi berdasarkan materi dan kegunaan informasi. Informasi jenis ini antara lain adalah :
  - a. Informasi yang menambah pengetahuan, misalnya: peristiwa- peristiwa, pendidikan, kegiatan sosial.
  - b. Informasi yang mengajari pembaca (informasi edukatif), misalnya makalah yang berisi tentang cara berternak itik, artikel tentang cara membina persahabatan, dan lain-lain.
  - c. Informasi berdasarkan format penyajian, yaitu informasi yang dibedakan berdasarkan bentuk penyajian informasinya. Misalnya: informasi dalam bentuk tulisan (berita, artikel, esai, resensi, kolom, tajuk rencana, dll),
2. Informasi berdasarkan format penyajian, adalah informasi yang berdasarkan bentuk penyajian. Informasi jenis ini, antara lain berupa tulisan teks, karikatur, foto, ataupun lukisan abstrak.
3. Informasi berdasarkan lokasi peristiwa, adalah informasi berdasarkan lokasi peristiwa berlangsung, yaitu informasi dari dalam negeri dan informasi dari luar negeri.
4. Informasi berdasarkan bidang kehidupan adalah informasi berdasarkan bidang-bidang kehidupan yang ada, misalnya

pendidikan, olahraga, musik, sastra, budaya, dan iptek.

5. Berdasar penyampaian:
  - a. Informasi yang disediakan secara berkala
  - b. Informasi yang disediakan secara tiba-tiba
  - c. Informasi yang disediakan setiap saat
  - d. Informasi yang dikecualikan
  - e. Informasi yang diperoleh berdasarkan permintaan

## **2.3 Konsep Dasar Sistem Informasi**

### **2.3.1 Pengertian Sistem Informasi**

Untuk memahami pengertian sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang (Fatta, 2017). Sistem informasi dapat didefinisikan sebagai suatu sistem didalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yagn lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambil keputusan yang cerdas. (Fatta, 2017).



Menurut (Kristanto, 2018) Sistem informasi adalah Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.

### **2.3.2 Komponen Sistem Informasi**

Untuk mendukung kelancaran berjalannya sebuah sistem informasi, dibutuhkan beberapa komponen vital. Secara rinci komponen-komponen sistem informasi dapat dijelaskan sebagai berikut: (Kristanto, 2018)

#### *1. Input*

*Input* disini adalah semua data yang dimasukkan kedalam sistem informasi. Dalam hal ini yang termasuk dalam input adalah dokumen-dokumen, formulir-formulir dan *file-file*. Dokumen-dokumen tersebut dikumpulkan dan dikonfirmasi ke suatu bentuk sehingga dapat diterima oleh pengolah yang meliputi:

- a. Pencatatan
- b. Penyimpanan
- c. Pengujian
- d. Pengkodean

#### *2. Proses*

Proses merupakan kumpulan prosedur yang akan memanipulasi input yang kemudian akan disimpan dalam bagian basis data dan seterusnya akan diolah menjadi suatu output yang akan digunakan oleh penerima. Komponen ini dalam tugasnya akan merubah segala masukan menjadi keluaran

### 3. *Output*

*Output* merupakan semua keluaran atau hasil dari model yang sudah diolah menjadi suatu informasi yang berguna dan dapat dipakai penerima. Komponen ini akan berhubungan langsung dengan pemakai sistem informasi dan merupakan tujuan akhir dari pembuatan sistem informasi. Komponen ini dapat berupa laporan-laporan yang dibutuhkan oleh pemakai sistem untuk memantau keberhasilan suatu organisasi.

### 4. Teknologi

Teknologi disini merupakan bagian yang berfungsi untuk memasukan input, mengolah input dan menghasilkan keluaran. Ada 3 bagian dalam teknologi ini yang meliputi perangkat keras, perangkat lunak dan manusia. perangkat keras contohnya *keyboard*, *mouse* dan lain-lain. Perangkat lunak contohnya program untuk mengolah data dan perangkat. Manusia contohnya analis sistem, *programmer*, teknisi dan sebagainya.

### 5. Basis Data

Basis data merupakan kumpulan data-data yang saling berhubungan satu dengan yang lain yang disimpan dalam perangkat keras komputer dan akan diolah menggunakan perangkat lunak. Basis data sendiri merupakan kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* yang lain sehingga membentuk satu bangunan data.

## 6. Kendali

Kendali dalam hal ini merupakan semua Tindakan yang diambil untuk menjaga sistem informasi tersebut agar bisa berjalan dengan lancar dan tidak mengalami gangguan. Komponen ini sangat penting agar sistem secara keseluruhan memiliki validasi dan integritas yang tinggi. Komponen kendali diperlukan terhadap : *backup file*, *reindexing*, pengujian kebenaran data tiap *entry* yang dilakukan.

### 2.4 Sistem Informasi Geografis

Sistem Informasi Geografi (SIG) merupakan ilmu pengetahuan yang berbasis pada perangkat lunak komputer yang digunakan untuk memberikan bentuk digital dan analisa terhadap permukaan geografi bumi sehingga membentuk suatu informasi keruangan yang tepat dan akurat Definisi SIG selalu berubah, hal ini terlihat dengan banyaknya definisi SIG yang telah beredar dan juga SIG merupakan bidang kajian ilmu dan teknologi yang relatif masih baru. Berikut adalah definisi SIG dari beberapa pustaka yang beredar (Suryantoro, 2019):

1. SIG adalah kumpulan yang terorganisir dari perangkat keras komputer, perangkat lunak, data geografi dan personil yang dirancang secara efisien untuk memperoleh, menyimpan, memperbarui, memanipulasi, menganalisis dan menampilkan semua bentuk informasi yang bereferensi geografi.
2. SIG adalah sistem komputer yang digunakan untuk memanipulasi data geografi. Sistem ini diimplementasikan dengan perangkat keras

komputer dan perangkat lunak komputer yang berfungsi untuk: (a) akuisisi dan verifikasi data, (b) kompilasi data, (c) penyimpanan data, (d) perubahan dan updating data, (e) manajemen dan pertukaran data, (f) manipulasi data, (g) pemanggilan dan presentasi data, (h) analisis data.

3. SIG merupakan suatu sistem yang mengorganisir perangkat keras (hardware), perangkat lunak (software), dan data, serta dapat mendayagunakan sistem penyimpanan, pengolahan, maupun analisis data secara simultan, sehingga dapat diperoleh informasi yang berkaitan dengan aspek keruangan.

#### **2.4.1 Jenis Data Masukan Sistem Informasi Geografis**

Jenis Data Masukan Sistem Informasi Geografis (SIG) Di dalam sistem informasi geografis terdapat 2 jenis data, yaitu:

##### **1. Data Spasial**

Data spasial merupakan data yang memuat tentang lokasi suatu objek dalam peta berdasarkan posisi geografis objek tersebut di atas bumi dengan menggunakan sistem koordinat. Data spasial direpresentasikan dengan model vektor dan model raster.

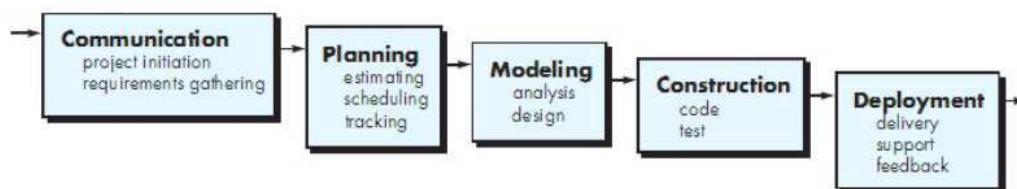
##### **2. Data Non-Spasial**

Data ini merupakan data yang memuat karakteristik atau keterangan dari suatu objek yang terdapat dalam peta yang sama sekali tidak berkaitan dengan posisi geografis objek tertentu. Sebagai contoh, data atribut dari sebuah kota adalah

luas wilayah, jumlah penduduk, tingkat kriminalitas, dan sebagainya (Tjiptanata Agus, 2017).

## 2.5 Metode Pengembangan Sistem

Metode pengembangan sistem penulis merancang sistem informasi dengan menggunakan metode pengembangan sistem *waterfall*. Menurut (Pressman, 2015) Proses metode waterfall yaitu pada pengerjaan dari suatu sistem dilakukan secara berurutan. Sistem yang dihasilkan akan berkualitas baik, dikarenakan pelaksanaannya secara bertahap sehingga tidak terfokus pada tahapan tertentu.



Gambar : 2.1 Siklus *Waterfall* (Pressman, 2015)

### 1. *Communication (Project Initiation & Requirement Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

### 2. *Planning (Estimating, Scheduling, & Tracking)*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko -

resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem.

3. *Modelling (Analysis & Design)* adalah *prototype* yang terbatas pada antar muka pengguna (user interface).

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

4. *Construction (Code & Test)* adalah *prototype* yang meliputi perawatan file dasar dan proses-proses transaksi.

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

5. *Deployment (Delivery, Support & Feedbak)*

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software*, dan pengembangan *software* berdasarkan umpan

balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015).

## **2.6 Database (Basis Data)**

Istilah database atau disebut juga basis data banyak memiliki definisi. Untuk sebagian kalangan, secara sederhana database diartikan sebagai kumpulan data seperti buku, nomor telepon, daftar pegawai, dan lain sebagainya. Ada juga menyebut database dengan definisi lain yang lebih formal dan tegas”. Basis Data atau Database didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat. Selain berisi data, database juga berisi metadata. Metadata adalah data yang menjelaskan tentang struktur dari data itu sendiri (Raharjo, 2015).

Pengertian *database* adalah sekumpulan data yang sudah disusun sedemikian rupa dengan ketentuan atau aturan tertentu yang saling berelasi sehingga memudahkan pengguna dalam mengelolanya juga memudahkan memperoleh informasi. Selain itu adapula yang mendefinisikan database sebagai kumpulan file, tabel, atau arsip yang saling terhubung yang disimpan dalam media elektronik. (Muiz, 2017).

## **2.7 Pengertian Diagram Flowchart**

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Biasanya, seorang analis sistem menggunakan flowchart sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang

akan dibangun kepada programmer. Dengan begitu, flowchart dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, flowchart digambarkan dengan menggunakan simbol- simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung.

Flowchart adalah gambaran secara grafis dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri dari langkah-langkah yang meliputi sekumpulan simbol, dimana masingmasing simbol menggambarkan kegiatan tertentu. Dari alur yang tergambar bisa diketahui apakah pekerjaan tersebut terlalu berbelit-belit, atau beban kerja tidak seimbang di unit tertentu. Menurut (Indrajani, 2015) Flowchart adalah penggambaran secara grafik dari langkahlangkah dan urutan prosedur suatu program.

Dengan adanya flowchart, setiap urutan proses dapat digambarkan menjadi lebih jelas. Selain itu, ketika ada penambahan proses baru dapat dilakukan dengan mudah menggunakan flowchart ini. Setelah proses membuat flowchart selesai, maka giliran programmer yang akan menerjemahkan desain logis tersebut kedalam bentuk program dengan berbagai bahasa pemrograman yang telah disepakati. (Irmayani, 2018).

## **2.8 Entity Relationship Diagram (ERD)**

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar

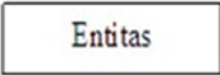





relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan symbol.

*Entity Relationship Diagram* (ERD) Alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek kedalam entitas – entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien”. *Entity Relationship Diagram* (ERD) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek – objek dasar yang disebut entitas dan hubungan antar objek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain Entitas digambarkan dalam basis data dengan kumpulan atribut. Relasi adalah hubungan antara beberapa entitas. Kumpulan semua entitas bertipe sama disebut kumpulan entitas (*Entity Set*), sedangkan kumpulan semua relasi bertipe sama disebut kumpulan relasi (*Relationship Set*). Struktur logis (skema *database*) dapat ditunjukkan secara grafis dengan ERD yang dibentuk dalam komponen – komponen (Simarmata, 2016).

- a. *Entity* (Entitas) Adalah suatu yang nyata atau abstrak dimana kita akan menyimpan data.
- b. *Relationship* (Relasi) Adalah hubungan alamiah yang terjadi antara satu atau lebih entitas. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan.
- c. *Attribute* (Atribut) Adalah ciri umum semua atau sebagian bsar instansi pada entitas tertentu. Sebutan lain atrbut adalah property, elemen data,

data field. Sebuah atribut atau kombinasi atribut yang mengidentifikasi satu dan hanya satu instansi suatu entitas disebut kunci utama atau pengenal

Komponen	Keterangan
 Entitas	<b>Persegi panjang mewakili entitas</b>
 Atribut	<b>Elips mewakili atribut</b>
 Relasi	<b>Belah ketupat mewakili relasi</b>
	<b>Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi</b>

**Gambar : 2.2** Simbol ERD (Simarmata, 2016)

## 2.9 OOP (*Object Oriented Programming*)

*Object-Oriented Programming* (OOP) adalah sebuah pendekatan untuk pengembangan / *development* suatu *software* dimana dalam struktur *software* tersebut didasarkan kepada interaksi *object* dalam penyelesaian suatu proses / tugas. Interaksi tersebut mengambil *form* dari pesan-pesan dan mengirimkannya kembali antar *object* tersebut. *Object* akan merespon pesan tersebut menjadi sebuah tindakan / *action* atau metode. Jika kita mencoba melihat bagaimana tugas disekitar kita diselesaikan, kita akan mengetahui bahwa kita berinteraksi dalam sebuah *object-oriented world*. Jika akan bepergian kita pasti berinteraksi dengan *object* mobil. Sebagai sebuah *object*, mobil berisi *object-object* lain yang

berinteraksi untuk melakukan tugasnya membawa kita.

*Object-oriented programs* terdiri dari *objects* yang berinteraksi satu sama lainnya untuk menyelesaikan sebuah tugas. Seperti dunia nyata, *users* dari *software programs* dilibatkan dari logika proses untuk menyelesaikan tugas.

### **2.9.1 Sejarah OOP (*Object Oriented Programming*)**

Konsep OOP bermula pada era 1960-an. Sebuah bahasa pemrograman simula memperkenalkan berbagai konsep yang mendasari OOP dengan SIMULA I (1962-65) dan SIMULA 67 (1967). Kemudian pada tahun 70-an, bahasa pemrograman *Smalltalk* menjadi yang pertama kali disebut *object-oriented*.

Pada tahun 1980-an, dua bahasa pemrograman ADA (*US Department of Defense*) dan PROLOG (*the Japanese "Fifth Generation Computer Project"*) dipercayai akan bersaing ketat sebagai bahasa pemrograman yang paling dominan. Namun justru OOP yang menjadi paradigma pemrograman yang paling dominan sampai sekarang. Bahasa pemrograman yang *object-oriented* seperti C++ pada tahun 80-an menjadi populer. Pada tahun 90-an, bahasa-bahasa pemrograman seperti *Java* mulai menerapkan OOP. Sampai pada 2002, Microsoft Visual Studio memperkenalkan bahasa *object-oriented* baru yang diberi nama C#. Disusul VB.NET yang merupakan penyempurnaan Visual Basic 6.0 yang tidak mendukung OOP.

## 2.9.2 Istilah-istilah OOP (*Object Oriented Programming*)

### 1. Kelas (*Class*)

Kumpulan atas definisi *data* dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh '*class of dog*' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah *class* adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi *object*. Sebuah *class* secara tipikal sebaiknya dapat dikenali oleh seorang *non-programmer* sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah *class* sebaiknya (relatif) bersifat mandiri dan *independen* (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

### 2. Objek (*Object*)

Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer, objek merupakan dasar dari modularitas dalam sebuah program komputer berorientasi objek.

### 3. Metode (*Method*)

Merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu *object*. Method didefinisikan pada *class* akan tetapi dipanggil melalui *object*. Contoh pada *object* mangga : terdapat method *ambilRasa* , *kupasKulit* dan lain-lain.

Metode menentukan perilaku *object*, yakni apa yang terjadi ketika *object* itu dibuat serta berbagai operasi yang dapat dilakukan *object* sepanjang hidupnya. Metode memiliki empat bagian dasar :

1. Nama metode
2. Tipe Objek atau tipe primitive yang dikembalikan metode.
3. Daftar parameter.
4. Badan atau isi metode.

Tiga bagian pertama mengindikasikan informasi penting tentang metode itu sendiri. Dengan kata lain, nama metode tersebut metode lain dalam program. Dalam java kita dapat memiliki metode-metode berbeda yang memiliki nama sama tetapi berbeda tipe kembalian atau daftar argumennya, sehingga bagian-bagian definisi metode ini menjadi penting. Ini disebut *overloading metode*.

#### 4. Abstraksi (*Abstraction*)

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap *object* dalam sistem melayani sebagai model dari “pelaku” abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan *object* lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

#### 5. Enkapsulasi (*Encapsulation*)

Memastikan pengguna sebuah *object* tidak dapat mengganti keadaan dalam dari sebuah *object* dengan cara yang tidak layak, hanya metode dalam *object* tersebut yang diberi ijin untuk mengakses keadaannya. Setiap *object* mengakses *interface* yang menyebutkan bagaimana *object* lainnya dapat berinteraksi dengannya. *Object* lainnya tidak akan mengetahui dan tergantung kepada representasi dalam *object* tersebut.

#### 6. Polimorfisme (*Polymorphism*)

Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi *object* dapat mengirim pesan, metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesan tersebut dikirim. Contohnya, bila sebuah burung menerima pesan “gerak cepat”, dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut *polimorfisme* karena sebuah *variabel* tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan *teks* program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai *polimorfisme* melalui penggunaan fungsi kelas-pertama.

## 7. Inheritas (*inheritance*)

Mengatur *polimorfisme* dan *enkapsulasi* dengan mengizinkan *object* didefinisikan dan diciptakan dengan jenis khusus dari *object* yang sudah ada *object-object* ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis *object* tidak selalu memiliki inheritas.)

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi *object-object* apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bagian administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui *object* petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap *object* memiliki deskripsi tugasnya sendiri.

### 2.10 *Unified Modeling Language (UML)*

UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi object dan design berorientasi object (*OOAD&D/object oriented analysis and design*) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. UML merupakan gabungan dari metode Booch, Rumbaugh

(OMT) dan Jacobson. Tetapi UML mencakup lebih luas daripada OOAD. Pada pertengahan saat pengembangan UML, dilakukan standarisasi proses dengan OMG (Object Management Group) dengan harapan UML bakal menjadi bahasa standar pemodelan pada masa yang akan datang (yang sekarang sudah banyak dipakai oleh berbagai kalangan) (Aditya, 2019). Saat ini UML sudah menjadi bahasa standar dalam penulisan blue print software. Unified Modeling Language adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artifak dari proses analisis dan disain berorientasi objek. UML menyediakan standar pada notasi dan diagram yang bisa digunakan untuk memodelkan suatu system. UML menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif objek antara user dengan developer, antara developer dengan developer, antara developer analis dengan developer disain, dan antara developer disain dengan developer pemrograman.

UML memungkinkan developer melakukan permodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Permodelan visual membantu untuk menangkap struktur dan kelakuan dari objek, mempermudah penggambaran interaksi antara elemen dalam system, dan mempertahankan konsistensi antara disain dan implementasi dalam pemrograman.

Namun karena UML hanya merupakan bahasa untuk pemodelan maka UML bukanlah rujukan bagaimana melakukan analisis dan disain berorientasi objek secara baik, sudah terdapat beberapa metodologi yang bisa diikuti, seperti Metode Booch, Metode Coad dan Yourdan, Metode Jacobson, Metode Rumbaugh, Metode




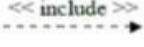
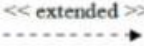


Wirfs-Brock, atau mengikuti metode pengembangan *system Unified Process*. (Hermanto, 2019)

Menurut Seidl, Scholz, Huemer, Kappel, UML (Unified Modeling Language) merupakan konsolidasi dari praktek-praktek terbaik yang telah ditetapkan selama bertahun-tahun dalam penggunaan bahasa pemodelan. UML memungkinkan kita untuk menyajikan aspek yang sangat beragam dari sistem perangkat lunak (persyaratan, struktur data, arus data, dan arus Informasi) dalam kerangka tunggal dengan menggunakan konsep object-oriented. (Seidl, M., Scholz, M., Huemer, C., & Kappel, 2017).

### **1. Diagram Use Case**

*Use case diagram*, yaitu diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan aktor. Sedangkan menurut John Satzinger dalam Triandini dan Suardika Menurut (Shalahuddin, 2019) mengemukakan bahwa *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang dibuat. (Mulyani, 2016).

SIMBOL	NAMA	KETERANGAN
	Aktor	Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang kita kembangkan.
	Use Case	Peringkat Tertinggi dari fungsional yang dimiliki sistem.
	Relasi Asosiasi	Relasi yang terjadi antara aktor dengan use case biasanya berupa asosiasi.
	Include Relationship	Relasi cakupan memungkinkan suatu use case untuk menggunakan fungsionalitas yang disediakan oleh use case yang lainnya
	Extends Relationship	Memungkinkan suatu use case memiliki kemungkinan untuk memperluas fungsional yang disediakan use case yang lainnya.








**Gambar : 2.3 Simbol Usecase Diagram** (Nugroho, 2018)

## 2. Diagram Activity

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram ini sangat mirip dengan sebuah flowchart karena dapat dimodelkan sebuah alur kerja dari satu aktivitas ke aktivitas Lainnya kedalam keadaan sesaat (*state*). Didalam diagram aktivitas banyak juga mendefinisikan hal-hal berikut ini : (Shalahuddin, 2019).

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- d. Rancangan menu yang ditampilkan pada perangkat lunak. Didalam *activity diagram* terdapat juga beberapa simbol.


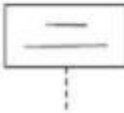



Simbol	Nama	Keterangan
	<i>Start State</i>	Memperlihatkan aliran kerja berawal.
	<i>End State</i>	Memperlihatkan aliran kerja berakhir.
	<i>State</i>	Menambahkan state suatu objek
	<i>Activity</i>	Menggambarkan langkah-langkah dalam aliran kerja.
	<i>Decision</i>	Memperlihatkan pengambilan keputusan dari dua atau lebih langkah pada aliran-aliran kerja.
	<i>Transition</i>	Memperlihatkan arah aliran-aliran kerja bergerak dari suatu <i>activity</i> ke <i>activity</i> lainnya.
	<i>Swimlane</i>	Memperlihatkan orang atau badan yang bertanggung jawab untuk melaksanakan tugas-tugas tertentu pada <i>activity diagram</i>

**Gambar : 2.4** Simbol *Activity Diagram* (Nugroho, 2018)

### 3. Diagram Sequence

sequence diagram merupakan suatu diagram interaksi yang menggambarkan bagaimana objek-objek berpartisipasi bagaimana objek-objek berpartisipasi Untuk menggambarkan sequence diagram harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas. Membuat sequencediagram harus disesuaikan dengan alur skenario use case. Banyaknya


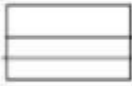

sequence diagram yang dibuat harus disesuaikan dengan banyaknya pendefinisian use case. (Indrajani, 2015). Diagram ini secara khusus berasosisasi dengan use case diagram. Sequence diagram memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam sequence diagram.

SIMBOL	NAMA	KETERANGAN
	Anchor Note to item	Melekatkan catatan pada use case atau aktor tertentu dalam diagram.
	Object	Menambahkan objek baru pada diagram.
	Object Message	Menggambarkan pesan antar dua objek.
	Return Message	Menggambarkan pengembalian diri pemanggilan prosedur.
	Message to self	Menggambarkan pesan yang menuju dirinya sendiri.

**Gambar 2.5** Simbol *sequence diagram* (Nugroho, 2018)

#### 4. Diagram Class

Class diagram membantu programmer dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Diagram ini digunakan untuk menggambarkan perbedaan yang mendasar antara class-class, hubungan antar class, dan dimana sub-sistem class tersebut.” Pada class diagram terdapat nama class, attribute, operation, serta association. (Indrajani, 2015).

SIMBOL	NAMA	KETERANGAN
	Interface	Menambahkan kelas antarmuka pada diagram
	Class	Menambahkan kelas baru pada diagram
	Association	Menggambar relasi Asosiasi

**Gambar 2.6** Simbol *class diagram* (Nugroho, 2018)

## 2.11 *Software* Pendukung Sistem

Software yang digunakan untuk membangun sistem inventory ini adalah PHP (Hypertext Preprocessor), HTML (*Hypertext Markup Language*), dan CSS.

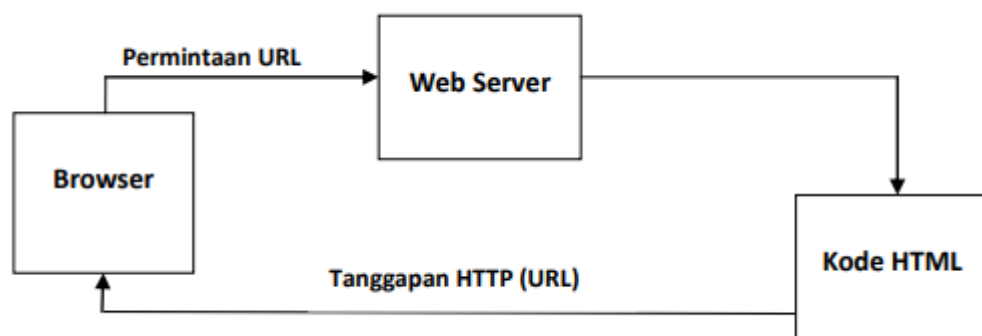
### 2.11.1 *Perl Hypertext Preprocessor (PHP)*

PHP memiliki kepanjangan dari Personal Home Page merupakan suatu bahasa pemrograman yang berfungsi untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML tetapi beda kondisi, HTML digunakan sebagai pembangun atau pondasi kerangka layout web, sedangkan PHP digunakan sebagai prosesnya. PHP berjalan pada sisi server, sehingga PHP juga disebut sebagai bahasa server side scripting. PHP membutuhkan web server dalam menjalankannya. PHP juga bersifat open source, sehingga dapat dipakai secara cuma-cuma dan mampu lintas platform, berjalan pada sistem operasi windows maupun linux. PHP juga dibangun sebagai modul pada web server apache dan sebagai binary yang dapat berjalan sebagai CGI (Saputra, Agus dan Agustin, 2017).

PHP adalah bahasa server side scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis, karena PHP merupakan server-side scripting maka sintaks dan perintah-perintah PHP akan dieksekusi ke server yang kemudian hasilnya dikirimkan ke browser. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. PHP bernama FI (Form Interpreted), pada saat tersebut PHP adalah sekumpulan script yang digunakan untuk mengolah data form dari web (Saputra, Agus dan Agustin, 2017).

PHP merupakan bahasa pemrograman script yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memprogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. PHP merupakan bahasa Server Side Scripting, dimana PHP selalu membutuhkan web server dalam menjalankan aksinya. Secara prinsip, server akan bekerja apabila ada permintaan dari klien, yaitu kode-kode PHP. Client tersebut akan dikirimkan ke server, kemudian server akan mengembalikan pada halaman sesuai instruksi yang diminta (Saputra, Agus dan Agustin, 2017).

Cara kerja PHP menurut Saputra (2012) adalah sebagai berikut :



**Gambar 2.7** cara kerja PHP (Saputra, Agus dan Agustin, 2017)

### ***2.11.2 Hypertext Markup Language (HTML)***

Hypertext Markup Language adalah salah satu format yang digunakan untuk menulis halaman web, HTML ini berjalan di web browser dan memiliki fungsi untuk melakukan pemrograman aplikasi di atas web. HTML merupakan pengembangan dari standar pemformatan dokumen teks, yaitu standar Generalized Markup Language. HTML sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu (Kadir, 2014).

### ***2.11.3 Cascading Style Sheet (CSS)***

Cascading Style Sheet atau sering disebut CSS merupakan suatu bahasa programan web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur dan seragam. CSS sebagai lintas platform, CSS dapat dijalankan pada berbagai macam sistem operasi dan web browser. Proses yang dilakukan oleh CSS adalah Pengaturan layout, kerangka, teks, gambar, warna, tabel, spasi, dan lain sebagainya. Ada dua sifat CSS, yaitu Internal dan Eksternal.

#### **1) Internal**

Kode CSS yang akan dibuat dimasukkan atau disisipkan ke dalam file kode HTML. Cara ini cukup merepotkan untuk membuat file lain dengan metode yang sama karena kita harus menyisipkan kode CSS secara berulang-ulang.

#### **2) Eksternal**

Pembuatan Kode CSS dan HTML terpisah. Hanya membuat satu file CSS dan dapat digunakan berulang-ulang untuk dihubungkan dengan file HTML

tanpa harus menyisipkan lagi kode CSS. Cara ini yang disarankan untuk pembuatan desain web karena lebih mudah (Saputra, Agus dan Agustin, 2017).

#### **2.11.4 Framework Code Igniter**

Menurut (Hakim, 2018) CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi *web* berbasis PHP dibanding jika menulis semua kode program dari awal. Saat ini, *CodeIgniter* dikembangkan oleh *Expression Engine Development Team*. Adapun beberapa keuntungan menggunakan *CodeIgniter*, diantaranya:

1. Gratis

*CodeIgniter* berlisensi dibawah Apache/BSD opensorce.

2. Ditulis Menggunakan PHP 4

Meskipun CodeIgniter dapat berjalan di PHP 5, namun sampai saat ini kode program CodeIgniter masih dibuat dengan menggunakan PHP 4.

3. Berukuran Kecil

Ukuran CodeIgniter yang kecil merupakan keunggulan tersendiri. Dibanding dengan *framework* lain yang berukuran besar.

4. Menggunakan Konsep MVC

CodeIgniter menggunakan konsep MVC yang memungkinkan pemisahan *layer application-logic* dan *presentation*.

5. URL yang Sederhana

Secara default, URL yang dihasilkan CodeIgniter sangat bersih dan *Serach*



*Engine Friendly* (SEF).

6. Memiliki Paket *Library* yang Lengkap

CodeIgniter mempunyai *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis *web*, misalnya mengakses *database*, mengirim email, memvalidasi form, menangani *session* dan sebagainya.

7. *Extensible*

Sistem dapat dikembangkan dengan mudah menggunakan *plugin* dan *helper*, atau dengan menggunakan *hooks*.

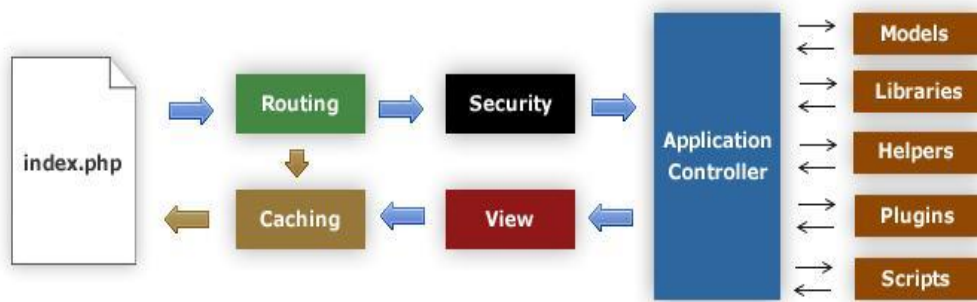
8. Tidak Memerlukan *Template Engine*

Meskipun *CodeIgniter* dilengkapi dengan *template parser* sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.

9. Dokumentasi Lengkap dan Jelas

Dari sekian banyak *framework*, *CodeIgniter* adalah satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas.

Proses aliran data aplikasi pada sistem dapat diilustrasikan seperti terlihat pada gambar 2.8.



**Gambar 2.8 Diagram struktur *Framework CI***

**Sumber : (Hakim, 2018) *Membangun Web Berbasis PHP dengan Framework CodeIgniter.***

Keterangan :

1. *Index.php* berfungsi sebagai *front controller*, menginisialisasi *base resource* untuk menjalankan *CodeIgniter*.
2. *Router* memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan dengannya.
3. Jika *Cache* aktif, maka hasilnya akan langsung dikirimkan ke *browser* dengan mengabaikan aliran data normal.
4. *Security*. Sebelum *Controller* dimuat, *HTTP request* dan data yang dikirimkan *user* akan disaring untuk keamanan.
5. *Controller* memuat *model*, *core libraries*, *plugins*, *helpers* dan semua *resource* yang diperlukan untuk memproses *request*.

Akhirnya *View* yang dihasilkan akan dikirimkan ke *browser*. Jika *Cache* aktif, maka *View* akan disimpan sebagai *Cache* dahulu, sehingga pada *request* berikutnya langsung dapat ditampilkan.

### **2.11.5 Mapbox**

*Mapbox* adalah sebuah *Maps Platform* yang dikhususkan bagi *developer* untuk membangun sebuah aplikasi desktop maupun *mobile* yang terintegrasi dengan peta. Beberapa layanan IT yang menggunakan *Mapbox* ini diantaranya adalah *Foursquare*, *Evernote*, *Pinterest* bahkan *Uber*. Untuk data petanya sendiri *Mapbox* menggunakan *base tiles* dari *OpenStreetMaps*, dan juga dari beberapa sumber lainnya, seperti *NASA*, *DigitalGlobe*, dan *USGS*.

*Mapbox* juga memiliki beberapa fitur yang tidak dimiliki oleh *google maps* seperti *access token*. *Access Token* ini untuk menampilkan peta secara *online*. *MapBox* sendiri defaultnya menyediakan 4 jenis tipe *Maps*, yaitu *MapBox Street*, *Emerald*, *Light* and *Dark*, dan *Satellite*. Namun pada aplikasi ini peneliti hanya menggunakan *Mapbox Street*.

#### **2.11.5.1 Mapbox Direction API**

*Mapbox Direction API* akan menunjukkan cara untuk mendapatkan di mana Anda akan pergi. Dengan *Direction API*, Anda dapat:

1. Menghitung rute mengemudi, jalan kaki dan bersepeda yang optimal.
2. Menghasilkan petunjuk arah *Direction*.
3. Menghasilkan rute dengan sampai 25 titik arah di mana saja

API ini mendukung empat *profil routing* yang berbeda :

1. *Mapbox.driving* untuk mobil dan motor. Menunjukkan rute tercepat dengan memilih jalan kecepatan tinggi seperti jalan raya.

2. *Mapbox.walking* untuk pejalan kaki dan hiking. Menunjukkan jalur terpendek dengan menggunakan trotoar dan jalan.
3. *Mapbox.cycling* untuk sepeda. Menunjukkan rute yang pendek dan lebih aman bagi pengendara sepeda dengan menghindari jalan raya dan lebih memilih jalan-jalan dengan jalur sepeda.

#### 2.11.5.2 *Mapbox Map API*

Untuk menggunakan *Mapbox API*, atau SDK, memerlukan *Access Token*. *Mapbox* menggunakan *Access Token* untuk mengaitkan permintaan ke sumber daya API dengan akun. *Developer* dapat menemukan semua *Access Token*, membuat yang baru, atau menghapus URL yang ada di halaman *Access Token API*.

Untuk membuat *Access Token* baru:

1. Di *Mapbox Studio*, navigasikan ke halaman *Account* dan klik tab *API Access Token*.
2. Klik *Buat token baru* dan berikan nama token baru untuk membantu mengingat tujuannya.
3. Jika ingin menentukan cakupan, klik *Tambahkan*.
4. Klik *Buat* untuk membuat token. *Developer* diminta memasukkan kembali kata sandi.

*Access Token*

```
https://api.mapbox.com/v4/mapbox.emerald/page.html?access_token=<your access token here>
```

### 2.11.6 Model View Controller (MVC)

*Model-View-Controller* (MVC) adalah model pembuatan program yang menerapkan arsitektur aplikasi menjadi tiga bagian yaitu memisahkan antara proses, tampilan dan bagian yang menghubungkan dengan *database*. MVC bertujuan untuk memisahkan proses bisnis dari pertimbangan antarmuka *user* agar para pengembang bisa lebih mudah mengembangkan salah satu bagian dari aplikasi sehingga tidak memengaruhi bagian yang lain (Badiyanto, 2013).

Dengan MVC, maka memungkinkan pemisahan antara *laye application-logic* dan *presentation*. Sehingga, dalam sebuah pengembangan *web*, seorang *programmer* bisa berkonsentrasi pada *core-system*, sedangkan *web designer* bisa berkonsentrasi pada tampilan *web*. Menariknya, skrip PHP, *query* MySQL, Javascript dan CSS bisa saling terpisah, tidak dibuat dalam satu skrip berukuran besar yang membutuhkan *resource* besar pula untuk mengesekusinya.

Dalam konteks CodeIgniter dan aplikasi berbasis *web*, maka penerapan konsep MVC mengakibatkan kode program dapat dibagi menjadi tiga kategori (Badiyanto, 2013), yaitu:

1. *Model*

*Model* merupakan kelas yang mendasari logika proses dalam aplikasi perangkat lunak dan kelas yang terkait dengannya. *Model* adalah suatu objek yang tidak mengandung informasi tentang *user interface*. *Model* juga merupakan suatu kelas yang berisi metode/fungsi dan digunakan untuk menyimpan data dan aturan bisnis yang relevan.

## 2. *View*

*View* merupakan kumpulan dari kelas yang mewakili unsur-unsur dalam antarmuka, dalam *view* terdapat nama yang dipakai untuk mengidentifikasi *file script* tampilan saat dipanggil lewat fungsi *render*. Nama *view* sama seperti nama *file script view*-nya.

## 3. *Controller*

*Controller* merupakan kelas yang menghubungkan *model* dan *view*, digunakan untuk berkomunikasi antara kelas dalam *model* dan *view*. *Controller* mempunyai *action standar*. Ketika permintaan *user* tidak menetapkan *action* mana yang dijalankan, program akan menjalankan *action standar*.

## **2.12 Coronavirus**

Menurut Kemenkes RI (2020a), Coronavirus (CoV) adalah keluarga besar virus yang dapat menyebabkan penyakit mulai dari gejala ringan, sedang sampai berat. Virus corona adalah zoonosis (ditularkan antara hewan dan manusia). Penelitian menyebutkan bahwa SARS-CoV ditransmisikan dari kucing luwak (*civetcats*) ke manusia dan MERS-CoV dari unta ke manusia. Di akhir tahun 2019 telah muncul jenis virus corona baru yakni coronavirus disease 2019 (COVID-19).

### **2.12.1 Pengertian Coronavirus Disease 2019 (COVID-19)**

Menurut WHO (2020a), penyakit coronavirus disease 2019 (COVID-19) adalah penyakit menular yang disebabkan oleh virus corona yang baru ditemukan.

Kebanyakan orang yang terinfeksi virus COVID-19 akan mengalami penyakit pernapasan ringan hingga sedang dan sembuh tanpa memerlukan perawatan khusus. Orang tua dan orang-orang yang memiliki komorbid seperti penyakit kardiovaskular, diabetes, penyakit pernapasan kronis, dan kanker mungkin tertular COVID-19.

Coronavirus disease 2019 (COVID-19) adalah penyakit yang disebabkan oleh turunan coronavirus baru. 'CO' diambil dari corona, 'VI' virus, dan 'D' disease (penyakit). Sebelumnya, penyakit ini disebut '2019 novel coronavirus' atau '2019-nCoV.' Virus COVID-19 adalah virus baru yang terkait dengan keluarga virus yang sama dengan *Severe Acute Respiratory Syndrome* (SARS) dan beberapa jenis virus flu biasa (UNICEF, 2020). Menurut Sun et al., 2020, COVID-19 adalah penyakit coronavirus zoonosis ketiga yang diketahui setelah SARS dan sindrom pernapasan Timur Tengah (MERS). Menurut Gennaro et al., 2020, penyakit Virus Corona 2019 (COVID-19) adalah virus RNA, dengan penampakan seperti mahkota di bawah mikroskop elektron karena adanya paku glikoprotein pada amplopnya.

## **BAB III**

### **ANALISIS MASALAH DAN PERANCANGAN PROGRAM**

#### **3.1. Analisis Sistem**

Pada tahap analisis sistem dilakukan pengumpulan data dan menganalisa segala dokumen-dokumen yang digunakan pada sistem yang sedang berjalan. Semua ini dilakukan untuk memudahkan mengevaluasi kekurangan-kekurangan apa saja yang ada pada sistem tersebut yang selanjutnya merancang perbaikan-perbaikan pada sistem tersebut dan menyusun sistem baru. Selain itu juga dibahas analisis kebutuhan non-fungsional yang terdiri dari analisis perangkat keras dan analisis perangkat lunak.

##### **3.1.1. Studi Literatur**

Studi Literatur merupakan teknik yang digunakan untuk memperoleh informasi atau data dengan mempelajari buku-buku serta jurnal yang berkaitan dengan perancangan aplikasi sistem geografis pendaftaran vaksin. Berikut adalah tabel 3.1 yang menjadi referensi penelitian.

TABEL 3.1 : Studi Literatur

<b>No</b>	<b>Penulis</b>	<b>Judul</b>	<b>Pembahasan</b>
1	Acmad Nurhadi	SISTEM INFORMASI PENDAFTARAN VAKSINASI COVID-	Adapun tujuan dan manfaat dari penelitian ini adalah: 1. Tujuan a. Merancang sebuah sistem untuk Pendaftaran yang



		19	<p>masih menggunakan sistem offline menjadi sistem online dengan menerapkan ilmu pengetahuan yang telah kita dapatkan dari bangku perkuliahan. b. Agar dapat mengetahui perbandingan antara sistem yang manual dengan sistem yang sudah terkomputerisasi. 2. Manfaat a. Memberikan suatu kemudahan kepada masyarakat dalam sistem informasi pendaftaran vaksinasi agar lebih efisien dan efektif. b. Membantu mempermudah pekerjaan pihak yang menyelenggarakan vaksinasi dalam membuat laporan yang lebih efisien. c. Data pendaftar yang telah disimpan ke dalam system lebih aman dan akurat. d. Hasil perancangan sistem ini</p>
--	--	----	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

			<p>diharapkan dapat  memperlancar dan  memperbaiki proses  pendaftaran vaksinasi.  (Achmad Nurhadi, 2021)</p>
2	<p>Achmad  Mudhakhir, Dasvia  Yusherma, Falaah  Abdussalam, Irda  Sari</p>	<p>PERANCANGAN  SISTEM  INFORMASI  VAKSINASI COVID  19 DI  RSJ.dr.H.MARZOEKI  MAHDI BOGOR</p>	<p>Adapun tujuan dari  penelitian ini adalah  merancang Aplikasi Sistem  Informasi yang efektif untuk  pelayanan vaksin covid 19  menggunakan metode DBMS  dan perancangan terjadwal  bagi peserta vaksin di  RSJMM Bogor.Berdasarkan  uraian pada latar belakang,  masalah yang akan di bahas  pada penulisan ini adalah  bagaimana merancang sebuah  aplikasi Vaksinasi Covid 19  Rumah Sakit Jiwa dr. H.  Marzoeki Mahdi Bogor yang  memudahkan masyarakat</p>

			<p>           untuk mengakses dan melakukan pendaftaran serta memudahkan petugas pelaksana vaksinasi covid 19 dalam melakukan pelayanan dan pelaporan. Apakah aplikasi Vaksinasi Covid 19 ini dapat membantu masyarakat untuk memperoleh informasi pelaksanaan dan pendaftaran Vaksinasi di Rumah Sakit dr. H. Marzoeki Mahdi Bogor. Pada penulisan ini, masalah dibatasi dalam pembuatan Analisa dan perancangan aplikasi vaksinasi Covid 19 di Rumah Sakit Jiwa dr. H. Marzoeki Mahdi Bogor yang meliputi pelaksanaan vaksinasi Covid 19 yang menjadi objek dalam pembuatan aplikasi ini         </p>
--	--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

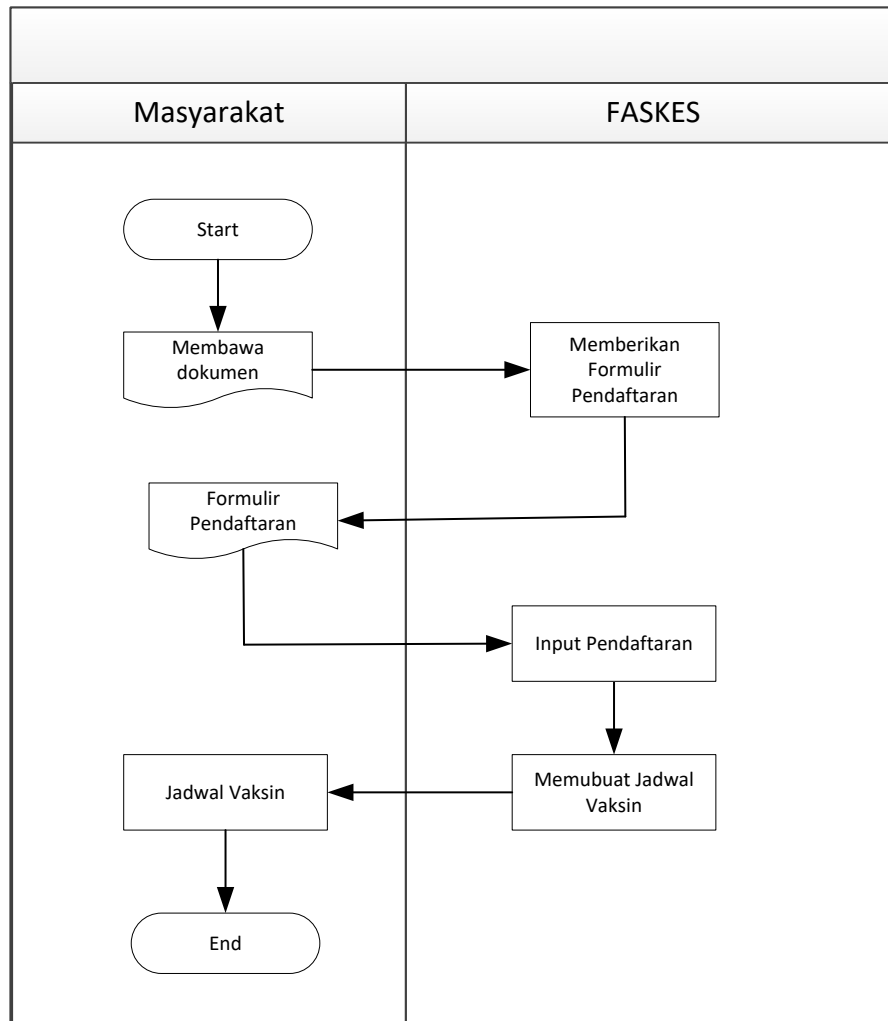
			<p>adalah pelaksanaan vaksinasi Covid 19 di Rumah Sakit Jiwa dr. H. Marzoeki Mahdi Bogor. (Mudhakhir, Yusherma, Abdussalam, Sari 2021)</p>
--	--	--	--------------------------------------------------------------------------------------------------------------------------------------------

Perbedaan penelitian ini dengan penelitian sejenis :

1. Studi kasus dilakukan ditempat yang berbeda
2. Web aplikasi ini dibuat menggunakan framework Codeigniter & API Mapbox
3. Pada penelitian ini aplikasi digunakan untuk menentukan lokasi pendaftaran vaksin yang terdekat.
4. Pada pembuatan aplikasi ini penulis memperhatikan faktor usability end user.

### **3.1.2. Analisis Sistem yang Sedang Berjalan**

Analisis sistem yang sedang berjalan berisi tentang pemaparan sistem informasi geografis lokasi dan pendaftaran vaksin covid 19 yang ada pada saat ini. Analisis ini dimaksudkan agar perangkat lunak yang dibangun tidak keluar dari cakupan sistem informasi geografis lokasi dan pendaftaran yang sudah ada. Berdasarkan penelitian yang telah dilakukan oleh penulis terdapat beberapa prosedur yang sedang berjalan.



**Gambar 3.1 Flowchart yang sedang berjalan**

### 3.1.3. Evaluasi Sistem Yang Berjalan

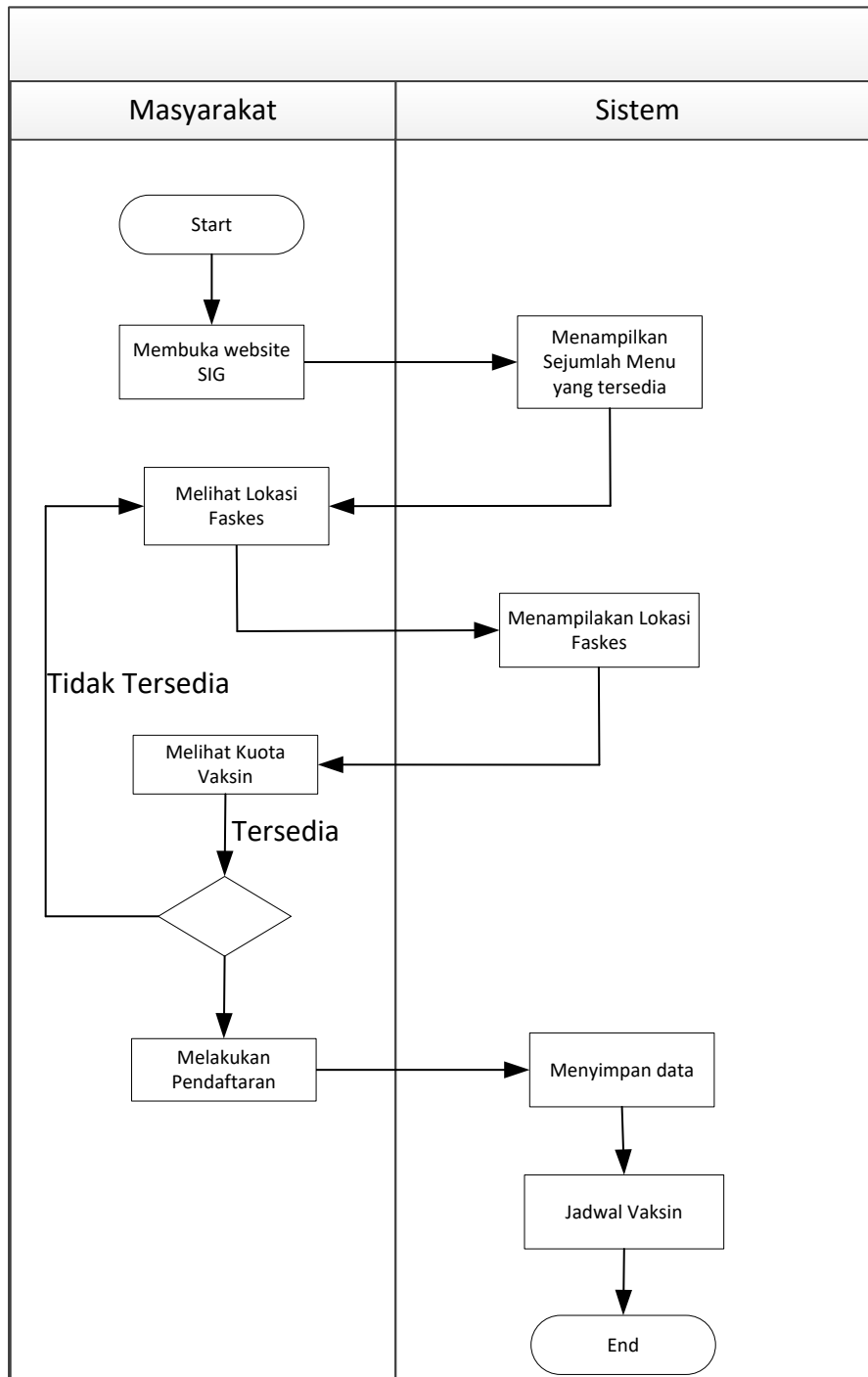
Sistem yang berjalan pada saat ini, apabila dilihat dari prosedur-prosedurnya, yang menjadi bahan evaluasi adalah prosedur pendaftaran serta lokasi vaksinasi yang kebanyakan masyarakat masih belum mengetahuinya, dimana masyarakat harus datang langsung ke lokasi fasilitas kesehatan (faskes) yang menyediakan vaksin untuk mendaftar. Ini mengakibatkan tidak efektifnya alur prosedur yang berjalan. Dikarenakan masyarakat harus datang langsung ke lokasi dan dimana ketika dilokasi faskes tersebut jumlah pendaftar sudah

terpenuhi maka masyarakat harus mencari lagi faskes yang masih menyediakan vaksin dan dalam pencarian tempat tersebut memakan waktu yang dapat menyebabkan masyarakat mengurungkan niat nya untuk segera divaksin.

#### **3.1.4. Solusi yang Diusulkan**

Dari permasalahan-permasalahan yang telah dijabarkan di atas, terdapat permasalahan dimana sistem yang ada sekarang ini belum terkomputerisasi dengan baik, terutama dalam tahap pendaftaran, pengecekan lokasi vaksinasi serta kuota yang tersedia di faskes. Oleh karena itu penulis mempunyai solusi yang diusulkan untuk memperbaharui sistem yang lama terutama didalam proses pendaftaran, pengecekan lokasi dan kuota vaksin di setiap faskes yang sudah terkomputerisasi tetapi masih belum optimal.

Ketika masyarakat ingin melakukan vaksin, cukup dengan melihat lokasi faskes mana yang diinginkan serta masyarakat juga bisa melihat langsung kuota yang tersedia pada faskes tersebut, sehingga ketika kuota pada faskes yang dituju sudah penuh, masyarakat tinggal memilih lokasi faskes lain yang masih memiliki kuota vaksin yang tersedia, selain itu juga masyarakat dapat melihat informasi – informasi mengenai covid 19 tentang jumlah kasus, jumlah orang yang sembuh serta program – program yang dilaksanakan oleh pemerintah untuk mengatasi penyebaran virus covid 19 ini.



**Gambar 3.2 Flowchat yang diusulkan**

### 3.1.5. Gambaran Umum Sistem yang Diusulkan

Aplikasi sistem berbasis *Web* yang diusulkan memiliki beberapa keunggulan dan perbedaan dari sistem yang sedang berjalan. Sistem yang diusulkan telah terotomatisasi sehingga lebih mudah digunakan, *integritas* data terjaga, tidak akan memakan waktu yang lama dalam mengolah data. Karena didalamnya telah disediakan pencetakan laporan-laporan dan fasilitas lainnya yang akan memudahkan *user* untuk menggunakan sistem ini.

Dalam pembuatan antarmuka, program aplikasi sistem berbasis *Web* ini menggunakan (contoh *Visual Studio Code* dan *MySQL* sebagai program aplikasi untuk membangun *database*-nya. Bahasa pemrograman yang digunakan dalam pembangunan aplikasi sistem berbasis *Web* ini menggunakan PHP).

### 3.1.6. Analisis Kebutuhan Sistem

Sistem yang dibangun akan diakses melalui komputer yang terkoneksi oleh *internet*. Untuk mendukung kecepatan pengaksesan dan kesempurnaan tampilan, maka membutuhkan perangkat lain sebagai pendukung agar berjalan sesuai fungsinya. Kebutuhan tersebut adalah :

#### 1. Kebutuhan Perangkat Keras

Beberapa spesifikasi perangkat keras yang digunakan diantaranya :

- a. *Processor* : *Pentium 4 1.8Gz*
- b. *RAM* : *1 Gb*
- c. *Harddisk* : *120 Gb*
- d. *Monitor* : *14 inch, Resolusi 1280 x 800 pixels*
- e. *VGA Card* : *512 Mb*



f. *Mouse*

g. *Keyboard*

## 2. Kebutuhan Perangkat Lunak

Sistem yang dibangun juga membutuhkan beberapa perangkat lunak dalam pengaksesannya, diantaranya :

a. Sistem operasi berbasis *windows (XP* atau *windows 7)*

b. *Web browser.*

c. *Aplikasi Server.*

### **3.2. Perancangan Sistem**

Perancangan sistem merupakan pengembangan sistem baru dari sistem lama yang sedang berjalan, dimana masalah-masalah yang terjadi pada sistem lama diharapkan sudah dapat teratasi pada sistem yang baru. Perancangan sistem merupakan tahapan setelah analisis dan persiapan untuk merancang dan membangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk, yang dapat berupa penggambaran, perancangan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi, juga menyangkut konfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

### **3.2.1. Tujuan Perancangan Sistem**

Tujuan dari perancangan sistem ini adalah untuk menghasilkan perancangan aplikasi sistem informasi geografis mengenai lokasi dan pendaftaran vaksin yang terkomputerisasi untuk memudahkan user untuk melakukan pencarian tempat vaksin serta langsung dapat melakukan pendaftaran. Dalam tahap perancangan sistem ini bertujuan untuk menghasilkan “RANCANG BANGUN APLIKASI SISTEM INFORMASI GEOGRAFIS LOKASI DAN PENDAFTARAN VAKSIN COVID-19 BERBASIS WEB DI KOTA CIMAHI”.

Adapun komponen-komponen sistem informasi yang dirancang ini meliputi :

1. Perancangan Proses (*Flowmap, usecase, activity diagram, sequence diagram dan class diagram*).
2. Perancangan *Basisdata* (Tabel Relasi).
3. Perancangan Program (Perancangan Arsitektur, Perancangan Antar Muka).

### **3.2.2. Perancangan Prosedur yang Diusulkan**

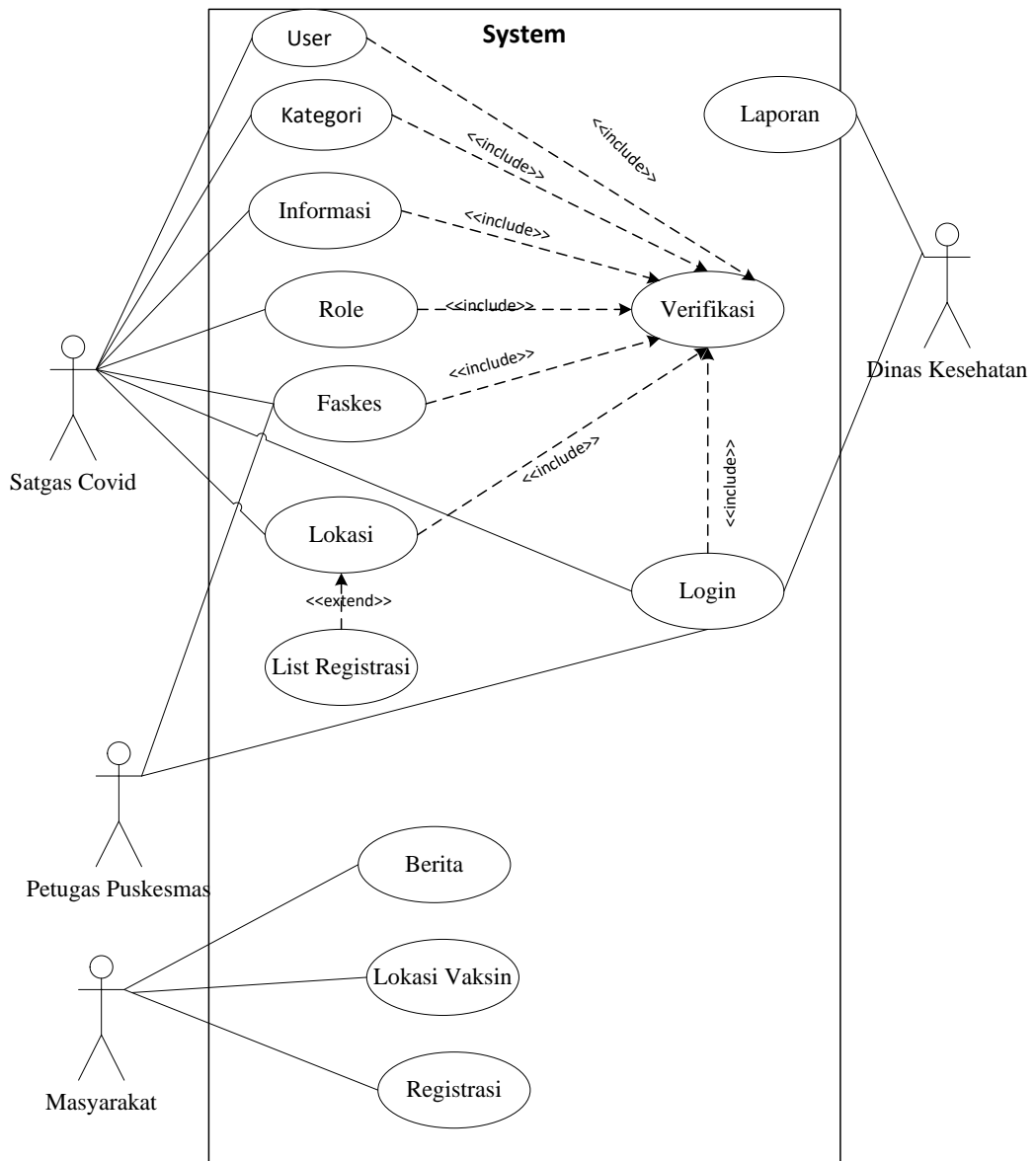
Adapun prosedur sistem informasi geografis mengenai lokasi dan pendaftaran vaksin yang diusulkan dalam sistem baru ini, yang dapat menggambarkan secara teori dan singkat sebagai berikut :

1. Aplikasi sistem informasi geografis akan memberikan informasi mengenai nama tempat serta lokasi yang menyelenggarakan vaksin.
2. Masyarakat dapat langsung mendaftar untuk di vaksin sesuai dengan faskes yang dituju serta masyarakat dapat juga melihat jumlah kuota yang tersedia.

3. Setelah proses *input* data selesai maka dilakukan simpan data, selanjutnya apabila ada laporan yang harus dicetak maka langsung mencetaknya karena sudah tersedia *button* untuk mencetak laporan, setelah *button* di klik maka akan terlihat tampilan laporan yang akan dicetak.

### **3.2.3. Use Case Diagram**

Berdasarkan analisis yang telah dijelaskan sebelumnya, maka didapat sebuah *use case* diagram yang menjelaskan hubungan antara fungsi yang disediakan oleh sistem, seperti yang terdapat pada gambar sebagai berikut ini:



**Gambar 3.3. Usecase Diagram**

**Tabel 3.2 Usecase Login**

<b>Identifikasi</b>	
Nama Usecase	Login
Aktor	Satgas Covid, Petugas Puskesmas dan Dinas Kesehatan
Tujuan	Mendapatkan akses menuju halaman manajemen data

Alur	1. Sistem difasilitasi oleh halaman Login untuk sebuah keamanan
	2. Pengguna sistem dapat lebih aman dan terjaga dalam menikmati bentuk informasi yang ditampilkan
	3. Satgas Covid berhak melakukan proses manipulasi data selama mempunyai hak akses ke dalam aplikasi setelah memenuhi proses validasi.

**Tabel 3.3 Usecase Lokasi**

<b>Identifikasi</b>	
Nama Usecase	Lokasi
Aktor	Satgas Covid dan masyarakat
Tujuan	Untuk menyediakan informasi mengenai lokasi faskes
Alur	1. Satgas Covid dan masyarakat dapat data melihat data lokasi faskes
	2. Satgas Covid dapat memanipulasi data lokasi faskes

**Tabel 3.4 Usecase Role**

<b>Identifikasi</b>	
Nama Usecase	Role
Aktor	Satgas Covid
Tujuan	Untuk menyediakan informasi mengenai role vaksin
Alur	1. Satgas Covid dapat melihat role
	2. Sistem menampilkan data role, serta Satgas Covid dapat memanipulasi data role

**Tabel 3.5 Usecase Faskes**

<b>Identifikasi</b>	
Nama Usecase	Lokasi
Aktor	Satgas Covid dan Petugas Puskesmas
Tujuan	Untuk menyediakan informasi mengenai data faskes
Alur	1. Satgas Covid dapat melakukan penambahan data, edit data dan menghapus data faskes
	2. Petugas Puskesmas dapat melihat faskes yang menyelenggarakan vaksinasi

**Tabel 3.6 Usecase Laporan**

<b>Identifikasi</b>	
Nama Usecase	Data Laporan
Aktor	Dinas Kesehatan
Tujuan	Untuk menyediakan informasi mengenai laporan
Alur	1. Dinas Kesehatan dapat melihat laporan yang tersedia

**Tabel 3.7 Usecase Berita**

<b>Identifikasi</b>	
Nama Usecase	Berita
Aktor	Satgas Covid dan Masyarakat
Tujuan	Untuk menyediakan seluruh informasi mengenai berita, kegiatan dan infografis
Alur	1. Satgas Covid dapat menambah, mengedit dan menghapus berita. 2. Masyarakat dapat melihat berbagai informasi yang tersedia.

**Tabel 3.8 Usecase Registrasi**

<b>Identifikasi</b>	
Nama Usecase	Registrasi
Aktor	Masyarakat
Tujuan	Agar masyarakat dapat terdata untuk dilakukan vaksin
Alur	1. Masyarakat dapat melakukan pendaftaran agar tercatat sebagai penerima vaksin. 2. Sistem menampilkan bukti bahwa

<b>Identifikasi</b>	
	masyarakat sudah terdaftar, dan bukti tersebut dapat di <i>print out</i> .

**Tabel 3.9 Usecase Kategori**

<b>Identifikasi</b>	
Nama Usecase	Kategori
Aktor	Satgas Covid
Tujuan	Untuk menyediakan seluruh kategori berita yang ada.
Alur	<ol style="list-style-type: none"> <li>1. Satgas Covid dapat menambah, mengedit dan menghapus kategori.</li> <li>2. Sistem menampilkan data kategori</li> </ol>

**Tabel 4.0 Usecase List Registrasi**

<b>Identifikasi</b>	
Nama Usecase	List Registrasi
Aktor	Satgas Covid & Petugas Puskesmas
Tujuan	Untuk melihat list registrasi vaksin setiap lokasi
Alur	<ol style="list-style-type: none"> <li>1. Satgas Covid dan Petugas Puskesmas dapat merubah status user yang teregistrasi</li> </ol>

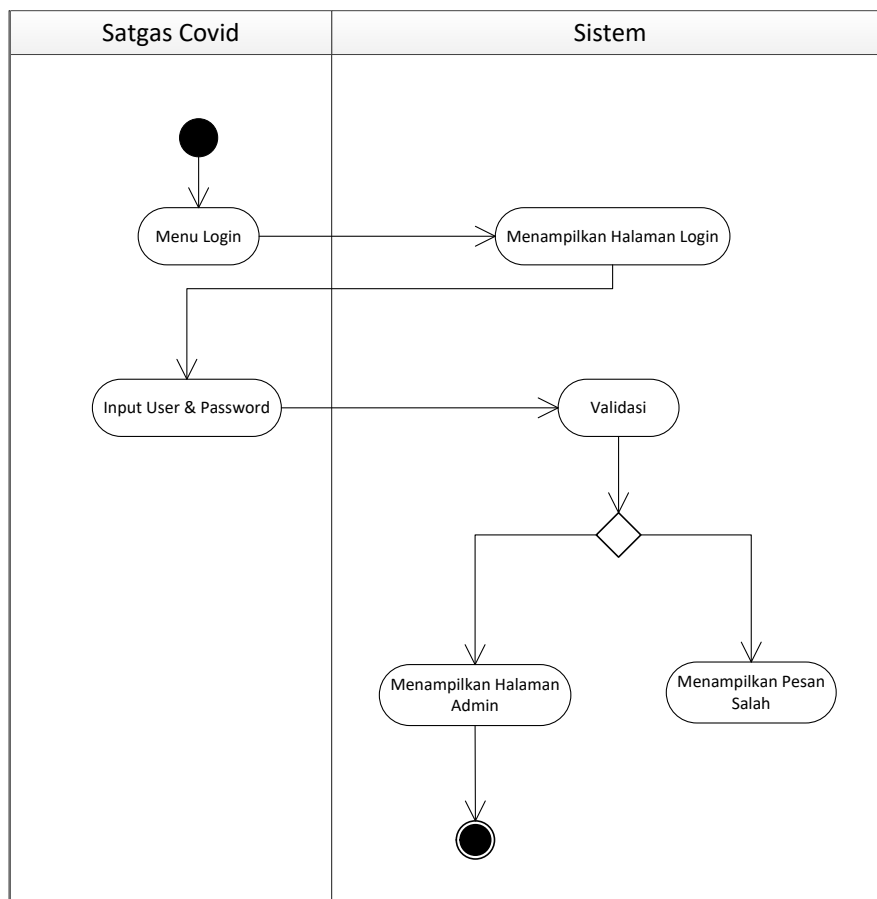


### 3.2.4. Activity Diagram

*Activity Diagram* menggambarkan berbagai alur aktivitas dalam aplikasi yang dirancang, bagaimana masing-masing alur berawal, keputusan yang mungkin terjadi dan bagaimana mereka berakhir.

#### 1. Activity Diagram login

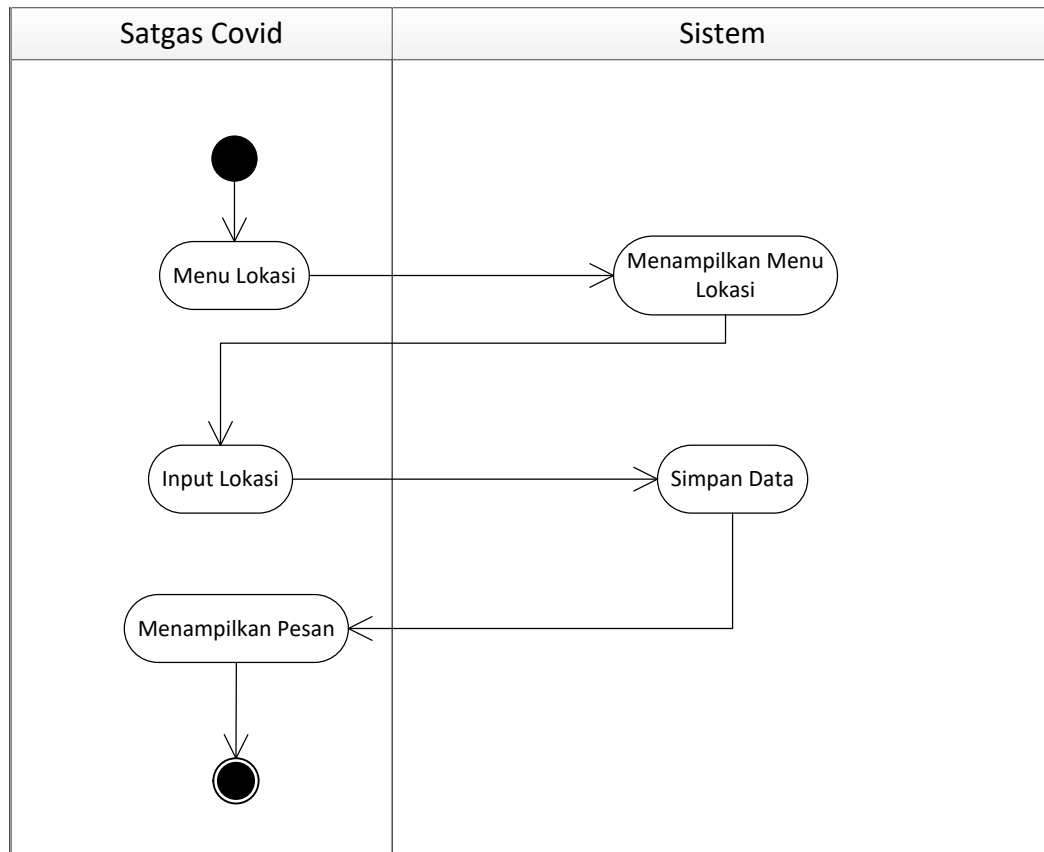
Proses pada Gambar 3.4 menggambarkan aktifitas pada saat login menuju menu pengelolaan data harus melakukan login sebelum melakukan pengelolaan data. Tujuan login disini agar tidak semua orang dapat mengakses menu manajemen data pada aplikasi.



**Gambar 3.4. Activity Diagram Login**

## 2. Activity Diagram Lokasi

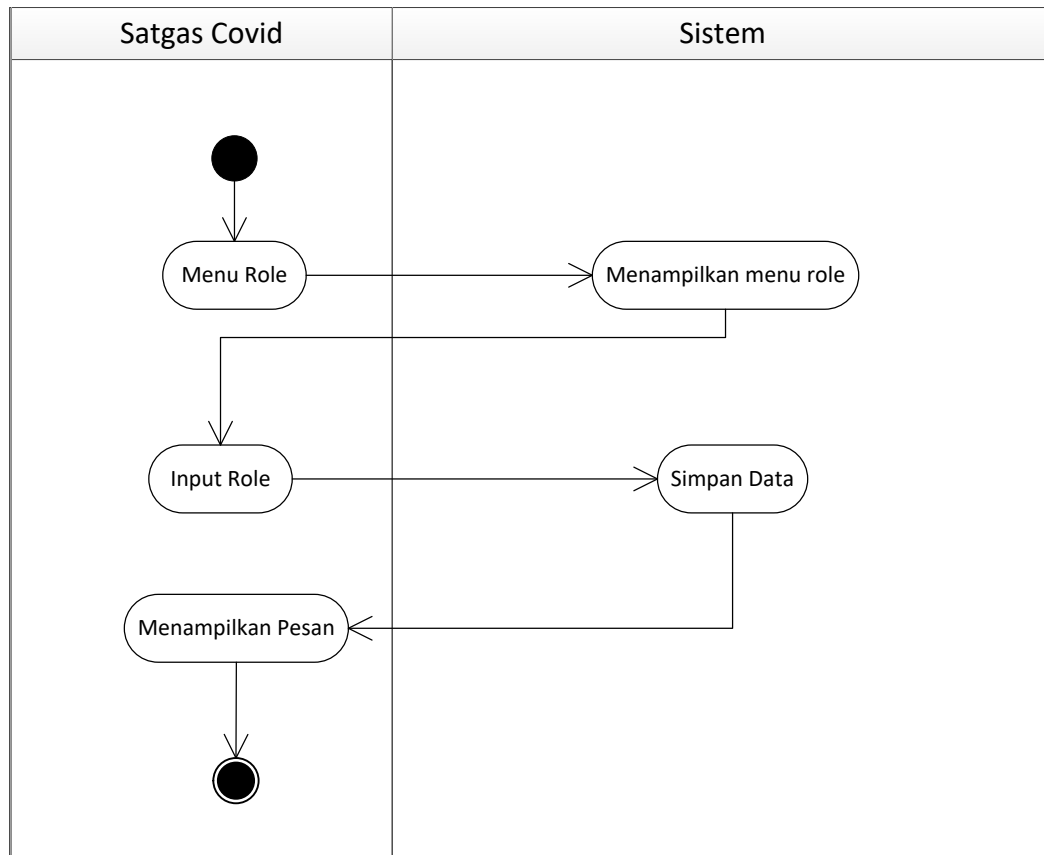
Proses pada Gambar 3.5 menggambarkan aktifitas pada saat melihat lokasi. Pada menu ini Satgas Covid dapat melihat dan menginputkan data pada aplikasi.



**Gambar 3.5 Activity Diagram Lokasi**

## 3. Activity Diagram Role

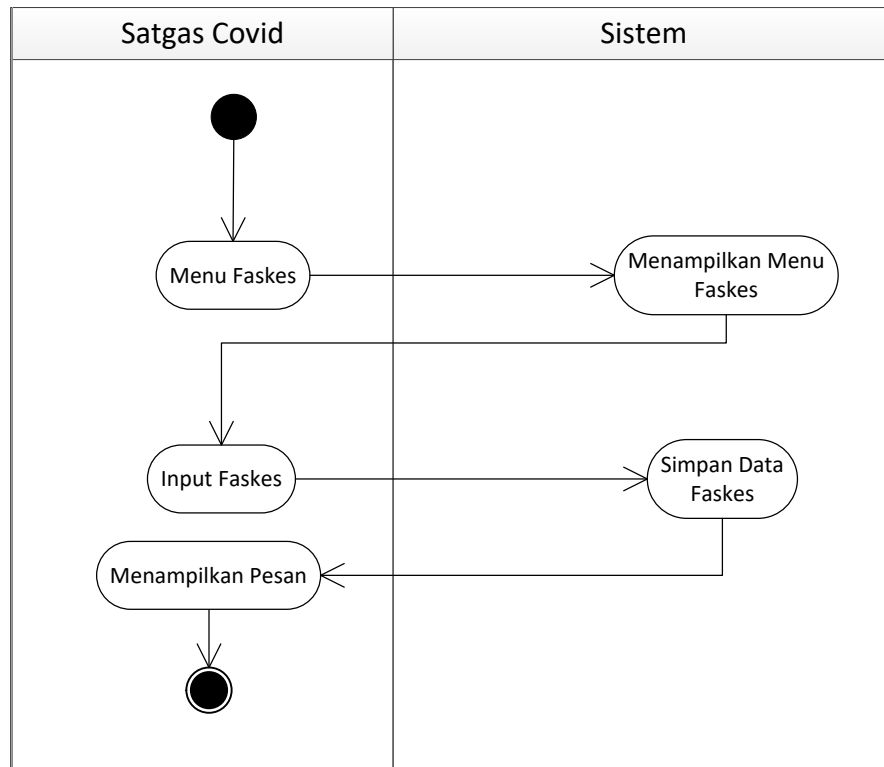
Proses pada Gambar 3.6 menggambarkan aktifitas pada saat mengelola atau memanipulasi role. Dalam menu ini dapat menambah, mengubah atau menghapus data sesuai kehendak Satgas Covid. Activity Diagram-nya sebagai berikut :



**Gambar 3.6 Activity Diagram Role**

4. *Activity Diagram* Data Faskes

Proses pada Gambar 3.7 menggambarkan aktifitas pada saat mengelola atau memanipulasi data faskes. Dalam menu ini dapat menambah, mengubah atau menghapus data transaksi. *Activity Diagram*-nya sebagai berikut :

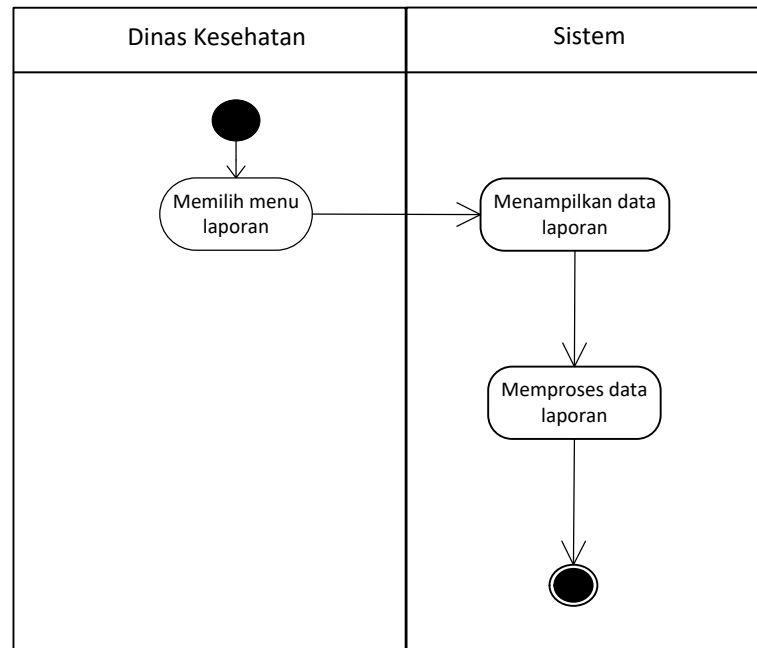


**Gambar 3.7 Activity Diagram Data Faskes**

5. Activity Diagram Laporan

Proses pada Gambar 3.8 menggambarkan aktifitas pada saat melihat laporan. Dalam menu ini dapat melihat laporan data jumlah yang telah divaksin.

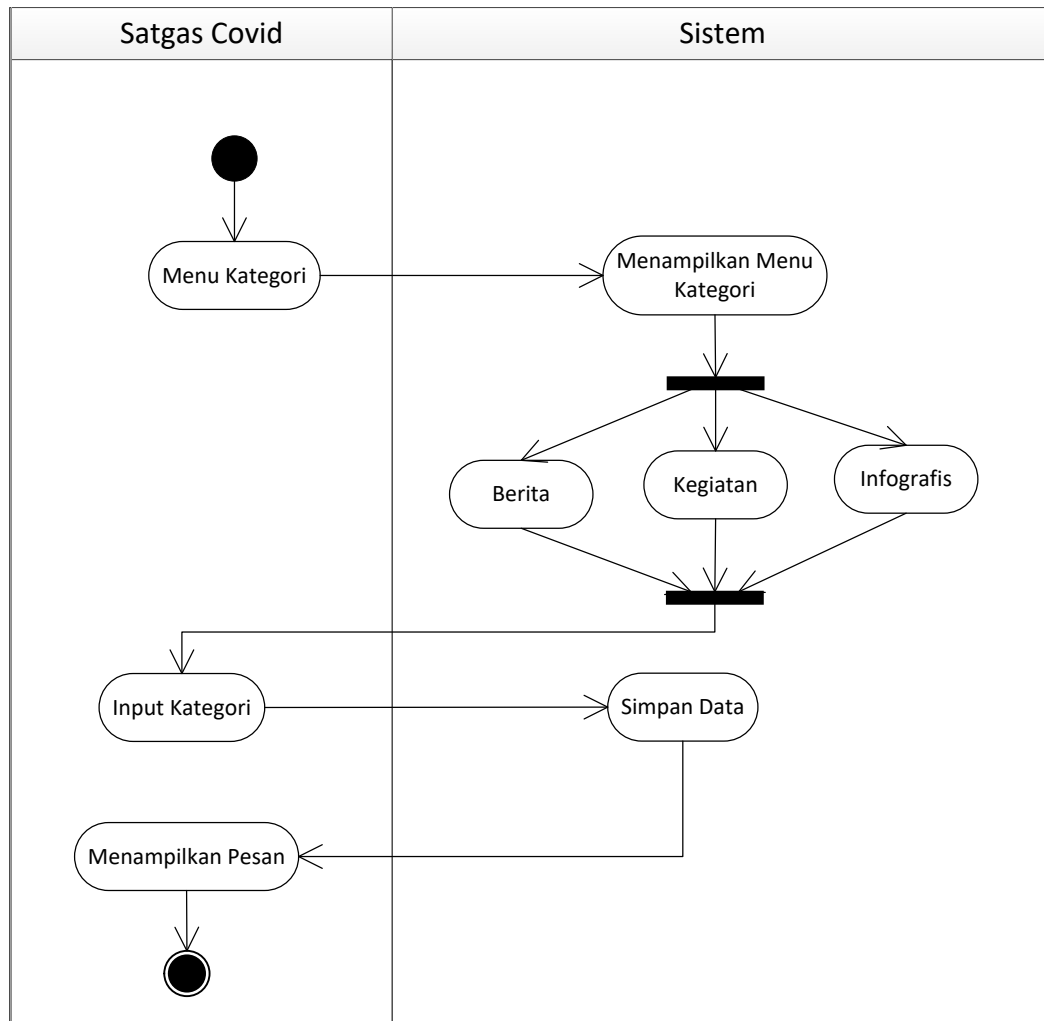
Activity Diagram-nya sebagai berikut :



**Gambar 3.8 Activity Diagram Laporan**

6. *Activity Diagram* Kategori

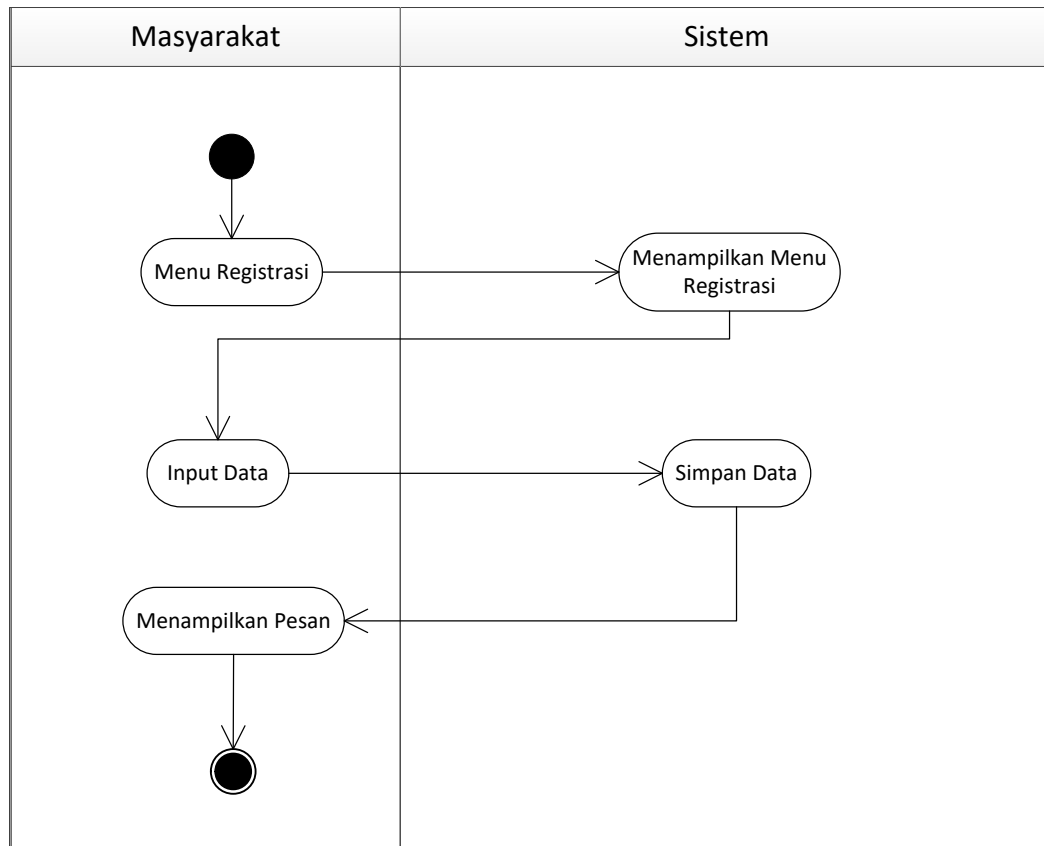
Proses pada Gambar 3.9 menggambarkan aktifitas pada saat melihat kategori. Dalam menu ini dapat melihat, menambah, dan mengedit data berita yang didalamnya terdapat berita, kegiatan dan infografis. *Activity Diagram*-nya sebagai berikut:



**Gambar 3.9 Activity Diagram Kategori**

7. *Activity Diagram* Registrasi

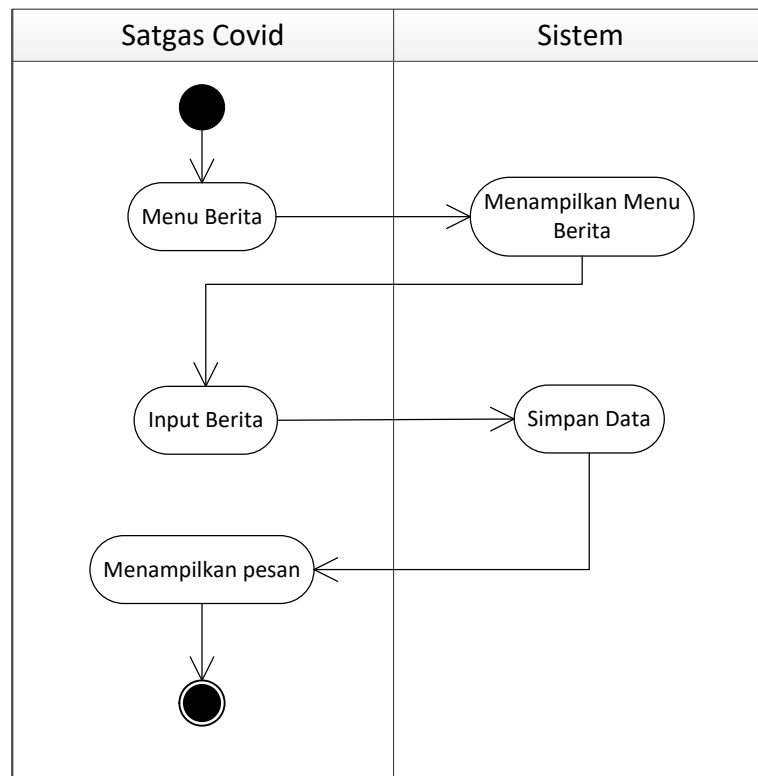
Proses pada Gambar 3.10 menggambarkan aktifitas pada saat melakukan registrasi. Dalam menu ini dapat melakukan proses pendaftaran. *Activity Diagram*-nya sebagai berikut :



**Gambar 3.10 Activity Diagram Registrasi**

8. *Activity Diagram* Berita

Proses pada Gambar 3.11 menggambarkan aktifitas pada saat melakukan berita. Dalam menu ini dapat memberikan informasi mengenai berita. *Activity Diagram*-nya sebagai berikut :

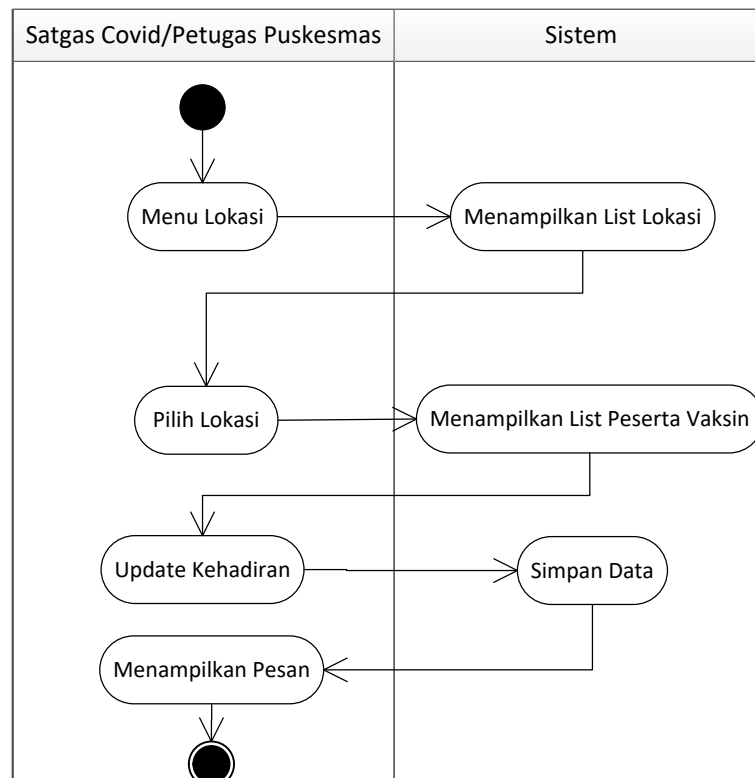


**Gambar 3.11 Activity Diagram Berita**

9. *Activity Update Kehadiran*

Proses pada Gambar 3.12 menggambarkan aktifitas pada saat melakukan update kehadiran peserta vaksin. Dalam menu ini dapat menampilkan list peserta vaksin per lokasi. *Activity Diagram*-nya sebagai berikut :



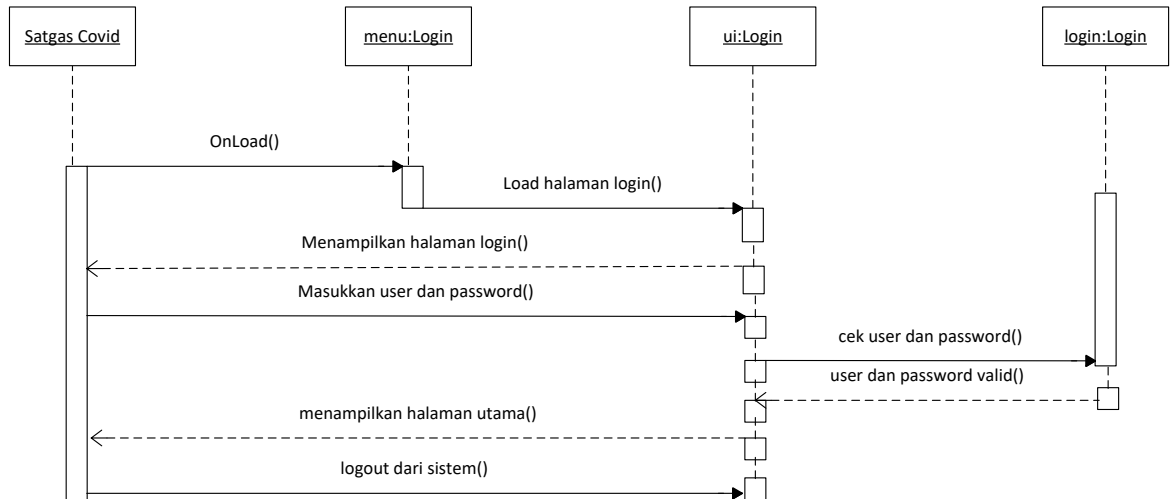


**Gambar 3.12 Activity Update Kehadiran**

### 3.2.5 Sequence Diagram

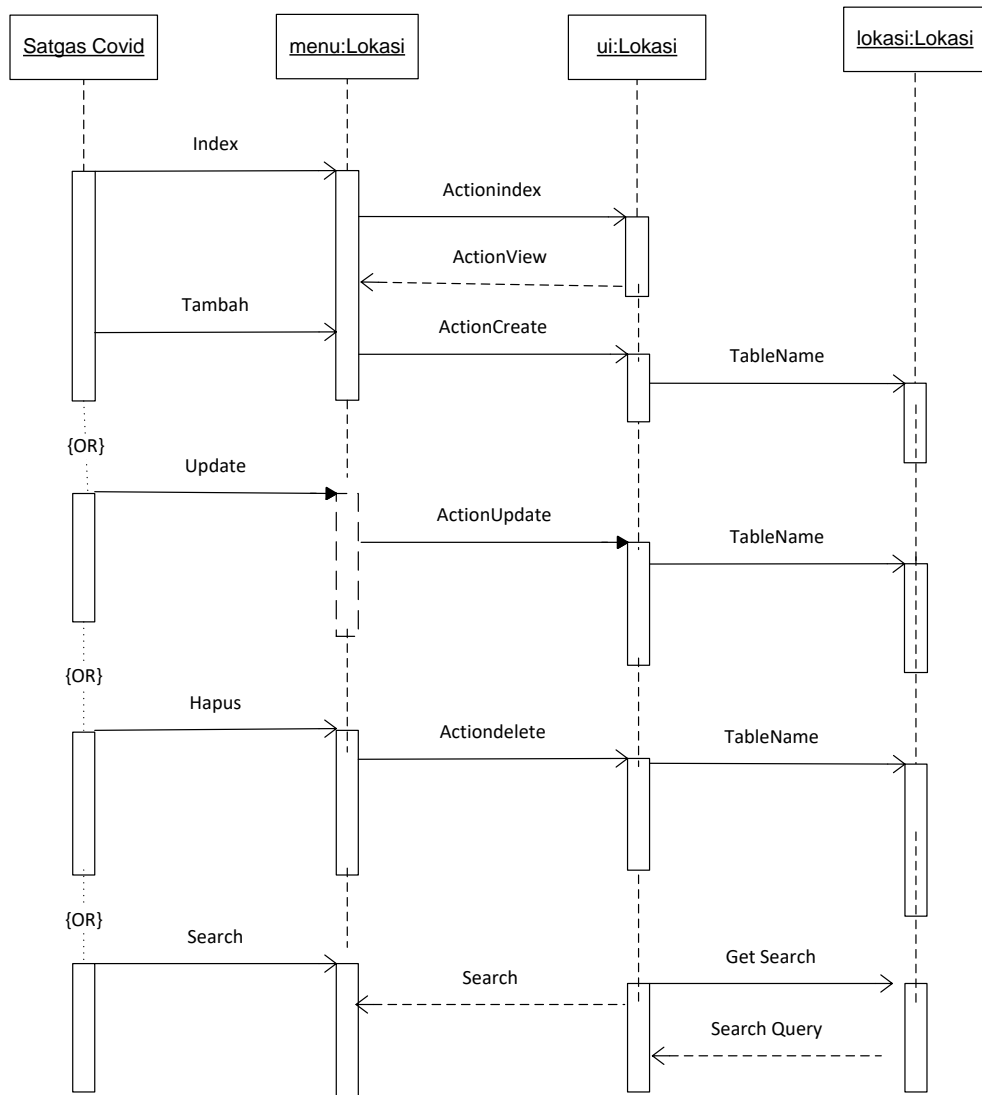
Merupakan bentuk dari interaksi diagram, menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa pesan yang di gambarkan terhadap waktu. Masing-masing objek termasuk aktor memiliki *line vertical*, pesan digambarkan garis berpanah dari satu objek ke objek lainnya, dan *activation bar* menunjukkan lamanya eksekusi sebuah proses. Sebuah *sequence diagram*, secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek dan pesan-pesan yang melewati objek- objek ini di dalam *use case*. Berikut ini adalah *sequence diagram* dari aplikasi yang dibangun.

Pada gambar 3.13 menggambarkan *sequence diagram login* yang dilakukan oleh Satgas Covid.



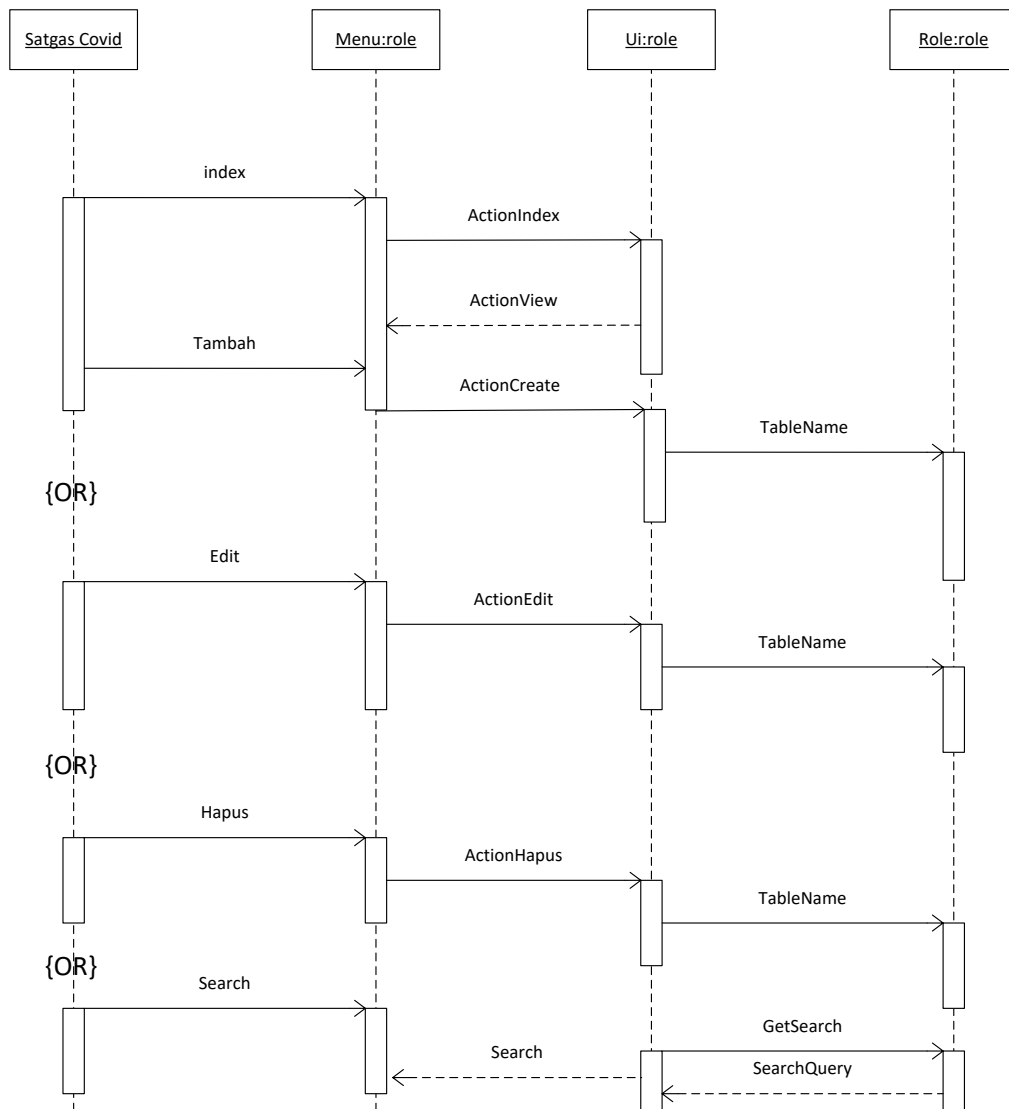
**Gambar 3.13 Sequence Diagram Login**

Pada gambar 3.14 menggambarkan *sequence diagram lokasi* yang dilakukan oleh Satgas Covid.



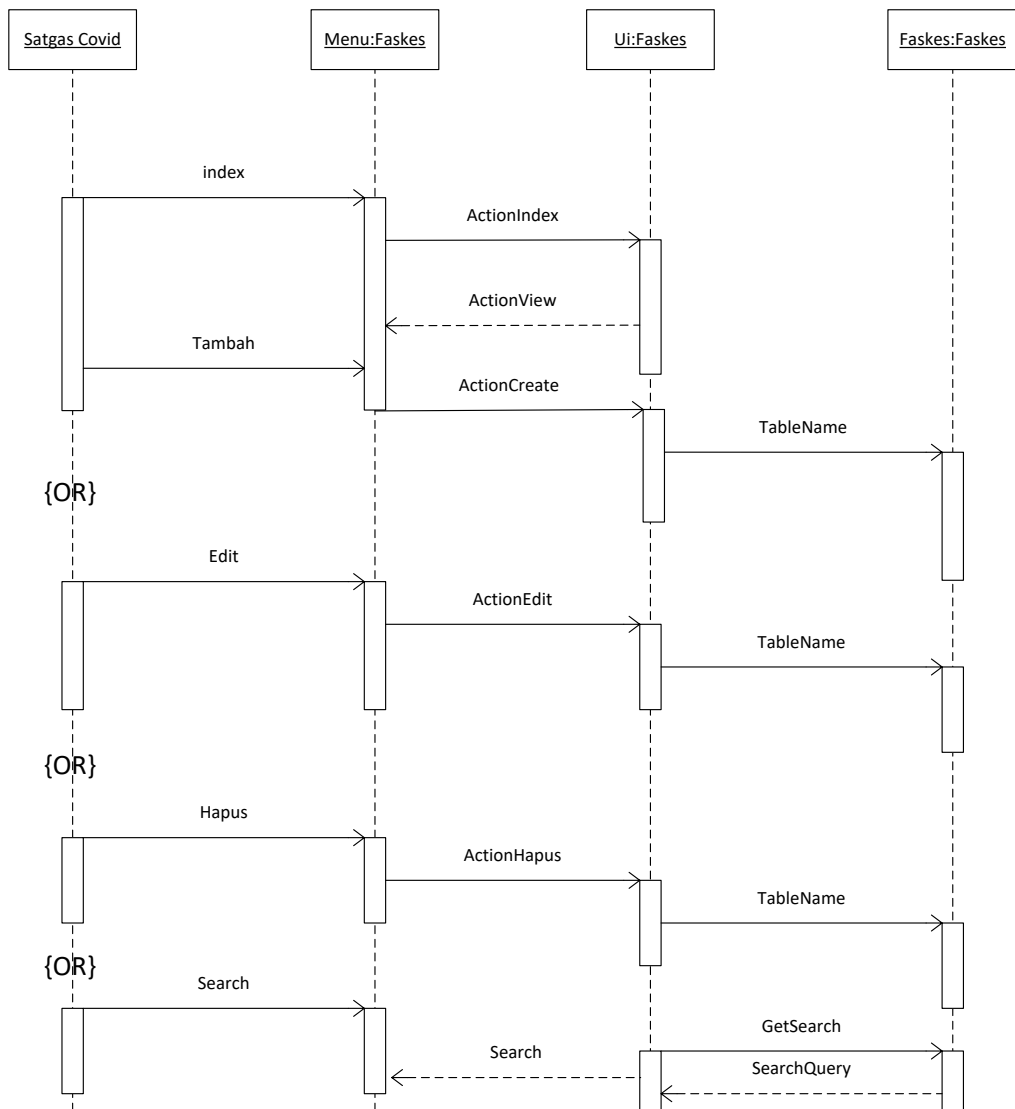
**Gambar 3.14 Sequence Diagram Lokasi**

Pada gambar 3.15 menggambarkan *sequence diagram* lihat role yang dilakukan oleh Satgas Covid.



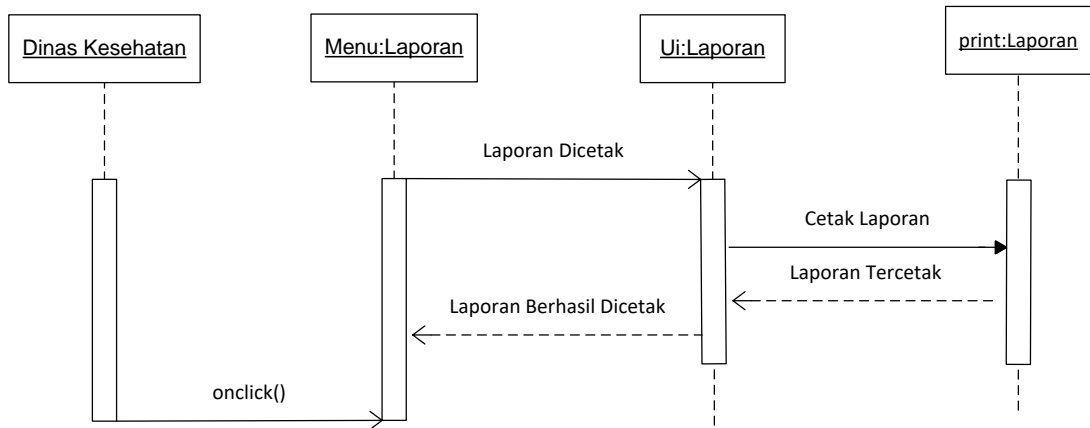
**Gambar 3.15 Sequence Diagram Role**

Pada gambar 3.16 menggambarkan *sequence diagram* lihat data Faskes yang dilakukan oleh Satgas Covid.



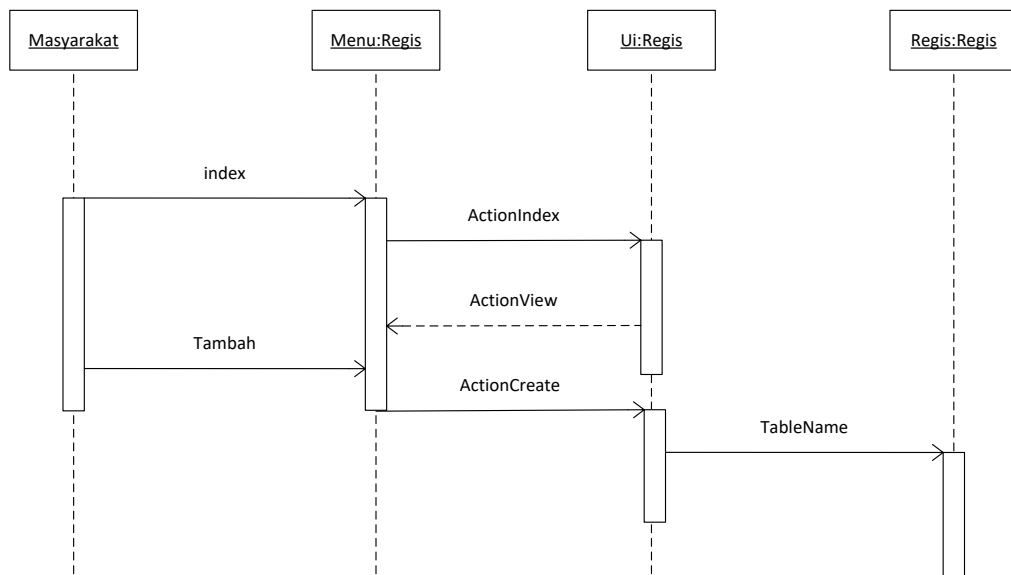
**Gambar 3.16 Sequence Diagram Faskes**

Pada gambar 3.17 menggambarkan *sequence diagram* lihat laporan yang dilakukan oleh Dinas Kesehatan.



**Gambar 3.17 Sequence Diagram Laporan**

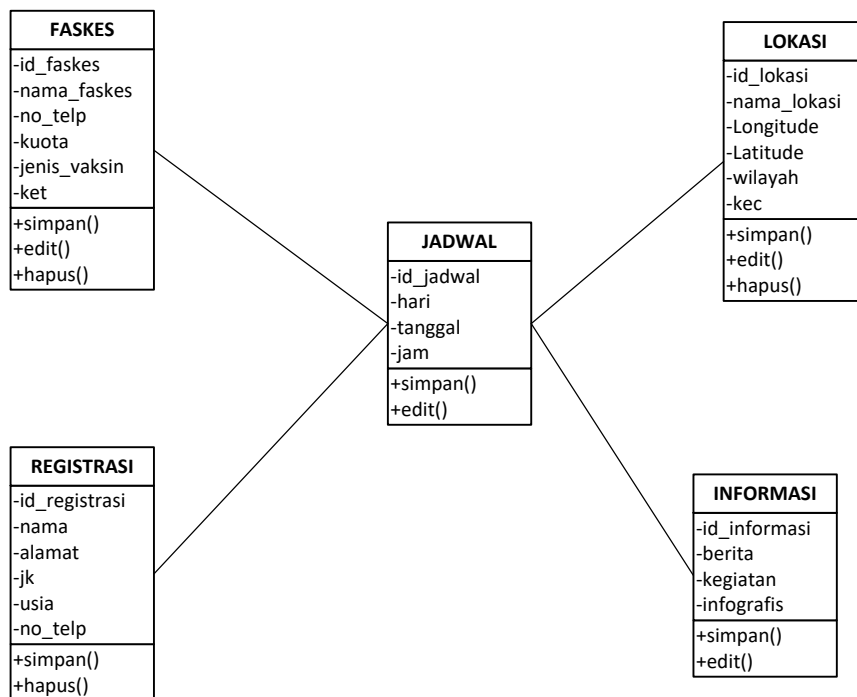
Pada gambar 3.18 menggambarkan *sequence diagram* lihat Registrasi yang dilakukan oleh masyarakat.



**Gambar 3.18 Sequence Diagram Registrasi**

### 3.2.6 Class Diagram

*Class diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*attribut* atau *property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metoda* atau *fungsi*). Berikut *class diagram* dari Rancang Bangun Aplikasi Sistem Informasi Geografis Lokasi dan Pendaftaran Vaksin Covid-19 Berbasis Web di Kota Cimahi pada gambar 3.19.

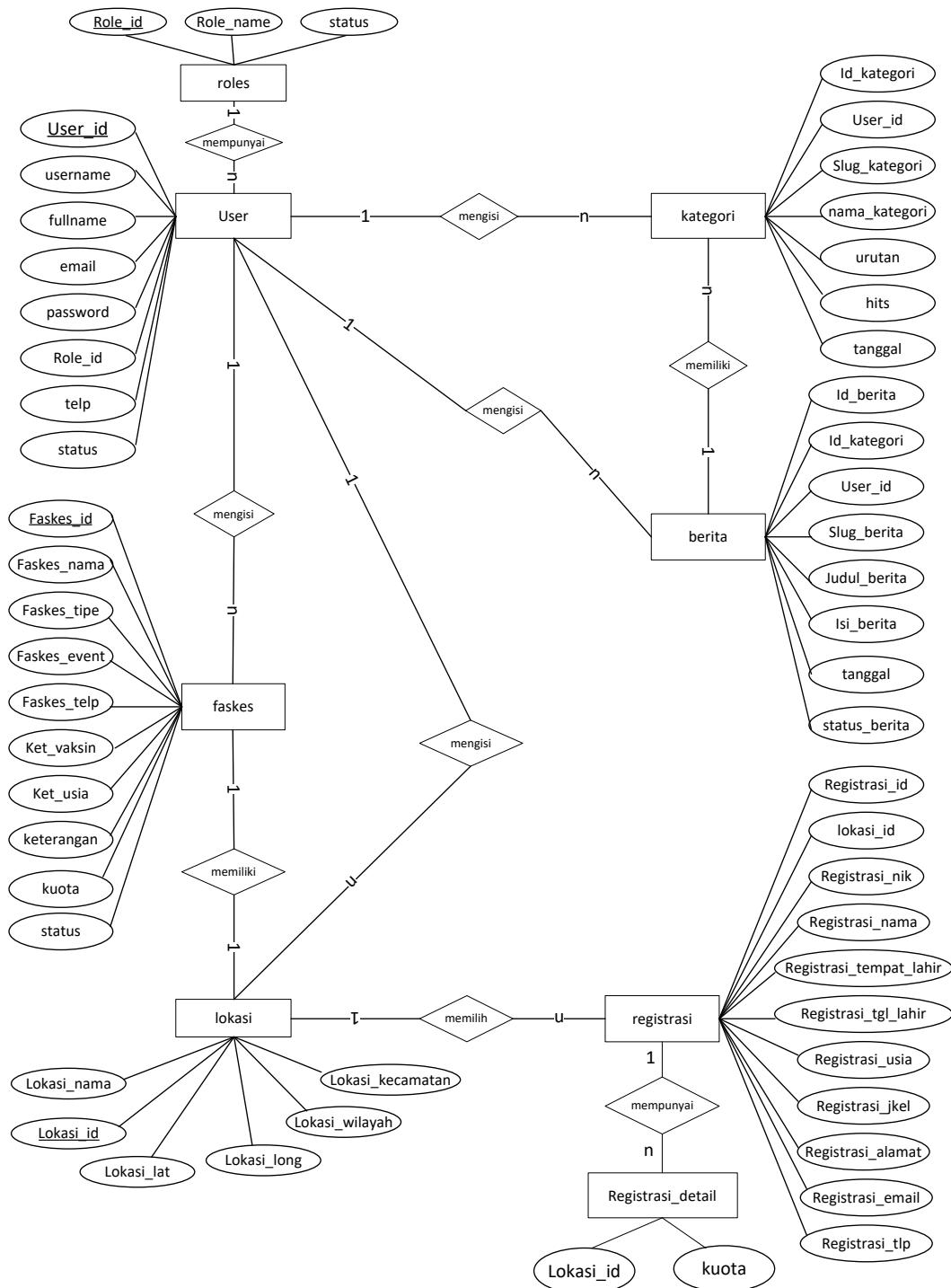


**Gambar 3.19 Class Diagram**

### 3.2.7 ERD (Entity Relationship Diagram)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan

antar relasi. Pada Gambar 3.20 merupakan rancangan ERD pada aplikasi komputerisasi pengarsipan *document hard copy* menggunakan *web application*.



**Gambar 3.20 ERD (Entity Relationship Diagram)**



### 3.2.8 Perancangan Struktur Database

Di dalam database terdapat beberapa table yang digunakan nantinya di dalam pengoperasian sistem absensi ini. Tabel-tabel tersebut akan disebutkan sebagai berikut:

**Tabel 3.10 Login**

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
123 user_id	1	int(10)	[v]	[v]	PRI	
ABC username	2	varchar(20)	[v]	[ ]		
ABC fullname	3	varchar(50)	[v]	[ ]		
ABC email	4	varchar(50)	[v]	[ ]		
ABC password	5	varchar(100)	[v]	[ ]		
123 role_id	6	int(10)	[v]	[ ]		
ABC telp	7	varchar(20)	[ ]	[ ]		NULL
123 status	8	int(11)	[v]	[ ]		
123 deleted	9	int(11)	[v]	[ ]		
ABC created_by	10	varchar(20)	[ ]	[ ]		NULL
created_date	11	timestamp	[ ]	[ ]		NULL
ABC updated_by	12	varchar(20)	[ ]	[ ]		NULL
updated_date	13	timestamp	[ ]	[ ]		NULL

**Tabel 3.11 Lokasi**

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
123 lokasi_id	1	int(10)	[v]	[v]	PRI	
ABC lokasi_nama	2	varchar(255)	[v]	[ ]		
ABC lokasi_lat	3	varchar(50)	[v]	[ ]		
ABC lokasi_long	4	varchar(50)	[v]	[ ]		
ABC lokasi_wilayah	5	varchar(50)	[v]	[ ]		
ABC lokasi_kecamatan	6	varchar(50)	[v]	[ ]		
123 deleted	7	int(11)	[v]	[ ]		
ABC created_by	8	varchar(20)	[ ]	[ ]		NULL
created_date	9	timestamp	[ ]	[ ]		NULL
ABC updated_by	10	varchar(20)	[ ]	[ ]		NULL
updated_date	11	timestamp	[ ]	[ ]		NULL
123 faskes_id	12	int(10)	[v]	[ ]		
ABC lokasi_alamat	13	varchar(255)	[v]	[ ]		

**Tabel 3.12 Role**

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
123 role_id	1	int(15)	[v]	[v]	PRI	
ABC role_name	2	varchar(20)	[v]	[ ]		
ABC created_by	3	varchar(20)	[ ]	[ ]		NULL
🕒 created_date	4	timestamp	[ ]	[ ]		NULL
ABC updated_by	5	varchar(20)	[ ]	[ ]		NULL
🕒 updated_date	6	timestamp	[ ]	[ ]		NULL
123 status	7	int(10)	[ ]	[ ]		NULL

**Tabel 3.13 Kategori**

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
123 id_kategori	1	int(11)	[v]	[v]	PRI	
123 user_id	2	int(11)	[v]	[ ]		
ABC slug_kategori	3	varchar(255)	[v]	[ ]		
ABC nama_kategori	4	varchar(255)	[v]	[ ]		
123 urutan	5	int(11)	[ ]	[ ]		NULL
123 hits	6	int(11)	[v]	[ ]		
🕒 tanggal	7	timestamp	[v]	[ ]		current_timestamp()

**Tabel 3.14 Berita**

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
123 id_berita	1	int(11)	[v]	[v]	PRI	
123 user_id	2	int(11)	[v]	[ ]		
123 id_kategori	3	int(11)	[v]	[ ]		
ABC slug_berita	4	varchar(255)	[v]	[ ]		
ABC judul_berita	5	varchar(255)	[v]	[ ]		
ABC ringkasan	6	varchar(500)	[v]	[ ]		
ABC isi	7	text	[v]	[ ]		
ABC status_berita	8	varchar(20)	[v]	[ ]		
ABC jenis_berita	9	varchar(20)	[v]	[ ]		
ABC keywords	10	text	[v]	[ ]		
ABC gambar	11	varchar(255)	[ ]	[ ]		NULL
ABC icon	12	varchar(255)	[ ]	[ ]		NULL
123 hits	13	int(11)	[v]	[ ]		
🕒 tanggal_post	14	datetime	[v]	[ ]		
🕒 tanggal_publish	15	datetime	[v]	[ ]		
🕒 tanggal	16	timestamp	[v]	[ ]		current_timestamp()

**Tabel 3.15 Faskes**

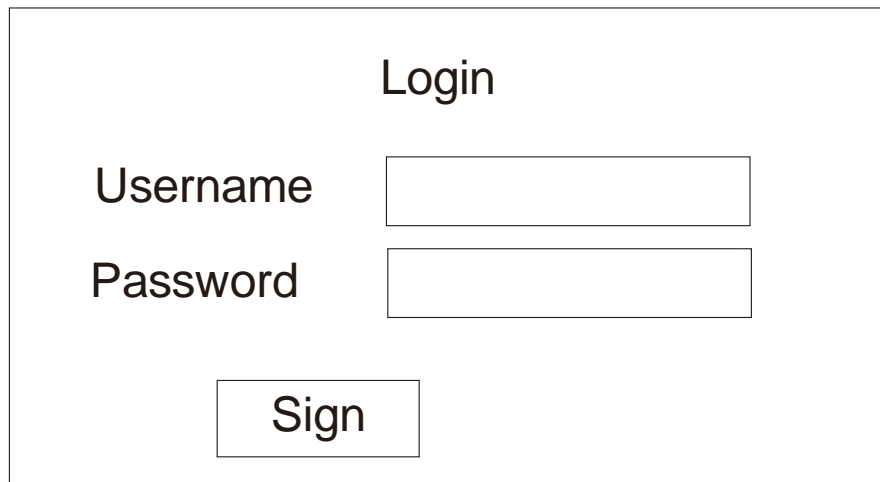
Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
faskes_id	1	int(10)	[v]	[v]	PRI	
faskes_nama	2	varchar(100)	[v]	[ ]		
faskes_tipe	3	varchar(20)	[v]	[ ]		
faskes_event	4	varchar(100)	[v]	[ ]		
faskes_telp	5	varchar(20)	[ ]	[ ]		NULL
ket_vaksin	6	text	[v]	[ ]		
ket_usia	7	text	[v]	[ ]		
keterangan	8	text	[v]	[ ]		
kuota	9	varchar(50)	[v]	[ ]		
reg_akhir	10	date	[v]	[ ]		
tgl_awal	11	date	[v]	[ ]		
tgl_akhir	12	date	[v]	[ ]		
status	13	int(11)	[v]	[ ]		
created_by	14	varchar(20)	[ ]	[ ]		NULL
created_date	15	timestamp	[ ]	[ ]		NULL
updated_by	16	varchar(20)	[ ]	[ ]		NULL
updated_date	17	timestamp	[ ]	[ ]		NULL
faskes_poster	18	varchar(255)	[ ]	[ ]		NULL
faskes_email	19	varchar(50)	[ ]	[ ]		NULL
deleted	20	int(11)	[v]	[ ]		
jam_awal	21	time	[v]	[ ]		
jam_akhir	22	time	[v]	[ ]		

**Tabel 3.16 Registrasi**

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
registrasi_id	1	int(10)	[v]	[v]	PRI	
lokasi_id	2	int(10)	[v]	[ ]		
registrasi_nik	3	varchar(16)	[v]	[ ]		
registrasi_nama	4	varchar(100)	[v]	[ ]		
registrasi_tempat_lahir	5	varchar(50)	[v]	[ ]		
registrasi_tgl_lahir	6	date	[v]	[ ]		
registrasi_usia	7	varchar(20)	[v]	[ ]		
registrasi_jkel	8	varchar(20)	[v]	[ ]		
registrasi_alamat	9	varchar(255)	[v]	[ ]		
registrasi_email	10	varchar(50)	[ ]	[ ]		NULL
registrasi_telp	11	varchar(20)	[v]	[ ]		
created_date	12	timestamp	[ ]	[ ]		NULL
registrasi_no	13	varchar(100)	[v]	[ ]		
status	14	int(11)	[v]	[ ]		
updated_by	15	varchar(20)	[ ]	[ ]		NULL
updated_date	16	timestamp	[ ]	[ ]		NULL



## 1. Halaman Login



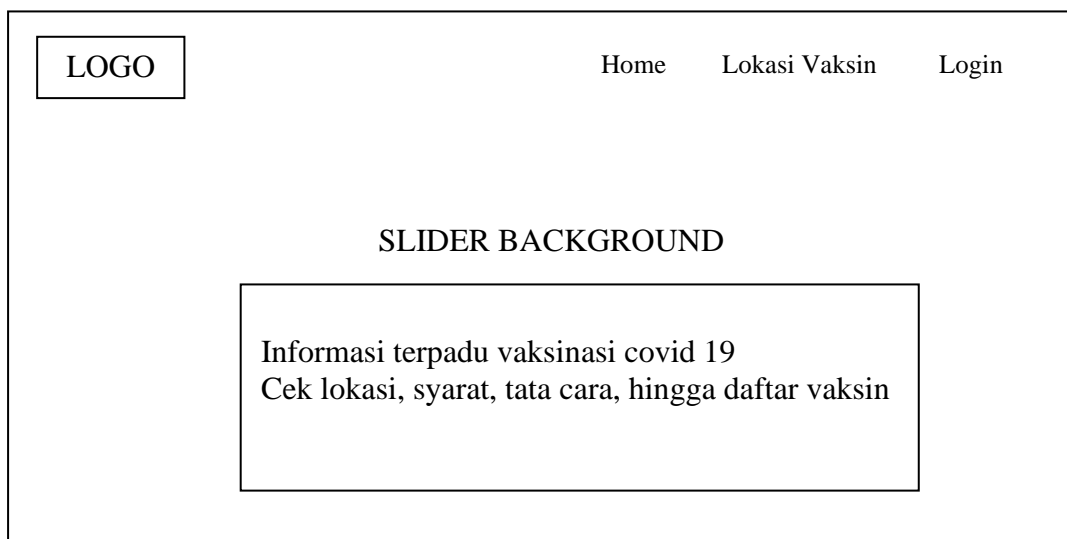
The diagram shows a login form with the following elements:

- The word "Login" is centered at the top.
- Below it, the label "Username" is followed by a rectangular input field.
- Below that, the label "Password" is followed by another rectangular input field.
- At the bottom center, there is a rectangular button labeled "Sign".

**Gambar: 3.21 Halaman login**

Rancangan antarmuka yang akan dibuat pada aplikasi ini terdapat halaman login yang berisi form berupa Username dan Password serta button untuk login.

## 2. Halaman Home



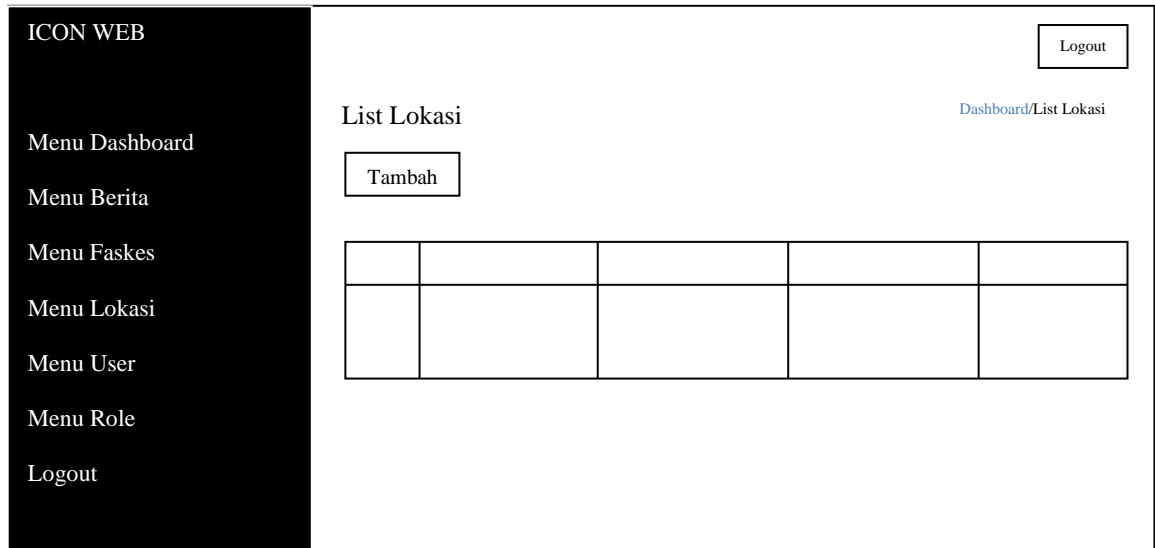
The diagram illustrates the layout of the home page with the following components:

- A "LOGO" box is located in the top left corner.
- Navigation links "Home", "Lokasi Vaksin", and "Login" are positioned in the top right area.
- The text "SLIDER BACKGROUND" is centered in the main content area.
- Below the slider text, a box contains the text: "Informasi terpadu vaksinasi covid 19" and "Cek lokasi, syarat, tata cara, hingga daftar vaksin".

**Gambar: 3.22 Halaman home**

Pada halaman ini merupakan rancangan antarmuka yang akan dibuat di Perancangan aplikasi sistem informasi geografis mengenai lokasi dan pendaftaran vaksin pada halaman utama terdapat menu utama.

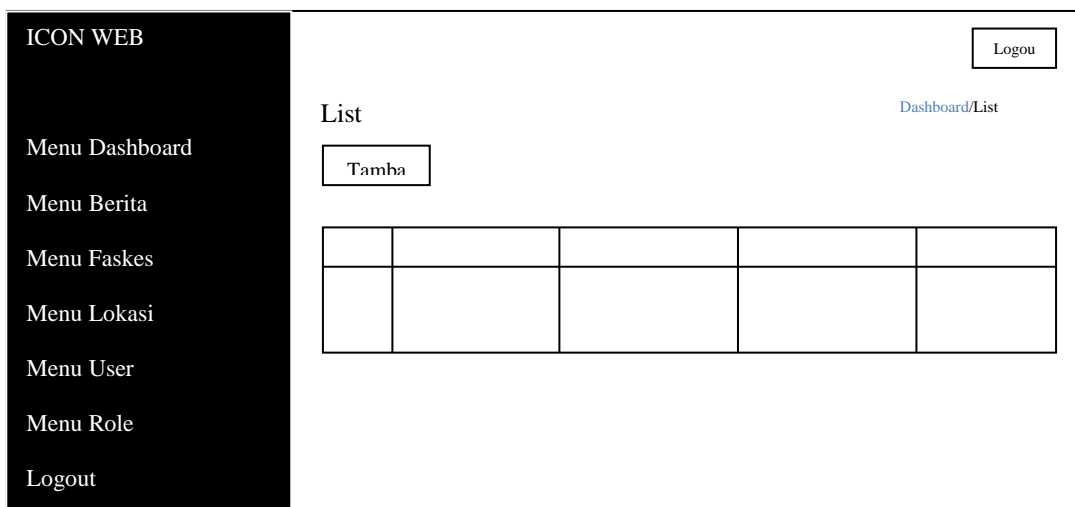
### 3. Lokasi



**Gambar: 3.23 Halaman Lokasi**

Halaman ini adalah halaman untuk menginput dan melihat lokasi Pada halaman ini terdapat sub menu tambah lokasi dan melihat lokasi.

### 4. Halaman Faskes



**Gambar: 3.24 Halaman Faskes**

Halaman ini merupakan halaman untuk menginput dan melihat data faskes. Pada halaman ini terdapat sub menu faskes yang tersedia.

### 5. Halaman Berita

ICON WEB

Menu Dashboard

Menu Berita

Menu Faskes

Menu Lokasi

Menu User

Menu Role

Logout

Logout

List Informasi [Dashboard/List info](#)

Tambah


Gambar: 3.25 Halaman berita

Halaman ini merupakan halaman untuk menginput dan melihat data berita, pada laman ini terdapat submenu berita.

### 6. Halaman Registrasi

LOGO

Home Lokasi Vaksin Login

Nama Menu

Lokasi Vaksin

NIK

Tempat Lahir

Usia

Alamat

Email

Nama

Jenis Kelamin

Telp

DAFTAR

Gambar: 3.26 Halaman registrasi

Halaman ini merupakan halaman untuk menginput data registrasi, pada laman ini terdapat submenu yang ada pada tampilan home.



## **BAB IV**

### **TESTING DAN IMPLEMENTASI SISTEM**

#### **4.1. Implementasi Sistem**

Implementasi sistem adalah prosedur yang dilakukan dalam menyelesaikan desain sistem yang telah disetujui, untuk menguji, menginstall dan memulai sistem baru atau sistem yang diperbaiki untuk menggantikan sistem yang lama. Dapat juga diartikan sebagai perbaikan sistem lama menjadi sistem yang baru.

##### **4.1.1 Tujuan Implementasi Sistem**

Adapun tujuan dari implementasi sistem ini adalah sebagai berikut :

- a. Menyelesaikan desain sistem yang telah disetujui sebelumnya.
- b. Memastikan bahwa pengguna dapat mengoperasikan sistem.
- c. Menguji apakah sistem baru tersebut sesuai dengan pemakai.

##### **4.1.2 Implementasi Perangkat Lunak**

Perangkat lunak yang digunakan pada sistem komputer yang digunakan untuk pengembangan Aplikasi Sistem Informasi Geografis Lokasi Dan Pendaftaran Vaksin Covid-19 Berbasis Web di Kota Cimahi adalah sebagai berikut:

1. Sistem operasi *windows 10*
2. *Apache web server*

3. *MySQL*
4. *Web Browser*

#### **4.1.3 Implementasi Perangkat Keras**

Kebutuhan minimum perangkat keras (*hardware*) yang diperlukan untuk mengimplementasikan program aplikasi yang dibuat adalah perangkat keras komputer PC kompatibel dengan spesifikasi yang disebutkan dibawah ini.

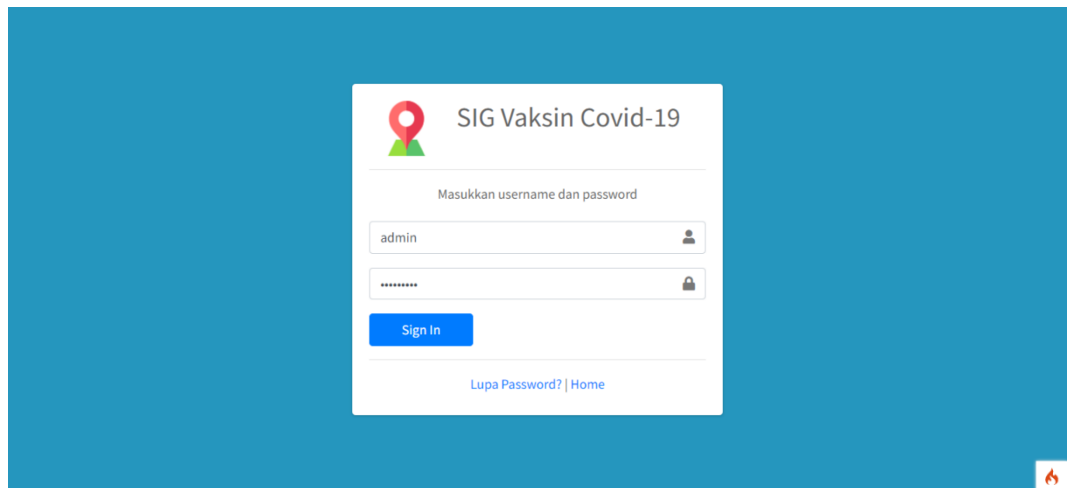
1. *Processor : Processor Intel(R) Core(TM) i3-4310 M CPU 2.70GHz*
2. *Harddisk : 500 GB*
3. *Memory : 8 GB*
4. *VGA : 2 GB*
5. *Monitor : 15''*
6. *Mouse dan Keyboard*

#### **4.1.4 Implementasi Antarmuka**

Implementasi antarmuka dilakukan dengan setiap halaman program yang dibuat dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antar muka

1. Login

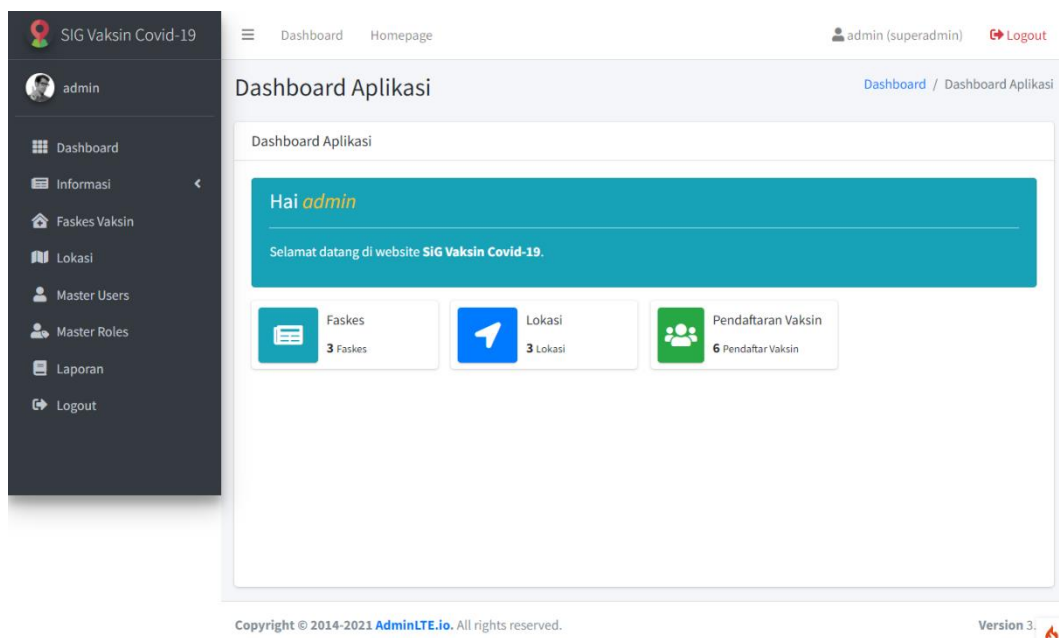
Pada halaman login, setiap user yang mempunyai akun maka dapat melakukan proses login



**Gambar 4.1 Halaman Login**

## 2. Home

Pada halaman home, pengguna dapat melihat setiap informasi yang tersedia.



**Gambar 4.2 Halaman Home**

### 3. Registrasi

Pada halaman registrasi, pengguna dapat melakukan pendaftaran untuk vaksin serta dapat langsung melakukan print out.

Sistem Informasi Geografis Vaksin Covid-19

**SIG VAKSIN COVID-19** HOME BERITA KEGIATAN INFOGRAFIS LOKASI VAKSIN Login

Form Pendaftaran Vaksin Home / Form Pendaftaran Vaksin

**Lokasi Vaksin**

Nama Faskes: Puskesmas Cigugur Tengah

Nama Lokasi: Gedung Cigugur Tengah

Alamat: Jl. Abdul halim No. 199 Kel. Cigugur, Kec. Cimahi Tengah

**Data Diri**

NIK\*: 3333333333333333 Nama Lengkap\*: Jajang

Tempat Lahir\*: Bandung Tanggal Lahir\*: 10-01-1990

Usia\*: 31 Jenis Kelamin\*: Laki-Laki

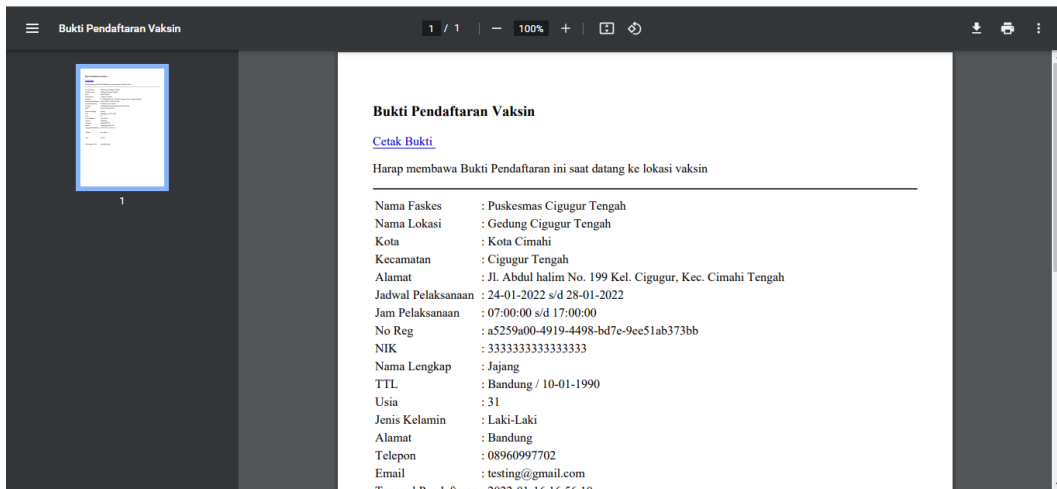
Alamat\*: Bandung Email: testing@gmail.com

Telepon \*: 08960997702

Registrasi Kembali

© Copyright Satia Nur Maulidin. All Rights Reserved  
Designed by BootstrapMade

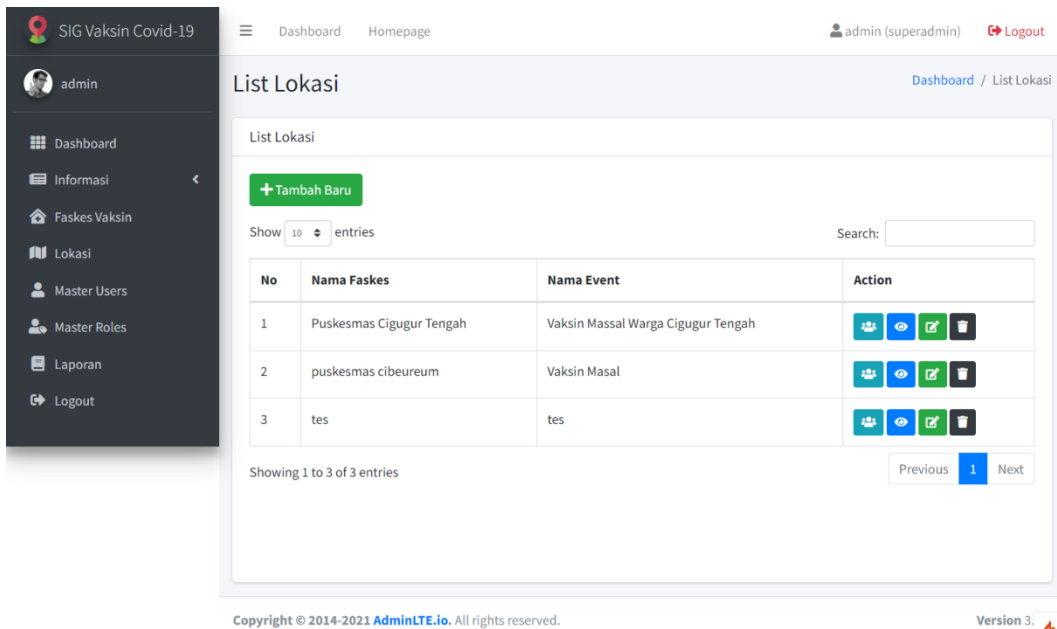
**Gambar 4.3 Halaman Registrasi**



**Gambar 4.4 Halaman Pendaftaran vaksin**

#### 4. Lokasi

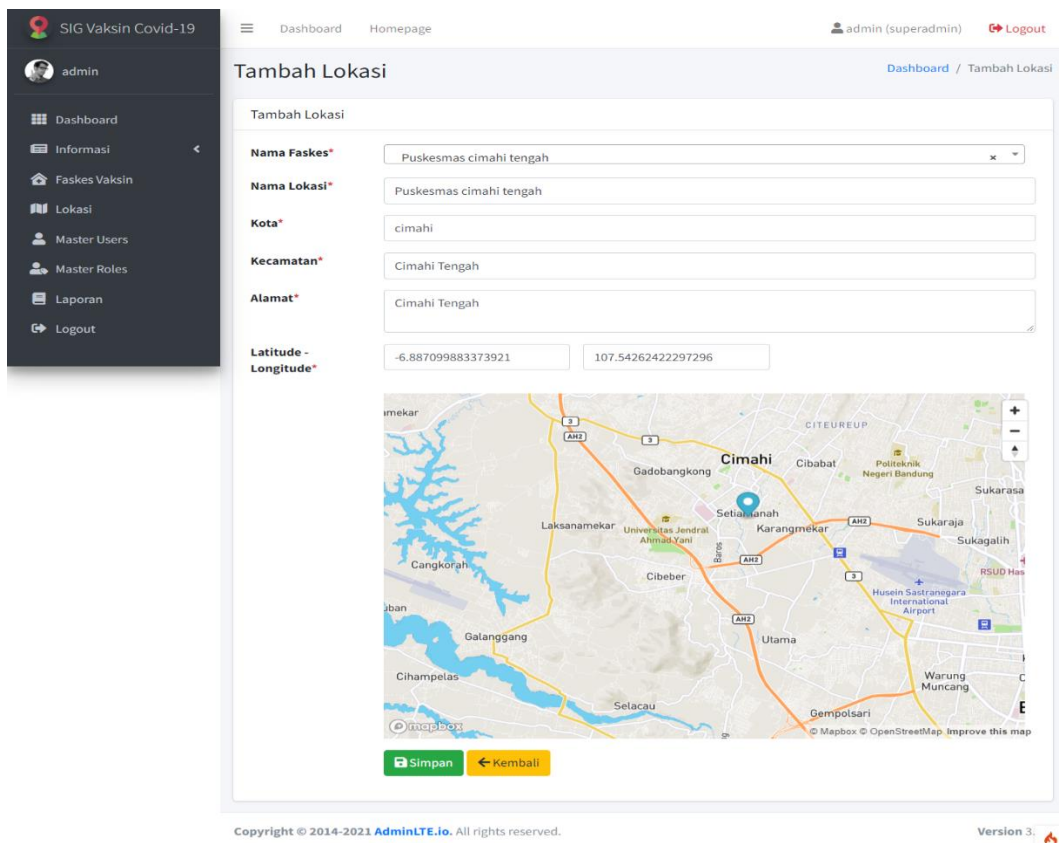
Halaman ini merupakan halaman untuk menginput dan melihat data lokasi. Pada halaman ini terdapat sub menu lokasi yang tersedia berserta maps.



**Gambar 4.5 Halaman Lokasi**



**Gambar 4.6 Halaman Lokasi Vaksin**



**Gambar 4.7 halaman Tambah Lokasi**

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.

**Gambar 4.8** halaman list registrasi vaksin

**Gambar 4.9** update kehadiran vaksin

## 5. Faskes

Halaman ini merupakan halaman untuk menginput dan melihat data Fasilitas Kesehatan (Faskes).

The screenshot displays the 'List Faskes Vaksin' page. The page header includes 'SIG Vaksin Covid-19', 'Dashboard', 'Homepage', and user information 'admin (superadmin)' with a 'Logout' button. The left sidebar contains navigation options: Dashboard, Informasi, Faskes Vaksin, Lokasi, Master Users, Master Roles, Laporan, and Logout. The main content area features a '+ Tambah Baru' button, a search bar, and a table with the following data:

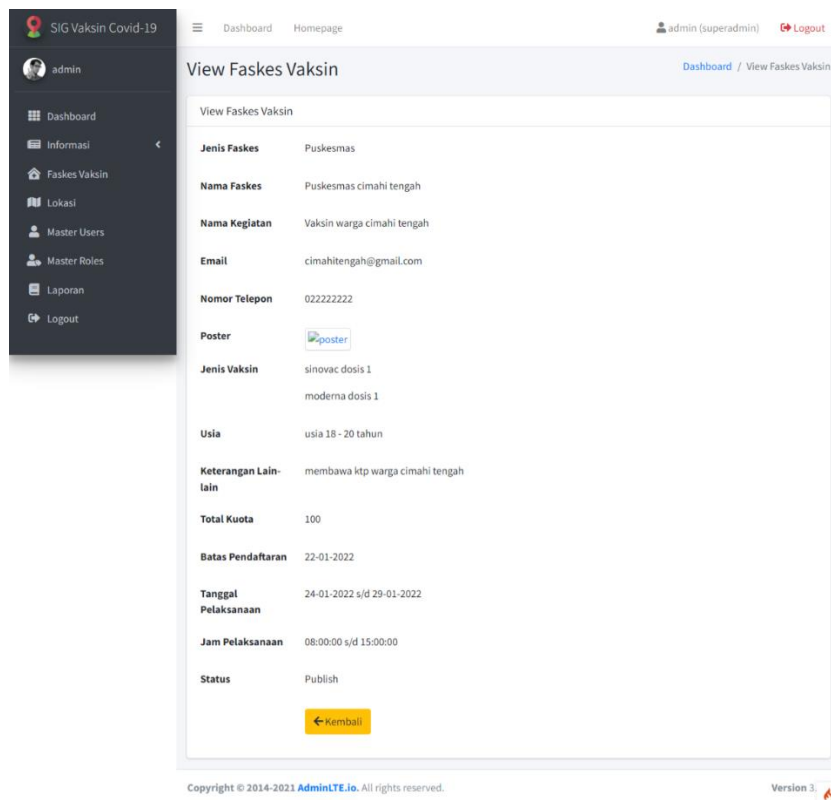
No	Nama Faskes	Tipe	Nama Kegiatan	Status - Author	Action
1	tes	Non Puskesmas	tes	admin 2022-01-15 23:11:18 ✓ Publish	[View] [Edit] [Delete]
2	puskesmas cibeureum	Puskesmas	Vaksin Masal	admin 2022-01-15 11:48:57 ✓ Publish	[View] [Edit] [Delete]
3	Puskesmas Cigugur Tengah	Puskesmas	Vaksin Massal Warga Cigugur Tengah	admin 2022-01-15 11:47:08 ✓ Publish	[View] [Edit] [Delete]

Showing 1 to 3 of 3 entries

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.1.0

**Gambar 4.10 Halaman Faskes**





**Gambar 4.11 Halaman View Faskes Vaksin**

## 6. Informasi

Pada halaman ini, pengguna dapat melihat berita yang tersedia pada laman web.

The screenshot shows the 'List Informasi' page. At the top, there is a navigation bar with 'Dashboard' and 'Homepage' links, and a user profile for 'admin (superadmin)' with a 'Logout' button. A sidebar on the left contains menu items: Dashboard, Informasi, Faskes Vaksin, Lokasi, Master Users, Master Roles, Laporan, and Logout. The main content area is titled 'List Informasi' and features a '+Tambah Baru' button. Below it is a table with the following data:

No	Gambar	Judul	Kategori - Jenis	Author - Status	Action
1	-	Tes Infografis 1 Hits: 16 Icon: Publish: 06 Januari 2022 jam 22:02:00 Updated: 06 Januari 2022 jam 23:46:21	Infografis Infografis	admin Publish	[Eye] [Edit] [Trash]
2	-	Tes Kegiatan 1 Hits: 17 Icon: Publish: 06 Januari 2022 jam 22:02:00 Updated: 06 Januari 2022 jam 23:46:16	Kegiatan Kegiatan	admin Publish	[Eye] [Edit] [Trash]
3	-	Tes Berita 3 Hits: 5 Icon: Publish: 06 Januari 2022 jam 22:00:00 Updated: 06 Januari 2022 jam 23:46:12	Berita Berita	admin Publish	[Eye] [Edit] [Trash]
4	-	Tes Berita 2 Hits: 2 Icon: Publish: 06 Januari 2022 jam 22:00:00 Updated: 06 Januari 2022 jam 23:20:14	Berita Berita	admin Publish	[Eye] [Edit] [Trash]
5		Besok, Monumen Pahlawan COVID-19 di Jawa Barat Diresmikan Hits: 7 Icon: fa-news Publish: 06 Januari 2022 jam 21:36:00 Updated: 06 Januari 2022 jam 23:44:43	Berita Berita	admin Publish	[Eye] [Edit] [Trash]

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.1.0

**Gambar 4.12 Halaman Informasi**

The screenshot shows the 'Tambah Informasi' page. At the top, there is a navigation bar with 'Dashboard' and 'Homepage' links, and a user profile for 'admin (superadmin)' with a 'Logout' button. A sidebar on the left contains menu items: Dashboard, Informasi, Faskes Vaksin, Lokasi, Master Users, Master Roles, Laporan, and Logout. The main content area is titled 'Tambah Informasi' and contains a form with the following fields:

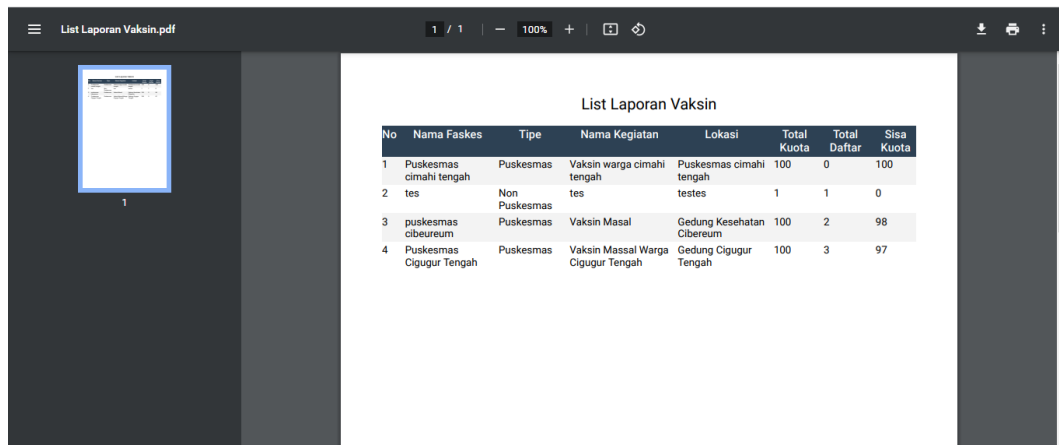
- Judul Berita:** Text input field containing 'Tes Berita 4'.
- Upload Gambar Berita:** 'Choose File' button and 'No file chosen' text.
- Kategori, Jenis & Status:** Three dropdown menus: 'Berita', 'Berita', and 'Publish'. Below them are labels for 'Kategori', 'Jenis konten', 'Status publikasi', and 'Icon Fontawesome'.
- Tanggal dan jam Publikasi:** Two date/time input fields. The first contains '16-01-2022' and the second contains '17:36:42'. Below them are format instructions: 'Format dd-mm-yyyy. Misal: 16-01-2022' and 'Format HH:MM:SS. Misal: 17:36:42'.
- Ringkasan:** Text area containing 'Tes Berita 4'.
- Isi Berita:** Rich text editor with a toolbar (File, Edit, View, Insert, Format, Tools, Table, Help) and a text area containing 'Tes Berita 4'. Below the text area is a word count: '3 WORDS POWERED BY TINY'.
- Keyword Berita (untuk SEO Google):** Text input field containing 'Tes Berita 4'.

At the bottom of the form are two buttons: 'Simpan' (Save) and 'Kembali' (Back). Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.1.0

**Gambar 4.13 Halaman Tambah Informasi**

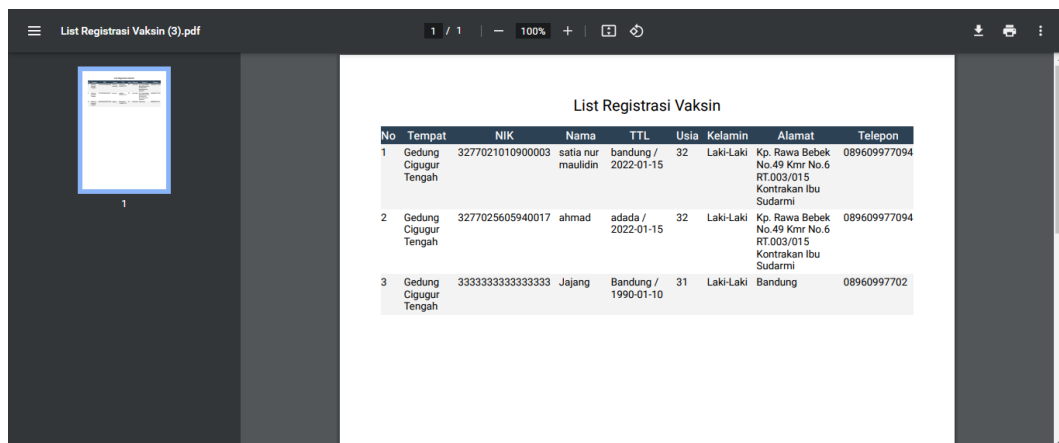
## 7. Laporan

Pada halaman ini, Dinas Kesehatan dapat melihat berbagai laporan yang ada, seperti laporan vaksin serta laporan masyarakat yang sudah registrasi.



No	Nama Faskes	Tipe	Nama Kegiatan	Lokasi	Total Kuota	Total Daftar	Sisa Kuota
1	Puskesmas cimahi tengah	Puskesmas	Vaksin warga cimahi tengah	Puskesmas cimahi tengah	100	0	100
2	tes	Non Puskesmas	tes	testes	1	1	0
3	puskesmas cibereum	Puskesmas	Vaksin Masal	Gedung Kesehatan Cibereum	100	2	98
4	Puskesmas Cigugur Tengah	Puskesmas	Vaksin Massal Warga Cigugur Tengah	Gedung Cigugur Tengah	100	3	97

**Gambar 4.14 Laporan Vaksin**



No	Tempat	NIK	Nama	TTL	Usia	Kelamin	Alamat	Telepon
1	Gedung Cigugur Tengah	3277021010900003	sattia nur maulidin	bandung / 2022-01-15	32	Laki-Laki	Kp. Rawa Bebek No.49 Km No.6 RT.003/015 Kontrakan Ibu Sudarmi	089609977094
2	Gedung Cigugur Tengah	3277025605940017	ahmad	adada / 2022-01-15	32	Laki-Laki	Kp. Rawa Bebek No.49 Km No.6 RT.003/015 Kontrakan Ibu Sudarmi	089609977094
3	Gedung Cigugur Tengah	3333333333333333	Jajang	Bandung / 1990-01-10	31	Laki-Laki	Bandung	08960997702

**Gambar 4.15 Laporan Registrasi Vaksin**

## 4.2 Pengujian Sistem

Pengujian merupakan bagian yang tidak dapat terpisahkan dari pembangunan sebuah sistem, karena dengan melakukan pengujian terhadap sistem yang akan diimplementasikan, maka dapat diketahui apakah sistem tersebut sudah sesuai dengan kebutuhan. Dan dimaksudkan agar kualitas dari sistem dapat terjamin

sebelum diimplementasikan. Dari pengujian sistem, kita dapat mengetahui kesalahan – kesalahan yang terdapat pada sistem seperti kesalahan penulisan sintaks yang menyebabkan sistem tidak berjalan sesuai dengan keinginan dari perancangan sistem tersebut.

#### 4.2.1 Rencana Pengujian

Pengujian merupakan bagian yang penting dalam pembangunan sebuah perangkat lunak. Pengujian ditujukan untuk menemukan kesalahan-kesalahan pada sistem dan memastikan sistem yang dibangun telah sesuai dengan apa yang direncanakan sebelumnya. Adapun metode yang digunakan dalam melakukan pengujian adalah pengujian *black box*. Metode pengujian *black box* adalah pengujian terhadap validasi masukan yang akan diolah oleh suatu unit dan memfokuskan pada fungsionalitas dari perangkat lunak. Pengujian *black box* yang dilakukan pada Aplikasi Sistem Informasi Geografis Lokasi Dan Pendaftaran Vaksin Covid-19 Berbasis Web di Kota Cimahi dapat dilihat pada tabel dibawah ini :

**Tabel 4.1 Rencana pengujian**

Kode kasus	Kelas uji	Butir uji	Jenis pengujian
1.	Pengujian Login	Validasi <i>form</i>	<i>Black box</i>
2.	Pengujian Registrasi	Validasi <i>form</i>	<i>Black box</i>
3.	Pengujian Lokasi	Validasi <i>form</i>	<i>Black box</i>
4.	Pengujian Faskes	Validasi <i>form</i>	<i>Black box</i>
5.	Pengujian Berita	Validasi <i>form</i>	<i>Black box</i>

Setelah dilakukan pengujian dengan metode *blackbox*, maka hasil pengujian yang didapatkan adalah:

**Tabel 4.2 Hasil pengujian**

Kode Kasus	Data Masukan	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	<i>Field</i> username tidak diisi data	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	<i>Field</i> password tidak diisi	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
2.	Nama pada registrasi tidak diisi	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	NIK tidak diisi	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Email tidak diisi	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Klik tombol simpan	<i>Updating database</i>	<i>Database ter-update</i>	<i>Valid</i>
3.	Tidak memasukkan nama lokasi	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Tidak memasukkan kota	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Tidak memasukkan kecamatan	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Tidak memasukkan alamat	Ditampilkan pesan kesalahan	Sistem menampilkan pesan	<i>Valid</i>

Kode Kasus	Data Masukan	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
			kesalahan	
	Tidak memasukkan latitude & longitude	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Klik tombol simpan	Basis data ter-update	Basis data ter-update	<i>Valid</i>
4.	Tidak memasukkan jenis faskes	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Tidak memasukkan nama faskes	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Tidak memasukkan nama kegiatan	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Tidak memasukkan email	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Nomor telpon diisi huruf	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Klik tombol simpan	Basis data ter-update	Basis data ter-update	<i>Valid</i>
5.	Tidak memasukkan judul berita	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Tanggal dan jam menggunakan huruf	Ditampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	<i>Valid</i>
	Klik tombol simpan	Basis data ter-update	Basis data ter-update	<i>Valid</i>

Berdasarkan hasil testing yang sudah dilakukan, maka dapat ditarik kesimpulan yaitu aplikasi sistem informasi geografis lokasi dan pendaftaran vaksin di Kota Cimahi sudah berjalan sesuai dengan harapan, tetapi tidak menutup kemungkinan terjadi kesalahan pada sistem yang akan digunakan.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Dari hasil pembuatan Tugas Akhir dalam perancangan pembuatan website ini dapat diambil kesimpulan sebagai berikut :

1. Aplikasi sistem informasi geografis lokasi dan pendaftaran vaksin ini dapat membantu masyarakat agar lebih mudah memperoleh informasi lokasi penyelenggaraan vaksin yang ada di Kota Cimahi.
2. Aplikasi sistem informasi geografis lokasi dan pendaftaran vaksin ini dapat memudahkan masyarakat memperoleh informasi mengenai kuota yang tersedia di fasilitas kesehatan yang menyelenggarakan serta masyarakat dapat dengan mudah untuk mendaftar sebagai calon penerima vaksin.



## 5.2 Saran

Aplikasi ini masih jauh dari kesempurnaan karena keterbatasan penulis dalam penguasaan ilmu PHP dan Mysql, sehingga penulis berharap :

1. Meningkatkan pengguna teknologi komputer yang lebih terkini .
2. Diharapkan desain website yang sudah ada ini dapat dikembangkan lagi, sehingga desain nantinya dapat di browser pada tools apapun dengan tampilan yang menarik sesuai desain.
3. Diharapkan kedepannya dibuatkan untuk versi mobilenya sehingga masyarakat tidak usah membuka aplikasinya lewat laptop cukup lewat handphone saja
4. Diharapkan kedepannya terintegrasi dengan service dukcapil atau service pemerintahannya lainnya untuk pengecekan data nik dan status vaksin.
5. Penerapan sistem baru akan berjalan dengan baik dan sukses jika semua pihak yang terlibat mendukung penerapan sistem yang baru.