

**SISTEM INFORMASI GEOGRAFIS PEMETAAN DAERAH  
RAWAN KRIMINALITAS DI KOTA BANDUNG**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan  
Jenjang Strata Satu(S1)  
Pada program Studi Teknik Informatika**

Oleh :

**IBRAHIM LESMANA SALAM**

**361662005**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER  
INDONESIA MANDIRI  
BANDUNG  
2020**

**LEMBAR PENGESAHAN**

**SISTEM INFORMASI GEOGRAFIS PEMETAAN DAERAH  
RAWAN KRIMINALITAS DI KOTA BANDUNG**

***GEOGRAPHICAL INFORMATION SYSTEM OF  
CRIMINALITY MAPPING IN BANDUNG CITY***

Oleh:

**IBRAHIM LESMANA SALAM**  
**361662005**

Skripsi ini telah diterima dan disahkan  
untuk memenuhi persyaratan mencapai gelar

SARJANA TEKNIK INFORMATIKA

Pada

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

Bandung, 7 Februari 2020

Disahkan oleh

Ketua Program Studi,

Dosen Pembimbing,

**Chalifa Chazar, S.T., M.T.**

NIDN: 0421098704

**Dr. Chairuddin, M.T., M.M**

NIDN: 0426076901

## LEMBAR PERSETUJUAN REVISI TUGAS AKHIR

### SISTEM INFORMASI GEOGRAFIS PEMETAAN DAERAH RAWAN KRIMINALITAS DI KOTA BANDUNG

Oleh  
Ibrahim Lesmana Salam  
361662005

Telah melakukan sidang tugas akhir pada Hari Rabu, 05 Februari 2020 dan telah melakukan revisi sesuai dengan perubahan yang diminta pada saat sidang tugas akhir

Bandung, 7 februari 2020

Menyetujui,

No	Nama	Keterangan	Tanda Tangan
1	Dr.Chairuddin,Ir.,M.M.,M.T.	Pembimbing	
2	Chalifa Chazar, S.T., M.T.	Penguji 1	
3	Moch.Ali Ramdhani,S.T.,M.Kom.	Penguji 2	

Mengetahui

Ketua Program Studi Teknik Informatika

**Chalifa Chazar, S.T., M.T.**

NIDN: **0421098704**

## **SURAT PERNYATAAN**

Dengan ini saya menyatakan bahwa:

- (1) Naskah Skripsi ini adalah asli dan belum pernah diajukan untuk mendapat gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- (2) Skripsi ini murni karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka saya telah sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- (3) seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya seni ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan 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 Indonesia Mandiri.

Bandung, 7 Februari 2020

Yang Membuat Pernyataan

**IBRAHIM LESMANA SALAM**  
**361662005**

# **SISTEM INFORMASI GEOGRAFIS PEMETAAN DAERAH RAWAN KRIMINALITAS DI KOTA BANDUNG**

## **ABSTRAK**

Kriminalitas memang merupakan masalah yang umum ada di manapun. Tindak kejahatan banyak terjadi di berbagai tempat dengan waktu kejadian yang berbeda, menyebabkan sulitnya menentukan daerah mana yang memiliki tingkat kerawanan tindak kejahatan. Informasi tentang banyaknya tindak kejahatan sangat dibutuhkan oleh masyarakat dan penegak hukum dalam hal ini jajaran kepolisian. Bagi semua pihak seperti masyarakat luas, informasi ini sangat berguna untuk tindakan antisipasi, khususnya bagi kepolisian membantu dalam mengambil keputusan apakah suatu daerah memerlukan pengawasan ekstra atau tidak, selain itu informasi tersebut dibutuhkan untuk mengetahui intensitas tindak kejahatan. Tindak kriminalitas bukan hanya tanggung jawab pihak kepolisian tetapi tanggung jawab semua lapisan masyarakat, sehingga dengan dipetakannya daerah rawan kriminalitas akan dapat diketahui dimana saja terjadi tindak kejahatan itu. Kemudian pemetaan daerah rawan kriminalitas dengan penyajian secara tersistem akan mempermudah dalam memantau menghindari dan mencegah tindak kejahatan, lalu juga bisa digunakan untuk memprediksi tindak kejahatan ditinjau dari lokasi dan waktu terjadinya kejahatan biasa terjadi. Maka dari itu dibutuhkan Sistem Informasi Geografis (SIG) sebagai salah satu teknologi yang baru berkembang, karena akan lebih dari cukup akurat untuk membantu mengurangi terjadinya kriminalitas di kota Bandung ini

Kata kunci: Sistem Informasi Geografis, Kriminalitas, Masyarakat, Bandung.

# ***GEOGRAPHICAL INFORMATION SYSTEM OF CRIMINALITY MAPPING IN BANDUNG CITY***

## ***ABSTRACT***

*Crime is indeed a common problem everywhere. Crimes often occur in various places with different time events, making it difficult to determine which areas have a degree of vulnerability to crime. Information about the number of crimes is needed by the community and law enforcement in this case the police. For all parties such as the wider community, this information is very useful for anticipatory actions, especially for the police to assist in making decisions whether an area needs extra supervision or not, besides that information is needed to determine the intensity of crime. Crime is not only the responsibility of the police, but the responsibility of all levels of society, so that the mapping of criminality-prone areas will be known wherever the crime occurred. Then mapping crime-prone areas with a systematic presentation will make it easier to monitor to avoid and prevent crime, and then it can also be used to predict crime in terms of location and time of occurrence of ordinary crime. Therefore, a Geographical Information System (GIS) is needed as one of the newly developing technologies, because it will be more than accurate enough to help reduce crime in Bandung.*

*Keywords: Geographic Information Systems, Crime, Society, Bandung.*

## **KATA PENGANTAR**

**Bismillahirrahmaanirrohim,**

Alhamdulillah segala Puji dan syukur penulis panjatkan kepada Allah SWT, karena atas Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan menyelesaikan Laporan Skripsi yang berjudul “SISTEM INFORMASI GEOGRAFIS PEMETAAN DAERAH RAWAN KRIMINALITAS DI KOTA BANDUNG”. Laporan Skripsi ini ditujukan untuk memenuhi dan melengkapi persyaratan Akademik pada jurusan Teknik Informatika di STMIK Indonesia Mandiri, Bandung.

Dalam penyusunan Laporan Skripsi ini penulis banyak mendapat saran, dorongan, bimbingan serta keterangan-keterangan dari berbagai pihak, yang membuat penulis semakin semangat untuk menyelesaikan Laporan Skripsi. Harapan penulis semoga Laporan ini akan menjadi bahan bacaan yang dapat memberikan wawasan kepada pembacanya.

Bandung, 7 Februari 2020

Penulis,

IBRAHIM LESMANA SALAM  
361662005

## UCAPAN TERIMA KASIH

**Assalamu'alaikum Wr.Wb.**

Penulis menyadari bahwa segala usaha yang dilakukan dalam penulisan ini tidak akan berhasil dengan baik tanpa bantuan dan saran dari berbagai pihak. Maka dalam kesempatan ini perkenankanlah penulis mengucapkan terimakasih kepada yang terhormat:

1. Tuhan Yang Maha Kuasa, karena oleh berkat kasih dan rahmat karuniaNya penulis bisa ada sebagaimana penulis dapat menyelesaikan tugas akhir skripsi.
2. Bapak Dr. Chairuddin, M. T.M. selaku Dosen pembimbing skripsi ini. Seorang figur yang sangat dihormati yang senantiasa selalu bersedia memberikan waktu luang dan membantu dalam membimbing sehingga penulis bisa menyelesaikan studi.
3. Ibu Chalifa Chazar, S.T., M.T., selaku Ketua Jurusan Teknik Informatika STMIK Indonesia Mandiri Bandung yang sangat penulis hormati yang senantiasa membantu sehingga penulis bisa menyelesaikan studi.
4. Kedua Orang Tua yang sangat saya cintai yang telah membantu memberikan semangat, doa, kasih harapan dan dorongan moral dan spiritual kepada Penulis dalam menyelesaikan tugas akhir skripsi.



5. Seluruh anggota dosen pengajar yang tidak dapat disebutkan satu per satu, yang telah menginspirasi dan memotivasi penulis dari awal, pertengahan hingga akhir penyelesaian tugas akhir.
6. Rekan mahasiswa Teknik Informatika angkatan 2016 dan juga semua kakak tingkat, adik tingkat serta tingkat jurusan dan universitas yang telah membantu Penulis dalam menyelesaikan tugas akhir skripsi ini.
7. Semua pihak yang pernah hadir dalam hidup penulis, yang tidak dapat disebutkan satu persatu. Terima Kasih!

Semoga Allah Swt Senantiasa Memberikan Imbalan Yang Berlipat Ganda.  
Semoga Laporan Skripsi Ini Dapat Bermanfaat Bagi Semua Pihak Yang  
Memerlukan. Amin.

Bandung, 7 Februari 2020

Penulis,

**IBRAHIM LESAMA SALAM**  
**361662005**

## DAFTAR ISI

LEMBAR PENGESAHAN

LEMBAR PERSETUJUAN REVISI

SURAT PERNYATAAN .....	i
ABSTRAK .....	ii
<i>ABSTRACT</i> .....	iii
KATA PENGANTAR .....	iv
UCAPAN TERIMA KASIH .....	v
DAFTAR ISI .....	vii
DAFTAR LAMPIRAN .....	xi
DAFTAR GAMBAR .....	xii
DAFTAR TABEL .....	xv

### BAB I PENDAHULUAN

1.1 Latar Belakang Penulisan.....	1
1.2 Identifikasi Masalah .....	3
1.3 Tujuan Penulisan.....	3
1.4 Batasan Masalah .....	3
1.5 Metode Penelitian .....	4
1.5.1 Metode Pengumpulan Data .....	4

1.5.2 Metode Pengembangan Sistem .....	6
1.6 Sistematika Penulisan .....	6

## **BAB II LANDASAN TEORI**

2.1 Kota Bandung .....	8
2.2 Kriminalitas .....	9
2.2.1 Ciri-ciri Kriminalitas .....	10
2.3 Pengertian Sistem .....	11
2.3.1 Jenis-jenis Sistem .....	12
2.3.2 Elemen Sistem .....	12
2.4 Pengertian Informasi .....	13
2.5 Pengertian Sistem Informasi Geografis .....	15
2.5.1 Jenis Data Sistem Informasi Geografis .....	16
2.6 Google Maps Api .....	17
2.7 Definisi Pemetaan .....	18
2.8 MVC ( <i>Model-View-Controller</i> ) .....	18
2.9 <i>Database</i> .....	19
2.9.1 Manfaat Penggunaan <i>Database</i> .....	20
2.10 Pengertian Diagram <i>Flowchart</i> .....	22
2.11 ERD ( <i>Entity Relationship Diagram</i> ) .....	23
2.12 <i>Unified Modelling Language</i> (UML) .....	23
2.12.1 Diagram <i>Use Case</i> .....	25
2.12.2 Diagram <i>Activity</i> .....	25
2.12.3 Diagram <i>Sequence</i> .....	25

2.12.4	Diagram <i>Class</i> .....	26
2.13	RAD ( <i>Rapid Application Development</i> ) .....	26
2.13.1	Tahapan RAD.....	26

### **BAB III REQUIREMENT PLANNING & DESIGN WORKSHOP**

3.1	<i>Requirement Planning</i> (Perencanaan Persyaratan) .....	28
3.1.1	Analisa Sistem Berjalan .....	28
3.1.2	Analisa Kelebihan Sistem .....	30
3.1.3	Analisa Kekurangan Sistem .....	31
3.1.4	Analisa Kebutuhan Pendukung Sistem .....	31
3.2	Design Workshop .....	32
3.2.1	<i>Unified Modelling Language</i> (UML).....	33
3.2.1.1	<i>Use Case Diagram</i> .....	34
3.2.1.2	<i>Activity Diagram</i> .....	37
3.2.1.3	<i>Sequence Diagram</i> .....	50
3.2.1.4	<i>Class Diagram</i> .....	55
3.2.2	Perancangan <i>Database</i> Sistem .....	57
3.2.2.1	Perancangan <i>Database</i> Dengan ERD .....	57
3.2.2.2	Perancangan Struktur <i>Database</i> .....	59
3.2.3	Perancangan <i>User Interface</i> Sistem .....	60

### **BAB IV IMPLEMENTATION**

4.1	Implementasi .....	66
4.1.1	Implementasi Pendukung Sistem .....	66

4.1.2	Implementasi <i>Database</i> .....	67
4.1.3	Implementasi <i>User Interface</i> .....	68
4.1.4	Testing Sistem .....	75

## **BAB V KESIMPULAN DAN SARAN**

5.1	Kesimpulan .....	80
5.2	Saran.....	80

## **DAFTAR PUSTAKA .....**

## DAFTAR LAMPIRAN

### LAMPIRAN A

1. Index.....	A1
2. Controller .....	A13
3. Model .....	A14
4. Maps Js .....	A15

## DAFTAR GAMBAR

Gambar 3.1 Flowchart Admin .....	29
Gambar 3.2 Flowchart User .....	30
Gambar 3.3 Use Case Diagram .....	34
Gambar 3.4 Activity Diagram Tambah Data Wilayah.....	37
Gambar 3.5 Activity Diagram Ubah Data Wilayah.....	38
Gambar 3.6 Activity Diagram Hapus Data Wilayah .....	39
Gambar 3.7 Activity Diagram Lihat Data Wilayah .....	40
Gambar 3.8 Activity Diagram Tambah Admin.....	41
Gambar 3.9 Activity Diagram Ubah Admin .....	42
Gambar 3.10 Activity Diagram Hapus Admin .....	43
Gambar 3.11 Activity Diagram Lihat Admin .....	44
Gambar 3.12 Activity Diagram Tambah Data Spasial.....	45
Gambar 3.13 Activity Diagram Ubah Data Spasial .....	46
Gambar 3.14 Activity Diagram Hapus Data Spasial .....	47
Gambar 3.15 Activity Diagram Lihat Data Spasial .....	48
Gambar 3.16 Activity Diagram User Lihat Daftar Wilayah .....	49
Gambar 3.17 Activity Diagram Login .....	49
Gambar 3.18 Sequence Diagram Tambah Daftar Wilayah.....	50
Gambar 3.19 Sequence Diagram Ubah Daftar Wilayah .....	50
Gambar 3.20 Sequence Diagram Hapus Daftar Wilayah.....	51

Gambar 3.21 Sequence Diagram Lihat Daftar Wilayah .....	51
Gambar 3.22 Sequence Diagram Tambah Admin .....	51
Gambar 3.23 Sequence Diagram Ubah Admin.....	52
Gambar 3.24 Sequence Diagram Hapus Admin .....	52
Gambar 3.25 Sequence Diagram Lihat Admin .....	52
Gambar 3.26 Sequence Diagram Tambah Data Spasial .....	53
Gambar 3.27 Sequence Diagram Ubah Data Spasial.....	53
Gambar 3.28 Sequence Diagram Hapus Data Spasial .....	53
Gambar 3.29 Sequence Diagram Lihat Data Spasial .....	54
Gambar 3.30 Sequence Diagram User Lihat Daftar Wilayah.....	54
Gambar 3.31 Sequence Diagram Login .....	55
Gambar 3.32 Perancangan Class Diagram.....	56
Gambar 3.33 Perancangan Database Menggunakan ERD.....	58
Gambar 3.34 Perancangan Interface Index .....	60
Gambar 3.35 Perancangan Perancangan Interface Login .....	61
Gambar 3.36 Perancangan Interface Dashboard.....	62
Gambar 3.37 Perancangan Interface Datapage .....	62
Gambar 3.38 Perancangan Interface Tambah Data.....	63
Gambar 3.39 Perancangan Interface Ubah Data .....	64
Gambar 3.40 Perancangan Interface Hapus Data .....	64
Gambar 3.41 Perancangan Interface Lihat Data .....	65
Gambar 4.1 Implementasi Database Data Admin.....	67
Gambar 4.2 Implementasi Database Data Spasial Wilayah.....	68
Gambar 4.3 Implementasi Database Data Wilayah .....	68
Gambar 4.4 Implementasi Interface Index.....	69



Gambar 4.5 Perancangan Implementasi Interface Login.....	69
Gambar 4.6 Implementasi Interface Dashboard.....	70
Gambar 4.7 Implementasi Interface Data Admin .....	70
Gambar 4.8 Implementasi Interface Tambah Data Admin .....	71
Gambar 4.9 Implementasi Interface Ubah Data Admin .....	71
Gambar 4.10 Implementasi Interface Data Wilayah.....	72
Gambar 4.11 Implementasi Interface Tambah Data Wilayah.....	72
Gambar 4.12 Implementasi Interface Ubah Data Wilayah .....	73
Gambar 4.13 Implementasi Interface Data Spasial Wilayah .....	73
Gambar 4.14 Implementasi Interface Tambah Data Spasial Wilayah .....	74
Gambar 4.15 Implementasi Interface Ubah Data Spasial Wilayah.....	74

## **DAFTAR TABEL**

Tabel 3.1 Penjelasan Aktor Use case .....	33
Tabel 3.2 Skenario Use Case Diagram .....	34
Tabel 3.3 Data Admin .....	59
Tabel 3.4 Data Wilayah.....	59
Tabel 3.5 Data Spasial Wilayah .....	60
Tabel 4.1 Pengujian Menu Data Admin.....	75
Tabel 4.2 Pengujian Menu Data Spasial Wilayah.....	76
Tabel 4.3 Pengujian Menu Daftar Wilayah.....	77
Tabel 4.4 Pengujian User .....	77
Tabel 4.5 Pengujian Login dan Logout Admin.....	78
Tabel 4.6 Pengujian Role Page Admin dan User .....	78

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang Penelitian**

Kota Bandung merupakan kota metropolitan terbesar di Provinsi Jawa Barat, sekaligus menjadi ibu kota provinsi tersebut. Kota ini terletak 140 km sebelah tenggara Jakarta, dan merupakan kota terbesar ketiga di Indonesia setelah Jakarta dan Surabaya menurut jumlah penduduk. Selain itu, Kota Bandung juga merupakan kota terbesar di wilayah Pulau Jawa bagian selatan. Sedangkan wilayah Bandung Raya (Wilayah Metropolitan Bandung) merupakan metropolitan terbesar ketiga di Indonesia setelah Jabodetabek dan Gerbang kertosusila. Dikarenakan kota Bandung sebagai metropolitan, tidak sedikit masyarakat yang memilih untuk melakukan tindakan kriminal demi mendapatkan uang lebih agar kehidupan ekonominya lebih sejahtera/layak. Kemungkinan faktor - faktor yang mempengaruhi timbulnya masalah-masalah terjadi di perkotaan sangat beragam contohnya peningkatan populasi masyarakat yang meledak, urbanisasi, kebutuhan ekonomi, finansial yang kurang.

Kriminalitas memang merupakan masalah yang umum ada di manapun. Tindak kejahatan banyak terjadi di berbagai tempat dengan waktu kejadian yang berbeda, menyebabkan sulitnya menentukan daerah mana yang memiliki tingkat kerawanan tindak kejahatan. Informasi tentang banyaknya tindak kejahatan sangat dibutuhkan oleh masyarakat dan penegak hukum dalam hal ini jajaran kepolisian. Bagi semua pihak seperti masyarakat luas, informasi ini sangat berguna untuk

tindakan antisipasi, khususnya bagi kepolisian membantu dalam mengambil keputusan apakah suatu daerah memerlukan pengawasan ekstra atau tidak, selain itu informasi tersebut dibutuhkan untuk mengetahui intensitas tindak kejahatan.

Tindak kriminalitas bukan hanya tanggung jawab pihak kepolisian tetapi tanggung jawab semua lapisan masyarakat, sehingga dengan dipetakannya daerah rawan kriminalitas akan dapat diketahui dimana saja terjadi tindak kejahatan itu. Kemudian pemetaan daerah rawan kriminalitas dengan penyajian secara tersistem akan mempermudah kita dalam memantau menghindari dan mencegah tindak kejahatan, lalu juga bisa digunakan untuk memprediksi tindak kejahatan ditinjau dari lokasi dan waktu terjadinya kejahatan biasa terjadi.

Dalam wilayah hukum Polrestabes Bandung terdapat sebanyak 3.027 kasus kriminalitas pada sepanjang tahun 2018, dengan jam operasional beberapa aksi diperkirakan terjadi antara pukul 03.00 sampai dengan 06.00.

Oleh karena itu, Sistem Informasi Geografis (SIG) sebagai salah satu teknologi yang baru berkembang, dirasakan cukup akurat untuk membantu mengurangi terjadinya kriminalitas di perkotaan terutama kota besar seperti kota Bandung ini.

Ditinjau dari latar belakang diatas penulis tertarik untuk melakukan penelitian dengan judul “SISTEM INFORMASI GEOGRAFIS PEMETAAN DAERAH RAWAN KRIMINALITAS DI KOTA BANDUNG”.

## **1.2 Identifikasi Masalah**

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

1. Bagaimana persebaran daerah rawan kriminalitas di kota Bandung?
2. Bagaimana menerapkan metode sistem informasi geografis dalam melakukan pemetaan daerah rawan kriminalitas di kota Bandung?

### **1.3 Tujuan Penelitian**

Dalam melakukan penelitian ini penulis memiliki tujuan sebagai berikut:

1. Memetakan daerah-daerah mana saja yang sering terjadi tindak kriminalitas.
2. Membuat sistem yang dapat melakukan pemetaan daerah dalam bentuk radius pada setiap daerah yang sering terjadi kriminalitas.

### **1.4 Batasan Masalah Penelitian**

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

1. Peta dibuat kedalam bentuk peta digital berbentuk website
2. Penulis akan menuliskan keterangan kriminalitas pada suatu daerah seperti jam operasional dan juga kejahatan apa saja yang terjadi di daerah tersebut.
3. Peta diambil menggunakan GoogleMaps API
4. Penentuan daerah rawan kriminalitas akan digambarkan dalam bentuk radius pada peta.
5. Data didapat dari jajarannya Polrestabes kota Bandung melalui media internet.

## **1.5 Metode Penelitian**

Metode penelitian disini akan menjelaskan metode-metode yang digunakan penulis dalam melakukan penelitian ini. Berikut adalah metode penelitian yang digunakan penulis. Penelitian ini menggunakan metode deskriptif. Metode ini memaparkan permasalahan yang ada, menggambarkan keadaan yang ada serta mengumpulkan data sesuai dengan yang sebenarnya dan mencoba memecahkan masalah dengan menggunakan program aplikasi PHP Codeigniter MySql serta menggunakan teknologi google maps api yang diharapkan dapat membantu proses pengolahan data, sehingga akurasi yang diharapkan dapat dicapai

### **1.5.1 Metode Pengumpulan Data**

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Sebelum melakukan penelitian, seorang peneliti biasanya telah memiliki dugaan berdasarkan teori yang digunakan, dugaan tersebut disebut dengan hipotesis.

- Data Primer

Data primer adalah data yang diperoleh dari sumbernya langsung.

- Data Sekunder

Data primer adalah data yang diperoleh tidak dari sumbernya langsung.

- 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.

- Wawancara

Wawancara adalah teknik pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab langsung antara peneliti dan narasumber. Seiring perkembangan teknologi, metode wawancara dapat pula dilakukan melalui media-media tertentu, misalnya telepon, *email*, atau *skype*.

- 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 Pengembangan Sistem penulis merancang sistem informasi evaluasi kinerja dosen dengan menggunakan metode pengembangan sistem Rapid Application Development (RAD) (Kendall, 2006), yaitu dengan tahap Perencanaan Syarat-syarat, Workshop Desain, dan Implementasi serta notasi yang digunakan adalah Unified Modeling Language (UML) dan Entity Relationship Diagram (ERD).

## **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 akan diuraikan mengenai dasar-dasar teori pendukung analisis sistem yaitu konsep dasar sistem, informasi, sistem informasi, metode pengumpulan data, metode pengembangan sistem, sistem informasi geografis, pengertian pemetaan, pengertian data.

### **BAB III : REQUIREMENT PLANING & DESIGN WORKSHOP**

Pada bab ini akan diuraikan mengenai pengembangan sistem yang akan digunakan dalam proses perancangan sistem informasi geografis pemetaan daerah rawan kriminalitas di kota Bandung dan uraian mengenai analisis sesuai kebutuhan sistem, lalu disamping itu juga akan membahas perancangan yang akan dibuat sesuai dengan analisa permasalahan.



## BAB VI : IMPLEMENTASI DAN UJI COBA

Pada bab ini dibahas tentang implementasi sistem informasi geografis pemetaan daerah rawan kriminalitas yang telah dirancang dan pada tahap ini akan dilakukan pengujian untuk mengetahui apakah hasil yang didapatkan sesuai dengan yang diharapkan

## 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 Kota Bandung**

Bandung adalah kota metropolitan yang secara geografis bagian dari wilayah Provinsi Jawa Barat. Bandung memiliki banyak nama. Satu di antaranya yang legendaris adalah Kota Kembang.

Secara harfiah, kata 'kembang' artinya bunga dalam Bahasa Indonesia. Sekarang sudah menjadi sinonim karena kembang sudah diserap ke dalam Bahasa Indonesia. Padu padan 'Bandung Kota Kembang' bermakna Bandung Kota Bunga. Dalam gambaran kata per kata, Kota Bandung pada saat itu memiliki keindahan yang sangat luar biasa dengan hiasan bunga yang tumbuh subur.

Secara geografis, kota Bandung terletak di tengah-tengah provinsi Jawa Barat, serta berada pada ketinggian kurang lebih 768 meter di atas permukaan laut. Kota Bandung memiliki luas wilayah 16.731 hektar, yang secara administratif terbagi atas 30 kecamatan, 151 kelurahan, 1.561 rw, dan 9.691 rt. kecamatan terluas adalah kecamatan gedebage, dengan luas 958 hektar dan kecamatan terkecil adalah wilayah kecamatan astana anyar dengan luas 89 hektar. sedangkan jumlah penduduk kota Bandung tahun 2012 tercatat 2.655.160 jiwa, terdiri dari 1.358.623 laki-laki, dan 1.296.537 perempuan.

Dari aspek pemerintahan, kota Bandung dipimpin oleh walikota dan wakil walikota. serta dibantu sekretaris daerah, yang membawahi 3 asisten sekretaris daerah, 17 kepala dinas, 6 kepala badan, 8 kepala bagian, 1 kepala kantor, 4 perusahaan daerah, 1 inspektorat, 1 kepala satuan polisi pamong praja. Jumlah penduduk Kota Bandung tahun 2002 adalah sebanyak 1.868.542 jiwa. Pertumbuhan penduduk rata-rata dari tahun 1997 adalah 1,65%. Pertumbuhan penduduk tertinggi adalah pada tahun 2000 dan pertumbuhan penduduk minus terjadi pada tahun 2001. Berikut ini adalah grafik pertumbuhan penduduk dari tahun 1997-2002 ().

## **2.2 Kriminalitas**

Kriminalitas merupakan asalnya dari kata "*crimen*" yang artinya kejahatan, tindak kriminal, atau juga diartikan suatu tindakan kejahatan, sehingga merupakan tindakan yang bersifat negatif. Seringkali, tindakan ini akan merugikan banyak pihak dan pelaku tindakannya disebut sebagai seorang kriminal.

Kriminalitas sederhananya merupakan segala tindakan atau sesuatu yang dilakukan individu, kelompok, ataupun komunitas yang melanggar hukum atau suatu tindakan kejahatan, sehingga mengganggu keseimbangan atau stabilitas sosial dalam masyarakat.

Menurut Susilo, Definisi kriminalitas dalam pandangannya merupakan suatu perbuatan yang merugikan parakorban, juga masyarakat karena fenomena ini menghilangkan ketentraman dan ketertiban (Susilo, 1998).

### 2.2.1 Ciri-Ciri Kriminalitas

- Sistem Yang Tidak Adil

Secara naluriah, manusia adalah pencemburu, terutama dalam hal kekayaan dan kekuasaan. Oleh sebab itu, kesenjangan yang lahir diikuti oleh aksi-aksi kriminal untuk menyetarakan atau mengimbangnya.

- Suasana atau Lingkungan yang Individualis

Acapkali, individu yang dari lingkungan individualis akan terjun bebas menjadi seorang kriminal. Oleh karena, lingkungan tadi mengarahkannya untuk egois dan menomersatukan dirinya dengan lingkungan sekitar.

- Kemiskinan Yang Diderita

Problem klasik dalam kehidupan dari ciri-ciri kriminal adalah alasan kemiskinan. Seringkali dilontarkan sebagai tameng dan pembenaran dari setiap pelaku kriminal yang tertangkap dan berharap mendapat keringanan dari jeratan hukum.

- Rasa Benci

Tindakan kriminal atau menyimpang yang dilakukan terpancing karena rasa benci di dalam hati seseorang atau kelompok pada seseorang, kelompok atau komunitas yang lain. Hal ini, secara nyata telah banyak terjadi di dalam kehidupan masyarakat.

### 2.3 Pengertian Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat. Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara di mana yang berperan sebagai penggerakya yaitu rakyat yang berada dinegara tersebut.

Kata “*sistem*” banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka.

Jadi bisa diartikan bahwa definisi sistem adalah suatu kumpulan objek atau unsur-unsur atau bagian-bagian yang memiliki arti berbeda-beda yang saling memiliki hubungan, saling berkerjasama dan saling memengaruhi satu sama lain serta memiliki keterikatan pada rencana yang sama dalam mencapai suatu tujuan tertentu pada lingkungan yang kompleks.

Menurut Dr. Harijono, Sistem merupakan gabungan obyek yang memiliki hubungan secara fungsi dan hubungan antara setiap ciri obyek, secara keseluruhan menjadi suatu kesatuan yang berfungsi (Harijono, 2014).

### **2.3.1 Jenis - Jenis Sistem**

Ada berbagai tipe sistem berdasarkan kategori, berikut penjelasannya :

- Atas dasar keterbukaan: sistem terbuka, di mana pihak luar dapat mempengaruhinya. Kemudian sistem tertutup.
- Atas dasar komponen: Sistem fisik, dengan komponen materi dan energi. Kemudian sistem non-fisik atau konsep, berisikan ide-ide.

### **2.3.2 Elemen Sistem**

Adapun elemen dari sistem diantaranya adalah sebagai berikut.

- Objek, merupakan bagian, elemen atau variabel. Ia dapat berupa benda fisik, abstrak atau keduanya.
- Atribut, merupakan penentu kualitas atau sifat kepemilikan sistem dan objeknya.
- Hubungan internal, merupakan penghubungan diantara objek-objek yang terdapat dalam sebuah sistem.
- Lingkungan, merupakan tempat dimana sistem berada.
- Tujuan, Setiap sistem memiliki tujuan dan tujuan inilah yang menjadi motivasi yang mengarahkan sistem.
- Masukan, adalah sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses.

- Proses, Bagian yang melakukan perubahan dari masukan menjadi keluaran.
- Keluaran, adalah hasil dari proses. Pada sistem informasi berupa informasi atau laporan, dll.

#### **2.4 Pengertian Informasi**

Informasi menjadi sebuah hal yang sangat penting saat ini. Informasi berbeda dengan data. Perbedaan data dan informasi adalah, jika data adalah fakta yang masih belum disimpulkan, sedangkan sebuah informasi adalah data yang sudah dihimpun, diolah dan disimpulkan.

Ada banyak fungsi informasi di antaranya yaitu menjadi sumber pengetahuan, sebagai media hiburan, menambah wawasan, sebagai sumber berita, untuk mempengaruhi opini, untuk mempersatukan pendapat dan lain sebagainya. Tujuan informasi penting untuk menyampaikan suatu hal yang terjadi atau sebuah fakta.

Teori mengenai definisi informasi banyak dikemukakan oleh para ahli bahasa, sastra dan komunikasi. Secara umum pengertian informasi menurut para ahli mengarah pada definisi yang sama, meski ada beberapa poin dan unsur yang ditonjolkan masing-masing.

Pengertian informasi secara umum adalah sebuah pesan (ucapan atau ekspresi) atau kumpulan pesan yang terdiri dari order sekuens dari simbol, atau makna yang dapat ditafsirkan dari pesan atau kumpulan pesan. Informasi berarti sekumpulan data atau fakta yang telah diproses dan dikelola sedemikian

rupa sehingga menjadi sesuatu yang mudah dimengerti dan bermanfaat bagi penerimanya.

Sedangkan menurut para ahli diantaranya:

- Romney dan Steinbart mendefinisikan arti informasi sebagai data yang telah diproses dan diorganisasikan, sehingga dapat memberikan arti atau manfaat bagi orang yang menggunakannya (Romney dan Steinbart, 2014).
- Dalam bukunya, Gelinas dan Dull mengatakan bahwa pengertian informasi adalah sebuah data yang disajikan dalam bentuk formulir. Data tersebut memiliki kegunaan dalam pembuatan atau pengambilan keputusan (Gelinas dan Dull, 2012).
- Menurut Tata Sutabri, Informasi merupakan sebuah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan (Sutabri, 2016).



## 2.5 Pengertian Sistem Informasi Geografis

Informasi Geografis merupakan data yang ditempatkan dalam konteks ruang dan waktu. Nah, Sistem Informasi Geografis (SIG) atau *Geographic Information System* (GIS) sendiri merupakan sistem berbasis komputer yang biasanya digunakan untuk menyimpan, memanipulasi, dan menganalisa informasi geografis.

Sebelum adanya Sistem Informasi Geografis (SIG) ini, sejumlah informasi permukaan bumi disajikan dalam peta yang dibuat secara manual. Hadirnya SIG dapat mengolah komponen peta tersebut dalam komputer, kemudian hasilnya berupa peta digital.

SIG dapat menggabungkan berbagai jenis data pada satu titik tertentu yang ada di bumi, menghubungkannya, menganalisanya, hingga memetakan hasilnya. Data yang diolah oleh sistem ini adalah data spasial yakni data yang berorientasi pada geografis. Selain itu juga merupakan lokasi yang mempunyai koordinat tertentu.

Hal tersebut sebagai dasar referensi analisa dan pemetaan hasilnya. Karena itu, aplikasi SIG ini dapat menjawab beberapa pertanyaan tentang geografis bumi seperti lokasi, kondisi, pola, pemodelan, serta tren. Kemampuan ini yang membedakan Sistem Informasi Geografis (SIG) dengan sistem informasi lainnya.

SIG sendiri dikenalkan di Indonesia pada 1972 dengan nama *Data Banks for Development*. Munculnya istilah Sistem Informasi Geografi atau *Geographic*

*Information System* sendiri setelah dicetuskan oleh *General Assembly* dari *International Geographical Union* di Ottawa, Kanada pada 1967.

Awalnya, sistem ini merupakan sebuah inisiatif untuk mengetahui kemampuan lahan di wilayah pedesaan Kanada. Caranya, dengan memetakan beberapa informasi seperti tanah, alam bebas, pariwisata, pertanian, unggas, pada skala 1:250.000.

### **2.5.1 Jenis Data Sistem Informasi Geografis**

Paling tidak ada 4 jenis data yang dikenal dalam Sistem Informasi Geografis, yakni:

- **Data Spasial**

Data ini merepresentasikan dan/atau mengidentifikasi posisi ruang (letak geografis) dari suatu fenomena. Contoh data spasial seperti letak suatu daratan, informasi garis lintang dan garis bujur, kepulauan, sumber minyak, hutan, sumber gas alam, pegunungan, serta lainnya. Data spasial ini dapat digunakan untuk mengidentifikasi lokasi, misalnya Kode Pos.

- **Data Atribut**

Data atribut merupakan data yang menjabarkan aspek dari suatu fenomena dalam bentuk deskripsi atau penjelasan yang terperinci. Data ini tergambar dalam bentuk kata-kata, angka, serta tabel. Data atribut yang dapat dijumpai pada data kepadatan penduduk, data luas wilayah, jenis-jenis tanah, data demografis, dan sebagainya.

- **Data Vektor**

Data vektor adalah data yang direpresentasikan sebagai suatu mozaik berupa titik/point, garis (*arc/line*), polygon yaitu daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama, serta nodes yaitu titik perpotongan antara dua garis. Kegunaan data vektor ini untuk menganalisa ketepatan posisi pada suatu wilayah atau mendefinisikan hubungan spasial dari beberapa fitur.

- Data Raster

Data raster atau sering juga disebut dengan sel grid merupakan data yang dihasilkan dari sistem penginderaan jauh. Pada data raster, objek geografis direpresentasikan sebagai struktur sel grid yang disebut dengan pixel (*picture element*). Resolusi pada data raster tergantung pada ukuran pixelnya. Nah, dengan kata lain resolusi menggambarkan ukuran sebenarnya di permukaan bumi yang diwakili oleh setiap pixel pada citra.

## 2.6 Google Maps API

*Application Programming Interface* (API) adalah sekumpulan komponen yang dibuat dalam kelas-kelas yang memiliki berbagai fungsi tertentu. Dengan demikian dapat disimpulkan Google Maps API merupakan sekumpulan komponen untuk melakukan berbagai fungsi terkait aplikasi pemetaan. Google Maps API dibuat dengan Javascript, sehingga API ini akan lebih mudah dipelajari bagi orang yang sudah mengenal *Javascript*, khususnya lagi yang berpengalaman dengan pemrograman berorientasi objek.

Google Maps adalah salah satu produk berbasis geospasial dari Google selain, *Google Earth Engine* dan *Google Earth KML*. Google Maps pertama kali

didesain dan dibuat oleh dua programmer Denmark yaitu Lars Rasmussen dan Jeins Eilstrup, di Sydney, Australia pada perusahaan yang bernama *Where 2 Technologies*. Perusahaan ini kemudian diakuisisi oleh Google pada tahun 2004. Google secara resmi mengumumkan Google Maps pada tahun 2005 melalui blog google. Sedangkan Google Maps API sendiri baru direlease pada bulan Juni 2005.

Ada dua lisensi dari Google Maps API yaitu standar dan bisnis. Untuk versi standar dibatasi maksimum 25000 akses. Pembatasan ini mulai dilakukan oleh Google pada September 2013 (Rahmad, 2016).

## **2.7 Definisi Pemetaan**

Pemetaan adalah kegiatan pendokumentasian atau perekaman data dalam bentuk grafis keletakan dan lokasi cagar budaya serta lingkungannya. Kegiatan pemetaan dimaksudkan untuk mengetahui gambaran situasi keberadaan cagar budaya yang dapat dipakai sebagai acuan dalam menentukan kebijakan lebih lanjut berkaitan dengan aspek pelestariannya.

## **2.8 MVC (Model-View-Controller)**

MVC adalah konsep arsitektur dalam pembangunan aplikasi berbasis web yang membagi aplikasi web menjadi 3 bagian besar. Yang mana setiap bagian memiliki tugas-tugas serta tanggung jawab masing-masing. Tiga bagian tersebut adalah: model, view dan controller (Pablo, 2017).

- Model: Bertugas untuk mengatur, menyiapkan, memanipulasi dan mengorganisasikan data (dari database) sesuai dengan instruksi dari controller.

- View: Bertugas untuk menyajikan informasi (yang mudah dimengerti) kepada user sesuai dengan instruksi dari controller.
- Controller: Bertugas untuk mengatur apa yang harus dilakukan model, dan view mana yang harus ditampilkan berdasarkan permintaan dari user. Namun, terkadang permintaan dari user tidak selalu memerlukan aksi dari model. Misalnya seperti menampilkan halaman form untuk registrasi user, *controller* juga berfungsi untuk menerima *request* dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

## 2.9 Database

Penggunaan teknologi dalam sebuah perusahaan, institusi ataupun organisasi mempunyai peranan penting guna mencapai tujuan. Suatu perusahaan dituntut untuk bekerja se-efisien mungkin supaya bisa bertahan di atas kerasnya persaingan. Salah satu teknologi yang harus dimiliki oleh sebuah perusahaan, institusi maupun organisasi adalah teknologi dalam memproses data sehingga menjadi informasi yang berguna, teknologi yang dimaksud adalah sistem pengolahan basis data atau database. Penggunaan database yang baik pada perusahaan retail misalnya, mampu membantu seorang kasir bekerja lebih cepat ketika mencari jumlah barang atau harga barang yang akan dijual. Begitupun dengan admin, database mempermudah ketika pencarian stok persediaan, barang paling laku dan banyak lagi yang lainnya.

Beberapa contoh aplikasi yang membutuhkan database sebagai landasannya antara lain: transaksi perbankan, pemesanan tiket, aplikasi

pemrosesan penjualan dan pembelian pada perusahaan dagang, absensi perusahaan serta sistem penggajian karyawan pada perusahaan, aplikasi akademik, aplikasi pencatatan pajak, dan lain sebagainya. Selain dapat meningkatkan kinerja sebuah perusahaan, penggunaan database masih memiliki banyak keuntungan lain yang bisa kita dapatkan.

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, 2007).

### **2.9.1 Manfaat Penggunaan Database**

- Kecepatan dan Kemudahan

Database memiliki kemampuan dalam menyeleksi data sehingga menjadi suatu kelompok yang terurut dengan cepat. Hal inilah yang akhirnya dapat menghasilkan informasi yang dibutuhkan secara cepat pula. Seberapa cepat pemrosesan data oleh database tergantung pula pada perancangan databasenya.

- Pemakaian Bersama-sama

Suatu database bisa digunakan oleh siapa saja dalam suatu perusahaan. Sebagai contoh database mahasiswa dalam suatu perguruan tinggi dibutuhkan oleh beberapa bagian, seperti bagian admin, bagian keuangan, bagian akademik. Kesemua bidang tersebut membutuhkan

database mahasiswa namun tidak perlu masing-masing bagian membuat databasenya sendiri, cukup database mahasiswa satu saja yang disimpan di server pusat. Nanti aplikasi dari masing-masing bagian bisa terhubung ke database mahasiswa tersebut.

- Kontrol data terpusat

Masih berkaitan dengan point ke dua, meskipun pada suatu perusahaan memiliki banyak bagian atau divisi tapi database yang diperlukan tetap satu saja. Hal ini mempermudah pengontrolan data seperti ketika ingin mengupdate data mahasiswa, maka kita perlu mengupdate semua data di masing-masing bagian atau divisi, tetapi cukup di satu database saja yang ada di server pusat.

- Menghemat biaya perangkat

Dengan memiliki database secara terpusat maka di masing-masing divisi tidak memerlukan perangkat untuk menyimpan database terhubung database yang dibutuhkan hanya satu yaitu yang disimpan di server pusat, ini tentunya memangkas biaya pembelian perangkat.

- Keamanan Data

Hampir semua Aplikasi manajemen database sekarang memiliki fasilitas manajemen pengguna. Manajemen pengguna ini mampu membuat hak akses yang berbeda-beda disesuaikan dengan kepentingan maupun posisi pengguna. Selain itu data yang tersimpan di database diperlukan password untuk mengaksesnya.

- Memudahkan dalam pembuatan Aplikasi baru

Dalam poin ini database yang dirancang dengan sangat baik, sehingga si perusahaan memerlukan aplikasi baru tidak perlu membuat database yang baru juga, atau tidak perlu mengubah kembali struktur database yang sudah ada. Sehingga Si pembuat aplikasi atau programmer hanya cukup membuat atau pengatur antarmuka aplikasinya saja.

## 2.10 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.

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, 2019).

### **2.11 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. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan symbol.

Menurut salah satu para ahli, Brady dan Loonam, *Entity Relationship diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analys dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database* (Brady dan Loonam, 2010).

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

UML merupakan singkatan dari “*Unified Modelling Language*” yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem software. 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 dikembangkan oleh 3 pendekar “berorientasi objek”, yaitu Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. 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. (Hermawan, 2005).

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, Scholz, Huemer, Kappel, 2015).

### **2.12.1 Diagram Use Case**

*Use Case Diagram* adalah gambaran grafis dari beberapa atau semua actor, use case, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detail tentang penggunaan use case, tetapi hanya memberi gambaran singkat hubungan antara use case, aktor, dan sistem. Di dalam use case ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat (Aditya, 2019).

### **2.12.2 Diagram Activity**

*Activity diagram*, sesuai dengan namanya diagram ini menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah – langkah dalam proses kerja sistem yang akan dibuat (Aditya, 2019).

### **2.12.3 Diagram Sequence**

*Sequence Diagram* adalah salah satu dari diagram - diagram yang ada pada UML, sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem (Aditya, 2019).

#### **2.12.4 Diagram Class**

*Class Diagram* adalah salah satu jenis diagram yang paling berguna di UML, hal ini karena dapat dengan jelas memetakan struktur sistem tertentu dengan memodelkan kelas, atribut, operasi serta hubungan antar objek.

Class Diagram menggambarkan serta deskripsi atau penggambaran dari class, atribut, dan objek disamping itu juga hubungan satu sama lain seperti pewarisan, containmet, asosiasi dan lainnya.

#### **2.13 RAD ( Rapid Application Development)**

*Rapid application development* (rad) atau *rapid prototyping* adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *incremental* (bertingkat). Rad menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini. *Rapid application development* menggunakan metode *iteratif* (berulang) dalam mengembangkan sistem dimana *working model* (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) user dan selanjutnya disingkirkan. *Working model* digunakan kadang-kadang saja sebagai basis desain dan implementasi sistem final.

##### **2.13.1 Tahapan RAD**

1. *Requirement planning*, dalam tahap ini diketahui apa saja yang menjadi kebutuhan sistem yaitu dengan mengidentifikasi kebutuhan informasi dan masalah yang dihadapi untuk menentukan tujuan, batasan-batasan sistem, kendala dan juga alternatif pemecahan

masalah. Analisis digunakan untuk mengetahui kebutuhan pengembangan sistem dan juga untuk mengetahui aktivitas apa saja yang dibutuhkan dalam sistem tersebut.

2. *Design workshop*, yaitu mengidentifikasi solusi alternatif dan memilih solusi yang terbaik. Kemudian membuat desain proses bisnis dan desain pemrograman untuk data-data yang telah didapatkan dan dimodelkan dalam arsitektur sistem informasi. *Tools* yang digunakan dalam pemodelan sistem biasanya menggunakan *unified modeling language* (uml).
3. *Implentation*, setelah *design workshop* dilakukan, selanjutnya sistem diimplementasikan (*coding*) ke dalam bentuk yang dimengerti oleh mesin yang diwujudkan dalam bentuk program atau unit program. Tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan.

## **BAB III**

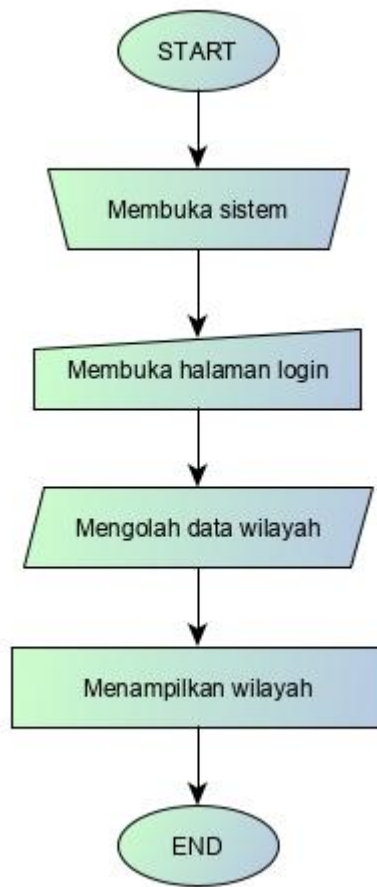
### **REQUIREMENT PLANNING & DESIGN WORKSHOP**

#### **3.1 Requirement Planning (Perencanaan Persyaratan)**

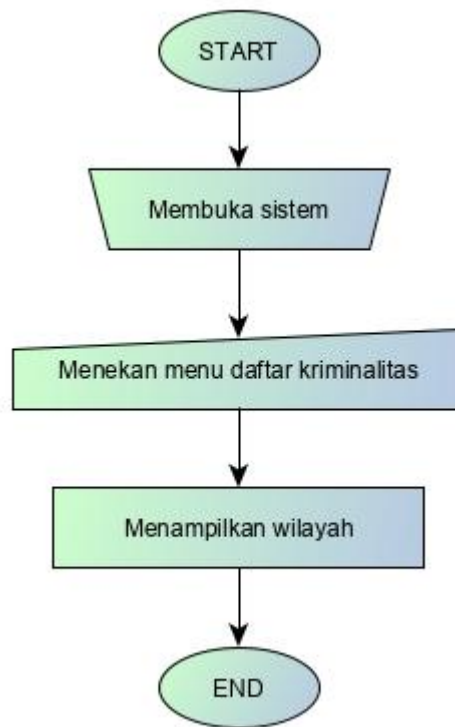
Pada tahap pertama disini akan dilakukan analisa dan pertemuan untuk menentukan kebutuhan pengembangan. Analisa kebutuhan pengembangan akan mengumpulkan data dan syarat-syarat informasi yang didapatkan dari hasil observasi untuk dapat memudahkan penulis dalam mengidentifikasi kebutuhan yang dibutuhkan dalam penelitian ini. Analisa kebutuhan pengembangan meliputi beberapa tahapan yaitu analisa terhadap sistem berjalan, analisa mengenai kelebihan dan kekurangan sistem, serta analisa kebutuhan pendukung sistem.

##### **3.1.1 Analisa Sistem Berjalan**

Analisa sistem berjalan akan menjelaskan bagaimana sistem akan berjalan nantinya. Pada analisa sistem berjalan ini akan penulis gambarkan ke diagram flowchart yang dapat dilihat sebagai berikut:



Gambar: 3.1 Flowchart Admin



Gambar: 3.2 Flowchart User

### 3.1.2 Analisa Kelebihan Sistem

Setelah dilakukan peninjauan terhadap analisa sistem berjalan diatas, penulis dapat menjabarkan beberapa kelebihan yang terdapat di dalam sistem ini. Berikut adalah kelebihan sistem informasi geografis pemetaan daerah rawan kriminalitas:

1. Sistem ini sudah dapat mengklasifikasikan jenis kriminalitas secara otomatis.
2. Sistem ini dapat memberi informasi berupa berapa radius wilayah yang rawan terjadi kriminalitas.



3. Sistem ini dapat diakses dimanapun dan kapanpun karena dibentuk kedalam basis website.

### **3.1.3 Analisa Kekurangan Sistem**

Selain dari pada kelebihan, terdapat juga kekurangan di dalam sitem ini, yaitu sistem ini belum bersifat realtime sehingga belum dapat mengolah informasi secara otomatis tetapi masih diharuskan olah data di dalam sistem oleh admin.

### **3.1.4 Analisa Kebutuhan Pendukung Sistem**

Dalam menjalankan penelitian ini, dibutuhkan beberapa perangkat dalam menunjang kelancaran pembuatan sistem ini seperti perangkat keras, perangkat lunak dan juga kebutuhan-kebutuhan lainnya yang dapat mendukung keberhasilan pembuatan sistem ini. Berikut adalah kebutuhan pendukung dalam melakukan pembangunan sistem informasi geografis pemetaan daerah rawan kriminalitas:

- a) Perangkat Keras
  - PC atau Laptop
  - Modem
- b) Perangkat Lunak
  - Browser
  - Sistem Operasi
  - Text Editor
  - Web service
  - Internet

c) Input Sistem

- Data Spasial Wilayah
- Data Wilayah
- Data Admin

d) Process Sistem

- Tambah, ubah, hapus, lihat data spasial wilayah
- Tambah, ubah, hapus, lihat data wilayah
- Tambah, ubah, hapus, lihat data admin

e) Output Sistem

- Wilayah Rawan Kriminalitas

### **3.2 Design Workshop**

Tahap ini adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi sebagai perancangan sistem dapat dirancang dalam bentuk bagan, yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem.

*Design workshop* sistem disini menentukan bagaimana suatu sistem akan menyelesaikan apa yang harus diselesaikan tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari analisa kebutuhan pendukung sistem pada tahap sebelumnya, sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem.

### 3.2.1 Unified Modelling Language (UML)

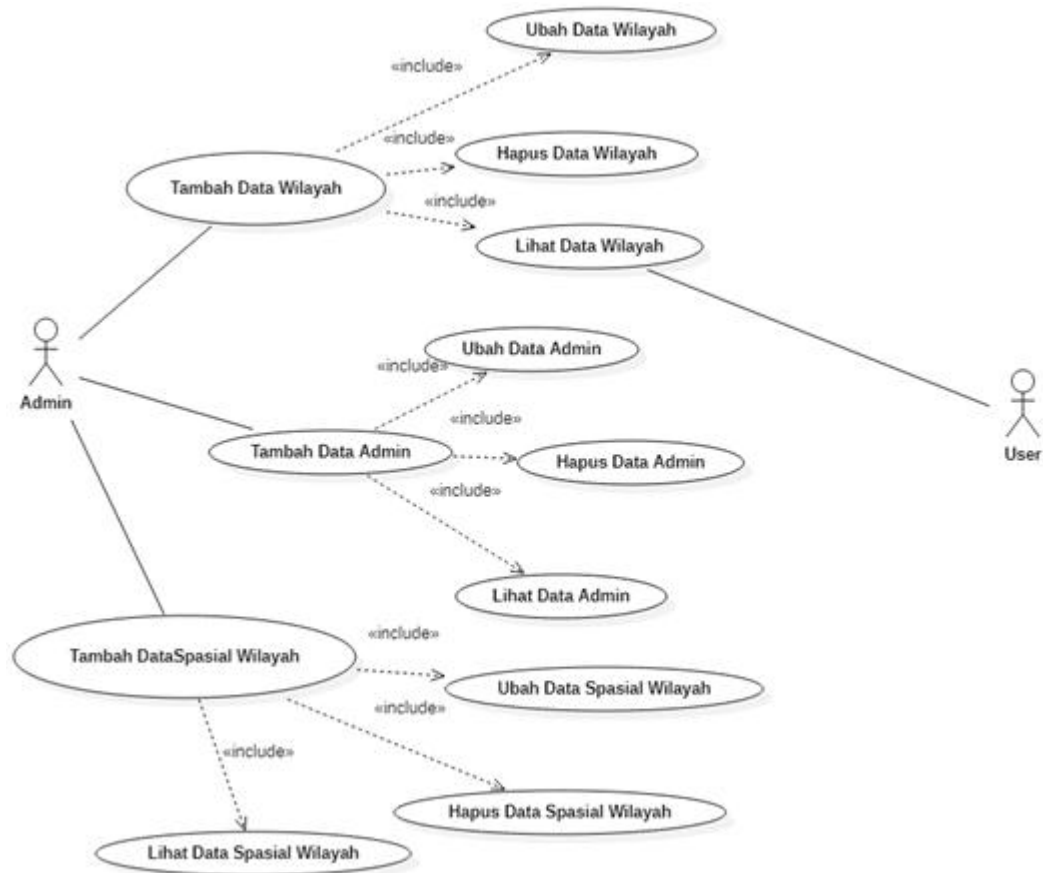
Perancangan terhadap alur sistem disini akan dibuat ke dalam UML diantaranya adalah *use case* diagram, *activity* diagram, *sequence* diagram dan *class* diagram yang dapat dilihat sebagai berikut:

#### 3.2.1.1 Use Case Diagram

Diagram use case akan menggambarkan sekelompok use cases dan aktor yang disertai dengan hubungan diantaranya. Diagram use cases ini menjelaskan dan menerangkan kebutuhan / *requirement* yang diinginkan dan dikehendaki user/pengguna, serta sangat berguna dalam menentukan struktur organisasi dan model dari pada sebuah sistem.

Tabel: 3.1 Penjelasan Aktor Use Case

Nama	Penjelasan
<i>Admin</i>	<i>Admin</i> akan mengolah seluruh data di dalam sistem untuk nantinya dapat dijadikan sebuah informasi kepada <i>user</i> .
<i>User</i>	<i>User</i> akan menerima informasi melalui admin berupa daerah yang rawan akan terjadinya kriminalitas.



Gambar: 3.3 Use Case Diagram

Tabel: 3.2 Skenario Use Case Diagram

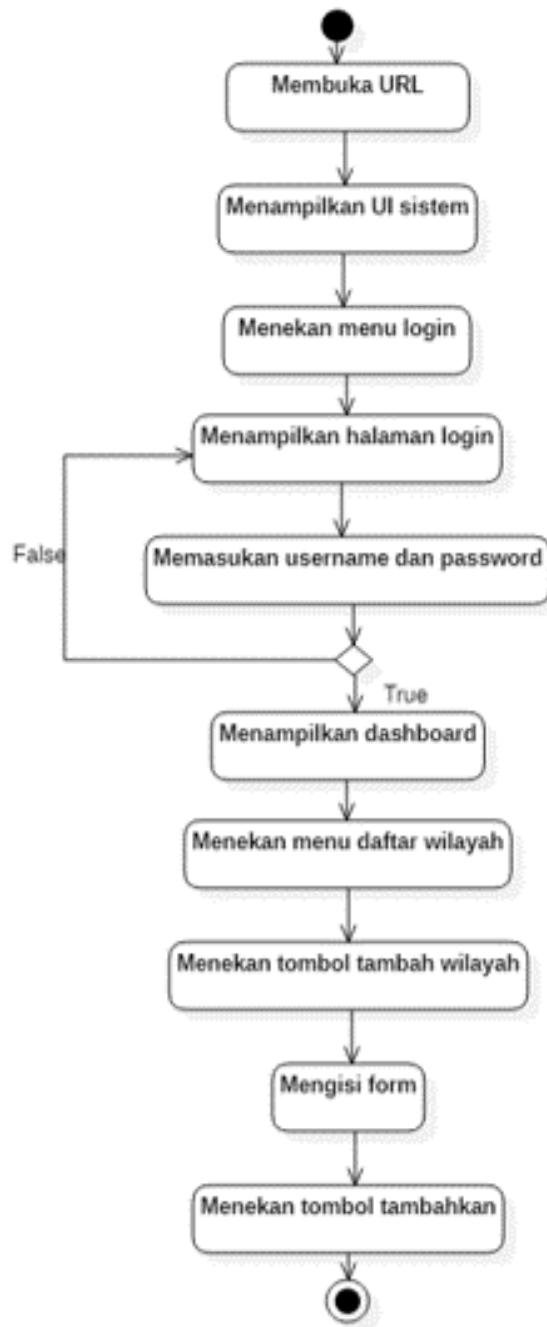
No	Use Case	Aktor	Skenario
1	Login	Admin	1) Membuka login page 2) Memasukan username dan password
2	Tambah data wilayah	Admin	1) Melakukan login 2) Menekan menu daftar wilayah 3) Mengisi form tambah data wilayah
3	Ubah data wilayah	Admin	1) Melakukan login

			<ul style="list-style-type: none"> <li>2) Menekan menu daftar wilayah</li> <li>3) Mengisi form ubah data wilayah</li> </ul>
4	Hapus data wilayah	Admin	<ul style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu daftar wilayah</li> <li>3) Menekan tombol hapus</li> <li>4) Menekan tombol ya</li> </ul>
5	Lihat data wilayah	Admin	<ul style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu daftar wilayah</li> <li>3) Menekan tombol lihat</li> </ul>
6	Tambah data admin	Admin	<ul style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu admin</li> <li>3) Mengisi form tambah data admin</li> </ul>
7	Ubah data admin	Admin	<ul style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu admin</li> <li>3) Mengisi form ubah data admin</li> </ul>
8	Hapus data admin	Admin	<ul style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu admin</li> <li>3) Menekan tombol hapus</li> <li>4) Menekan tombol ya</li> </ul>
9	Lihat data admin	Admin	<ul style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu admin</li> <li>3) Menekan tombol lihat</li> </ul>
10	Tambah Data Spasial Wilayah	Admin	<ul style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu data spasial</li> <li>3) Mengisi form tambah data spasial</li> </ul>
11	Ubah Data Spasial Wilayah	Admin	<ul style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu data spasial</li> <li>3) Mengisi form ubah data spasial</li> </ul>

12	Hapus Data Spasial Wilayah	Admin	<ol style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu data spasial</li> <li>3) Menekan tombol hapus</li> <li>4) Menekan tombol ya</li> </ol>
13	Lihat Data Spasial Wilayah	Admin	<ol style="list-style-type: none"> <li>1) Melakukan login</li> <li>2) Menekan menu data spasial</li> <li>3) Menekan tombol lihat</li> </ol>
14	Lihat data wilayah	User	<ol style="list-style-type: none"> <li>1) Membuka sistem</li> <li>2) Menekan menu wilayah</li> </ol>

### 3.2.1.2 Activity Diagram

Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *Statechart*. Diagram *Statechart* berfokus pada obyek yang dalam suatu proses (atau proses menjadi suatu obyek), diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.



Gambar: 3.4 Activity Diagram Tambah Data Wilayah



Gambar: 3.5 Activity Diagram Ubah Data Wilayah

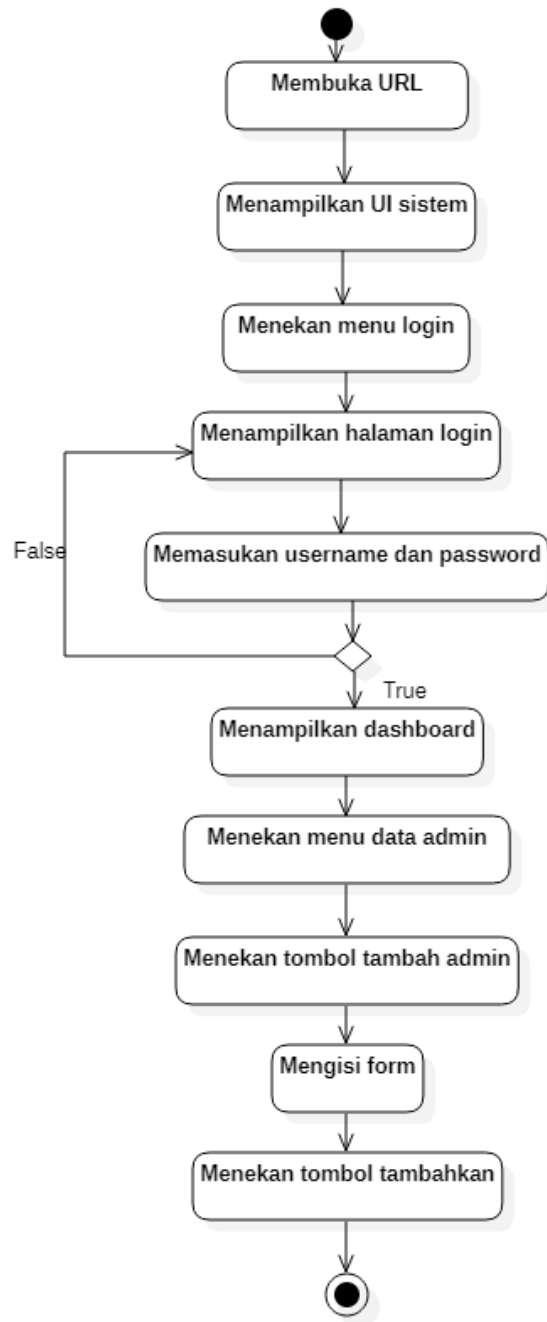




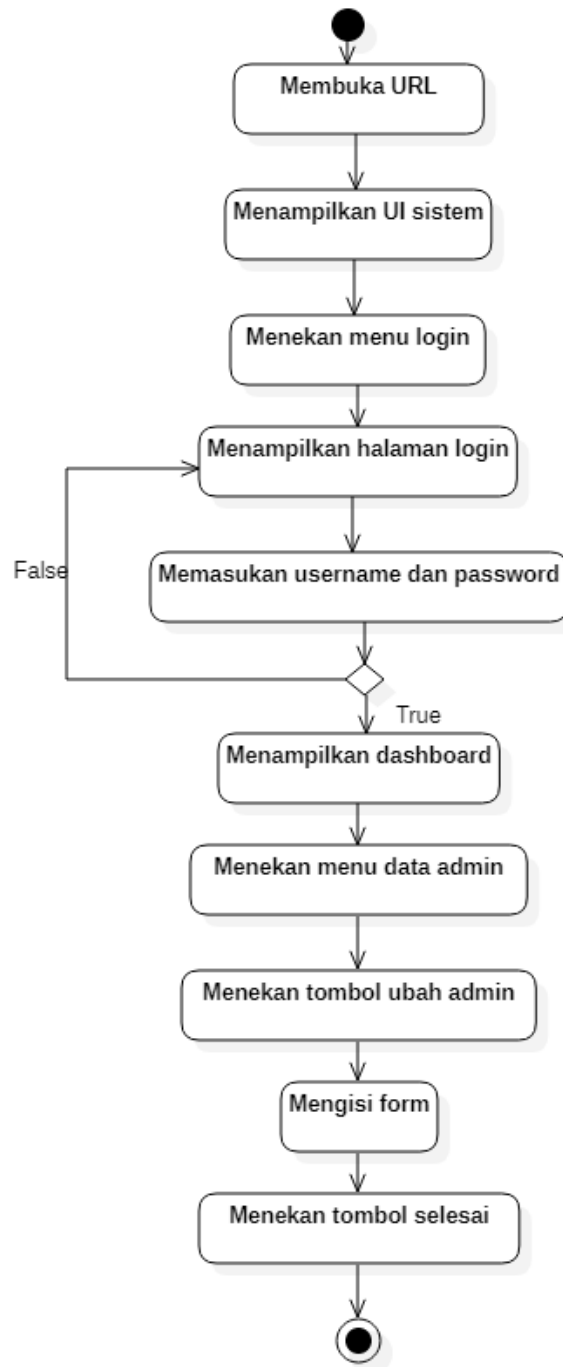
Gambar: 3.6 Activity Diagram Hapus Data Wilayah



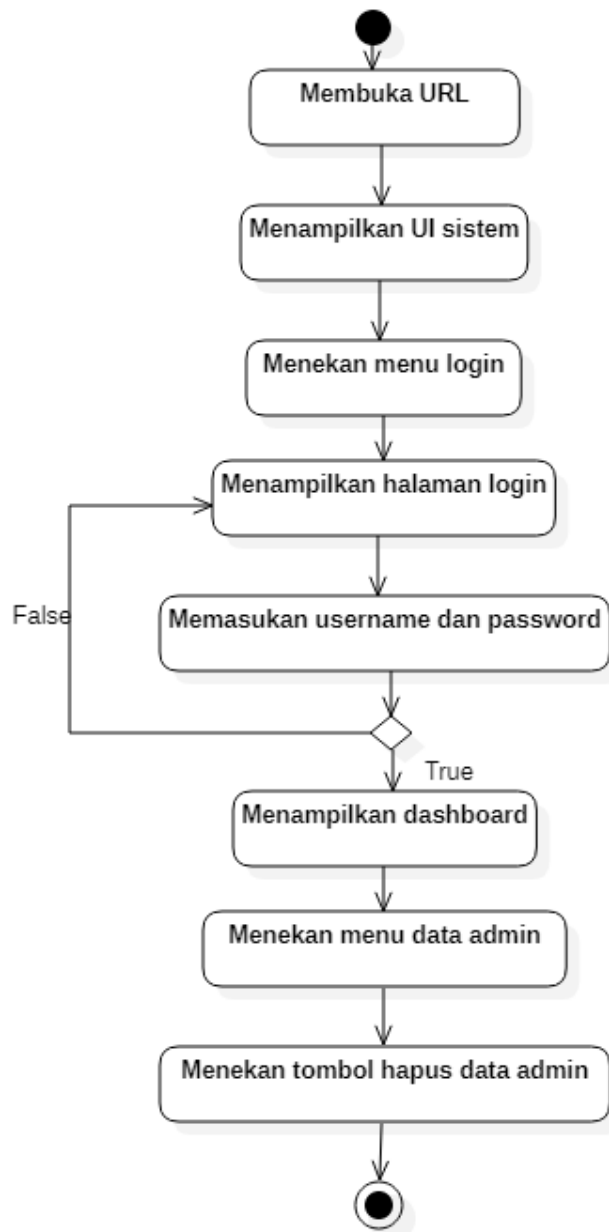
Gambar: 3.7 Activity Diagram Lihat Data Wilayah



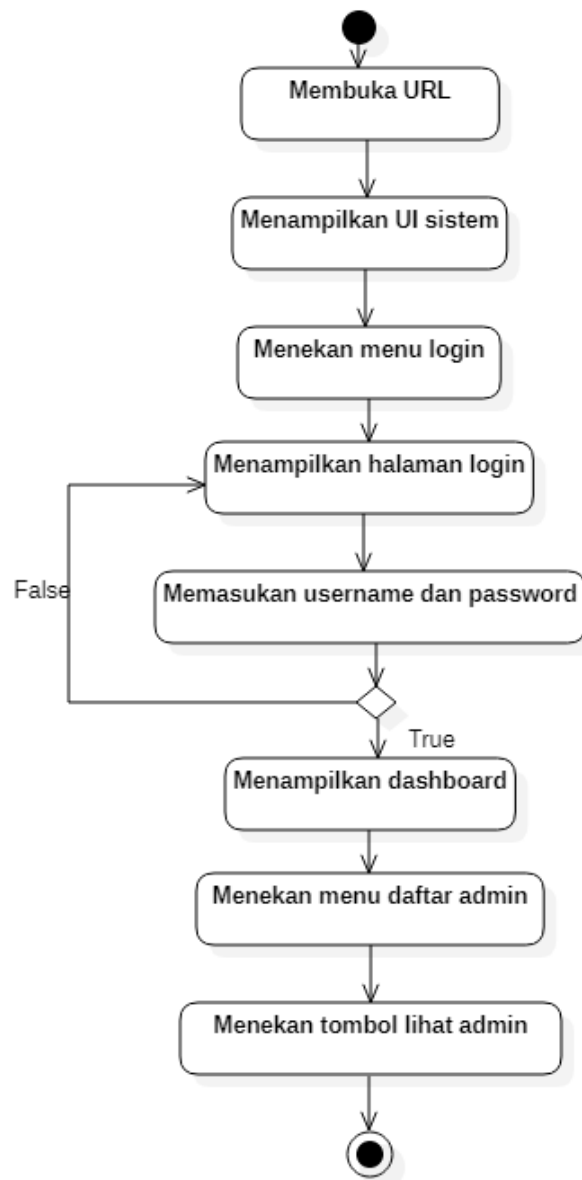
Gambar: 3.8 Activity Diagram Tambah Admin



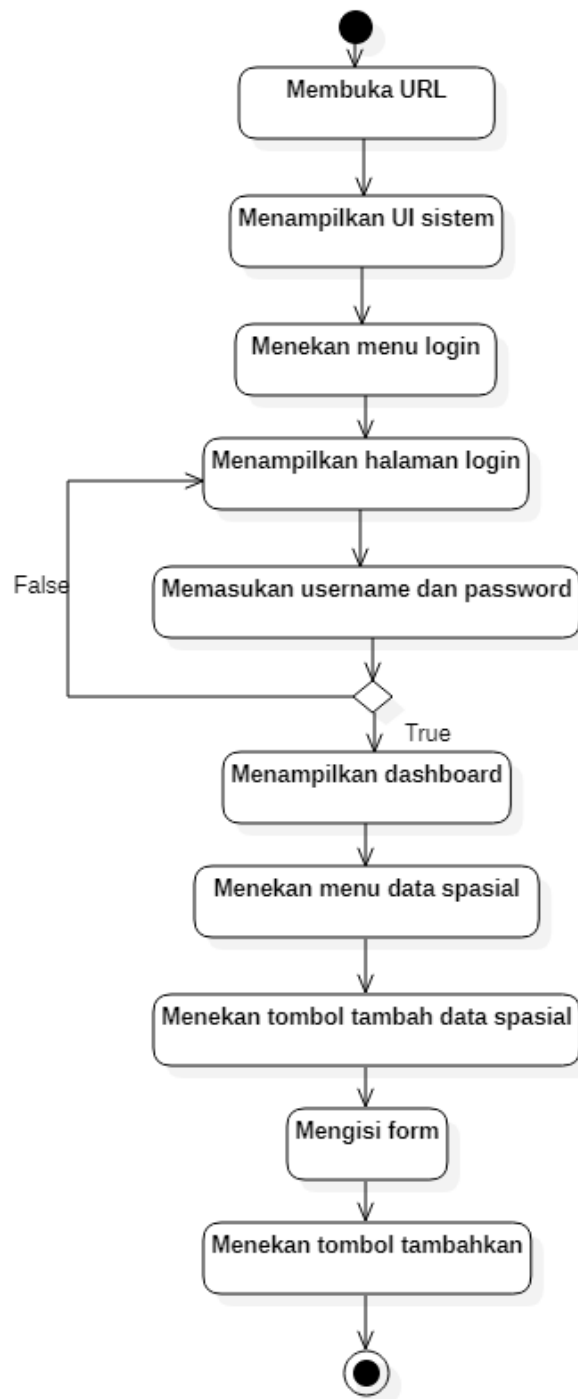
Gambar: 3.9 Activity Diagram Ubah Admin



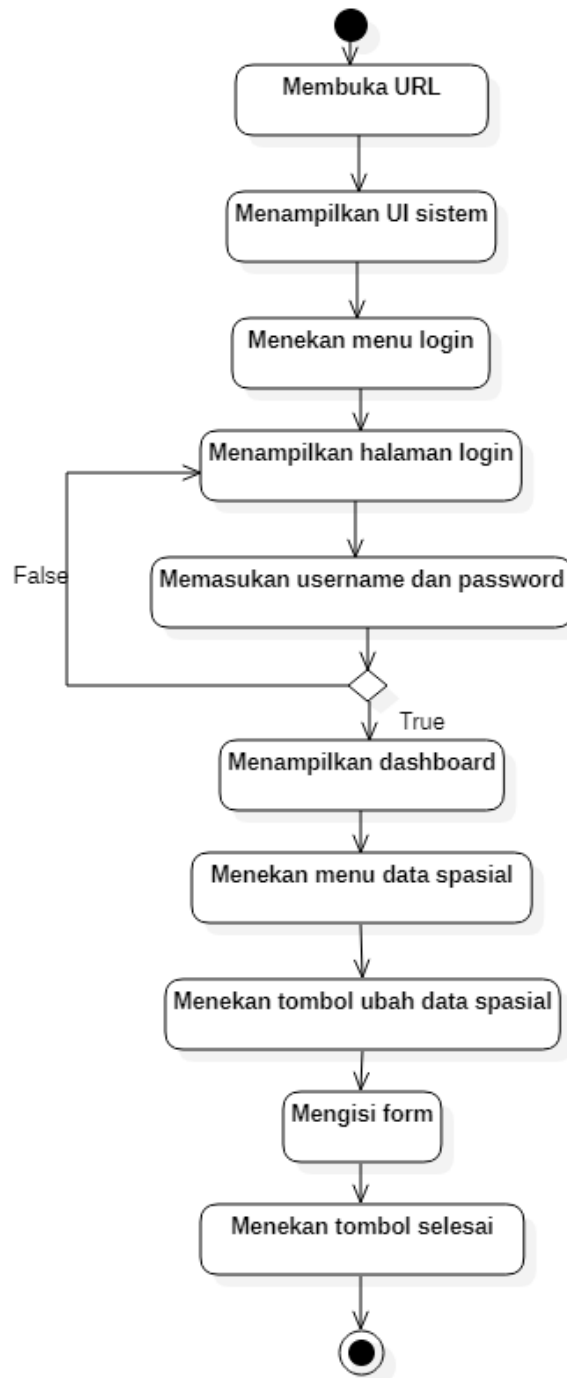
Gambar: 3.10 Activity Diagram Hapus Admin



Gambar: 3.11 Activity Diagram Lihat Admin

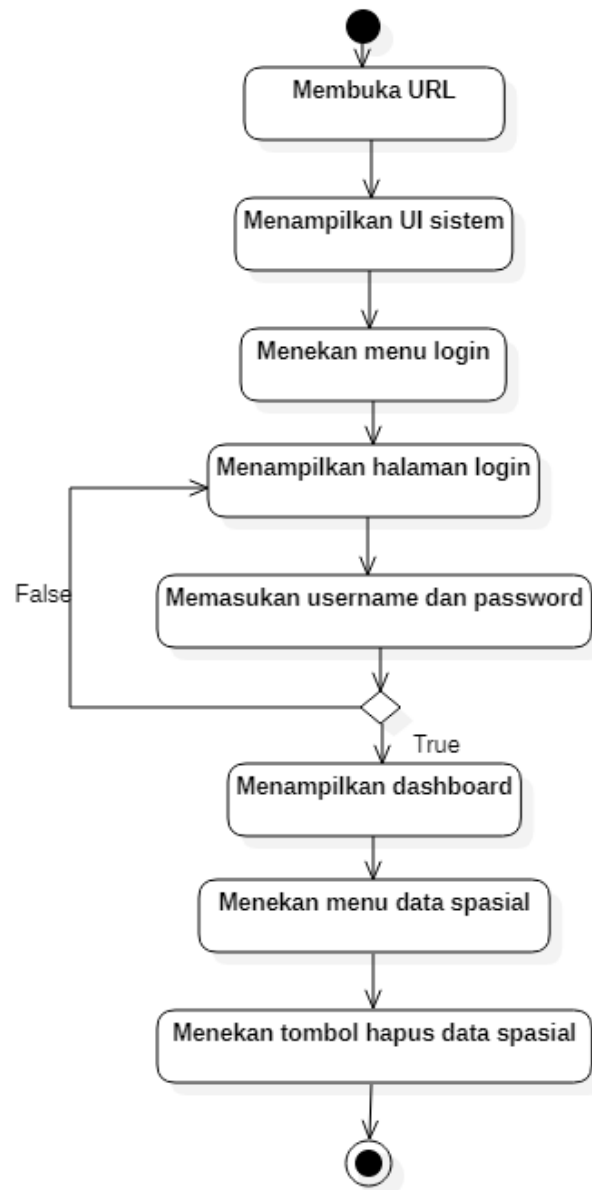


Gambar: 3.12 Activity Diagram Tambah Data Spasial

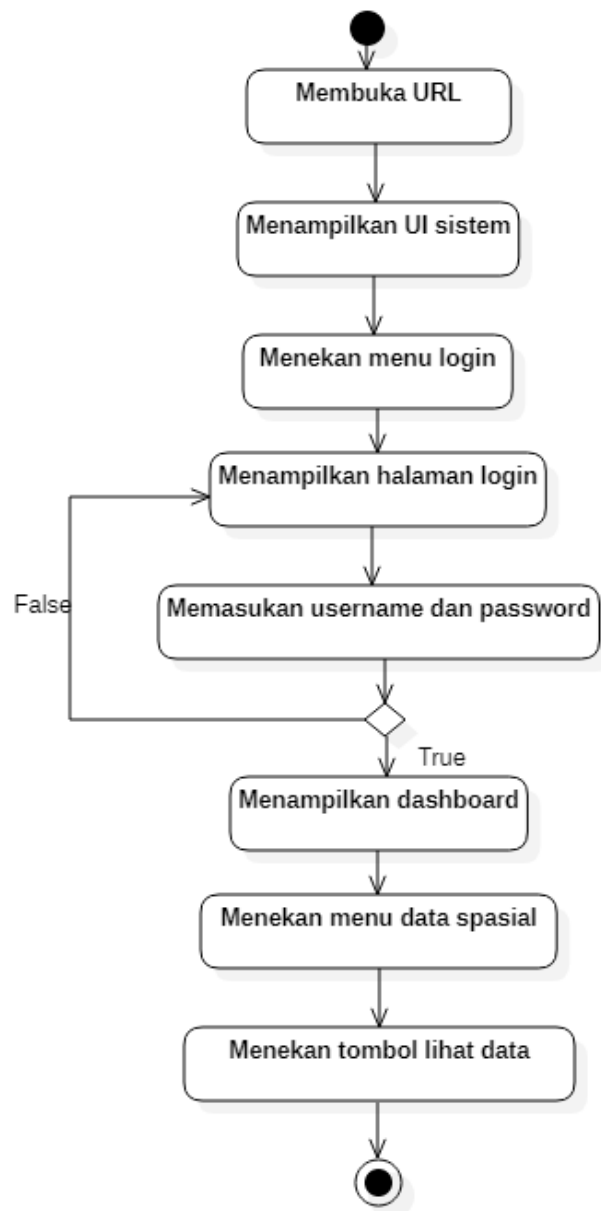


Gambar: 3.13 Activity Diagram Ubah Data Spasial

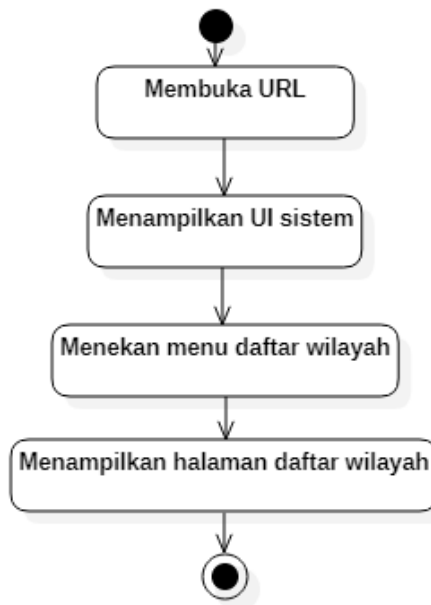




Gambar: 3.14 Activity Diagram Hapus Data Spasial



Gambar: 3.15 Activity Diagram Lihat Data Spasial



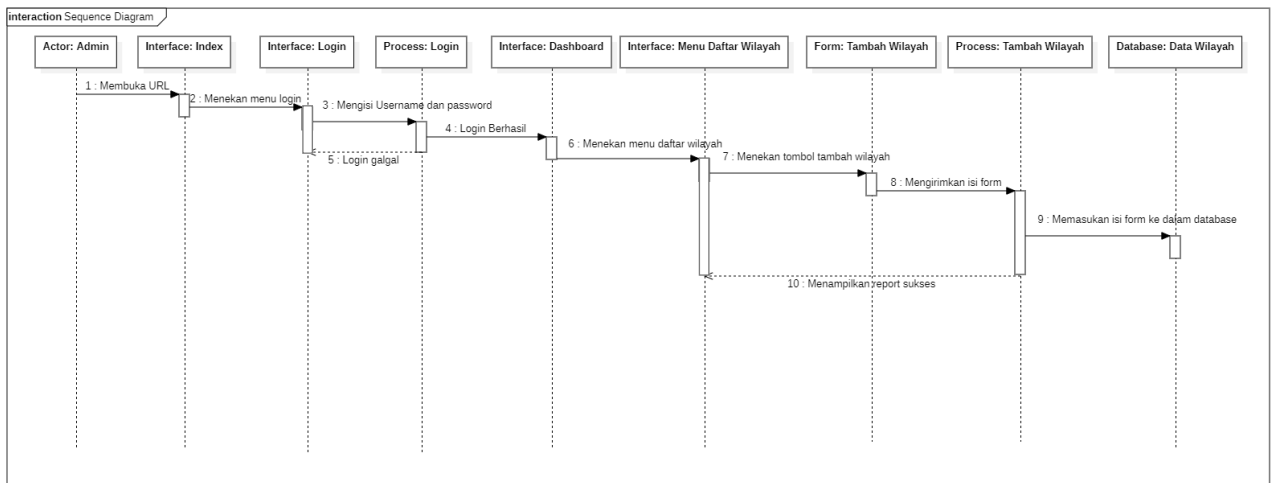
Gambar: 3.16 Activity Diagram User Lihat Daftar Wilayah



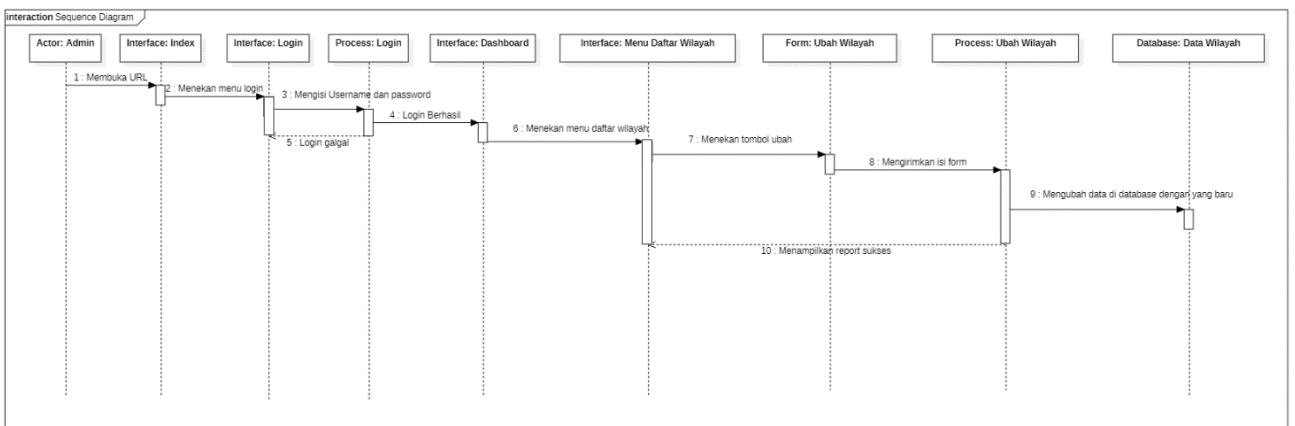
Gambar: 3.17 Activity Diagram Login

### 3.2.1.3 Sequence Diagram

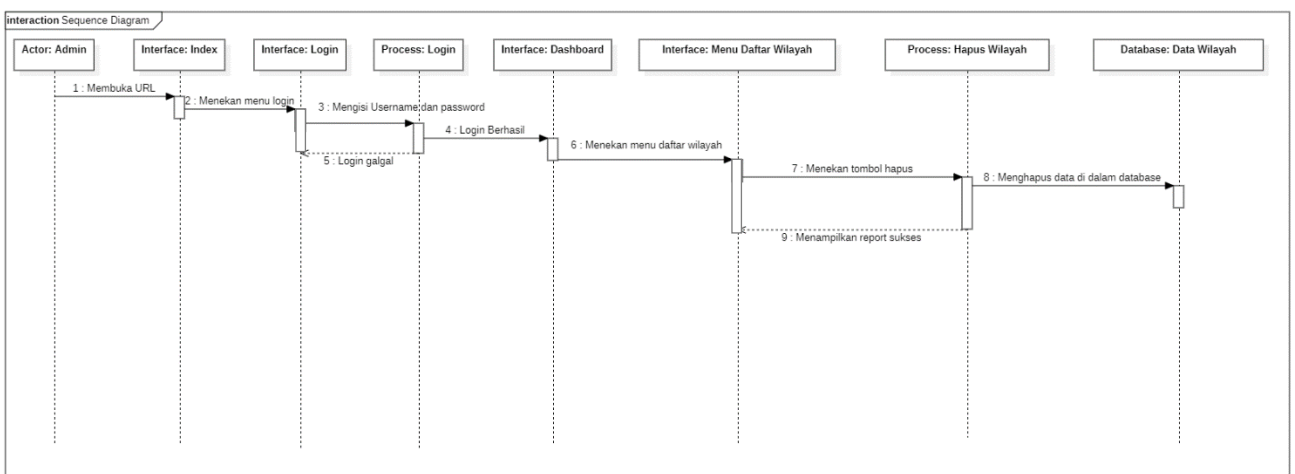
Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.



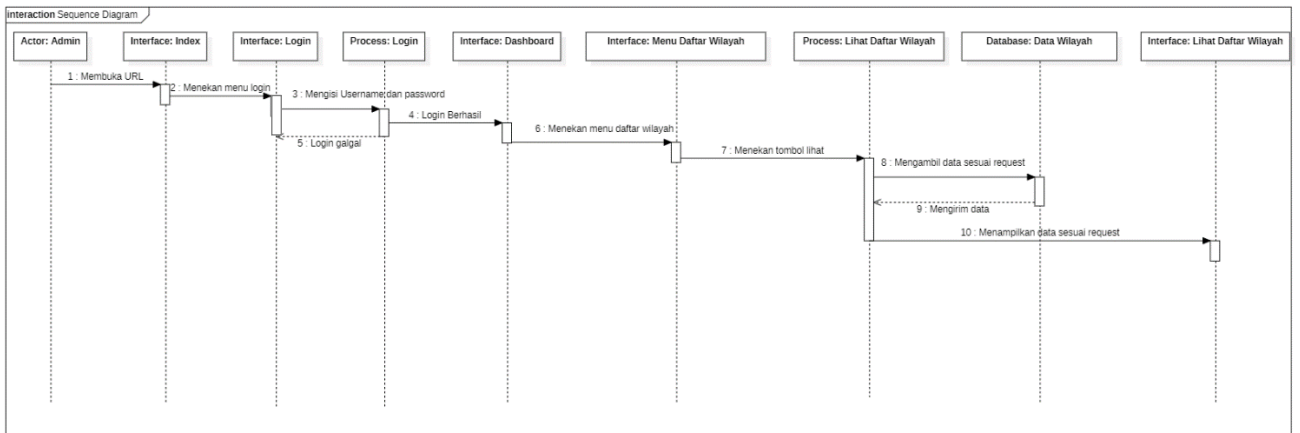
Gambar: 3.18 Sequence Diagram Tambah Daftar Wilayah



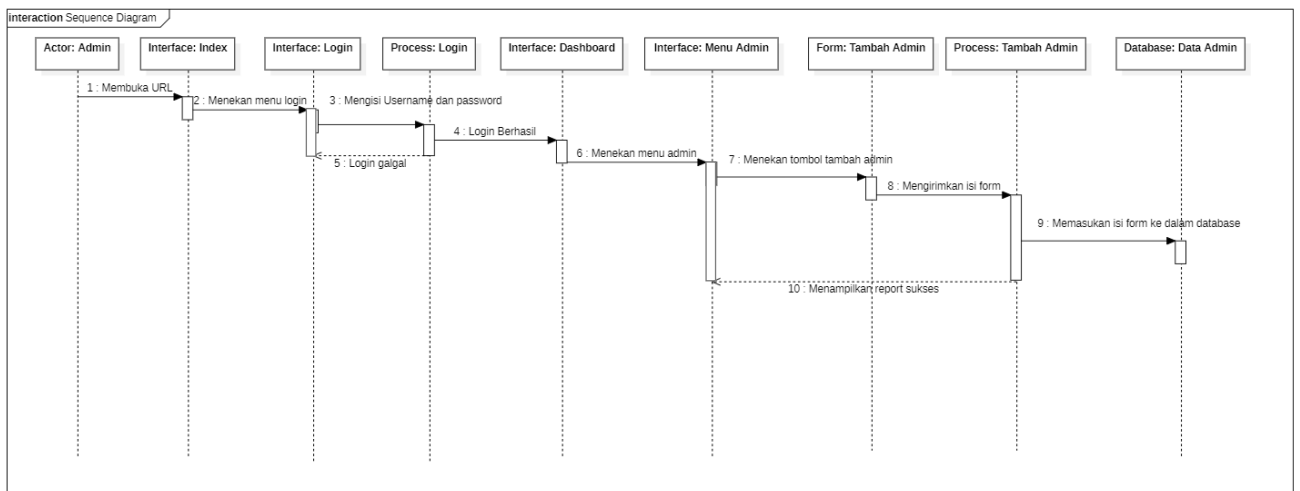
Gambar: 3.19 Sequence Diagram Ubah Daftar Wilayah



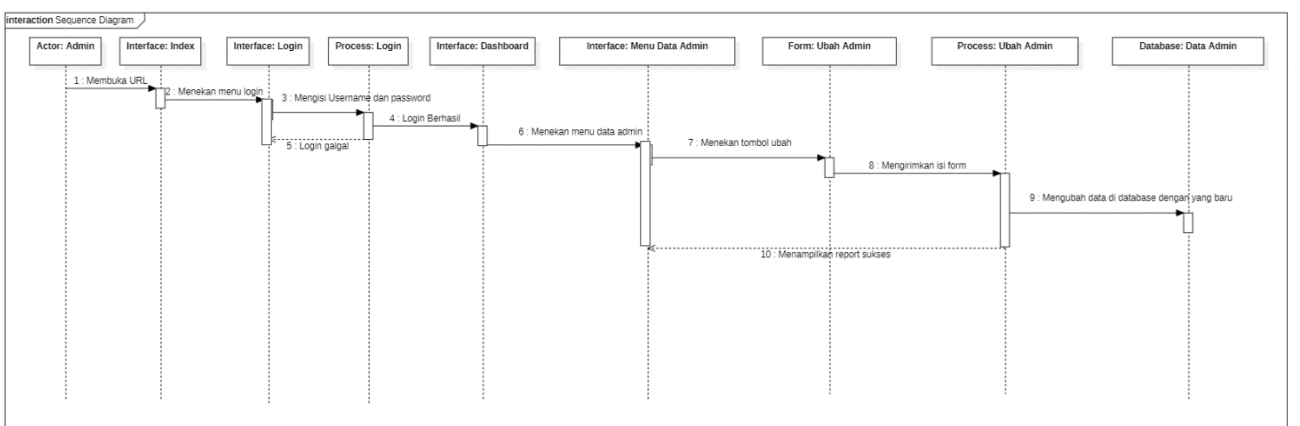
Gambar: 3.20 Sequence Diagram Hapus Daftar Wilayah



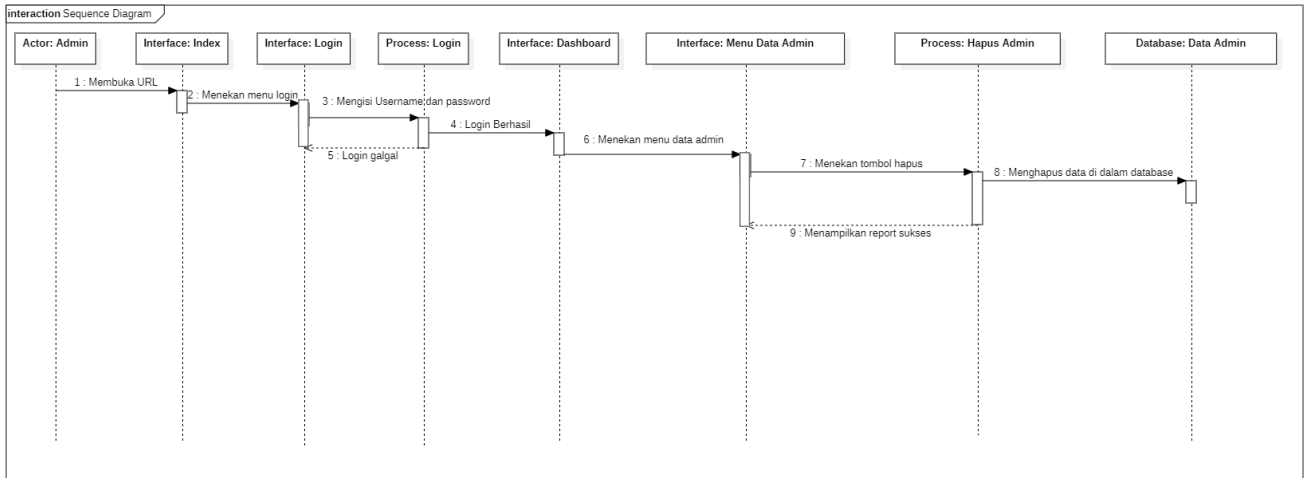
Gambar: 3.21 Sequence Diagram Lihat Daftar Wilayah



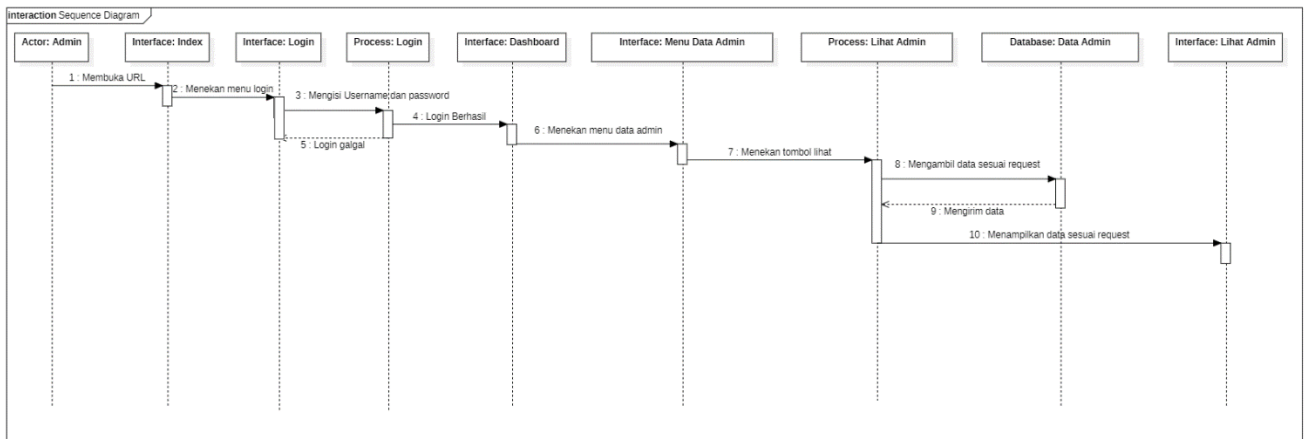
Gambar 3.22 Sequence Diagram Tambah Admin



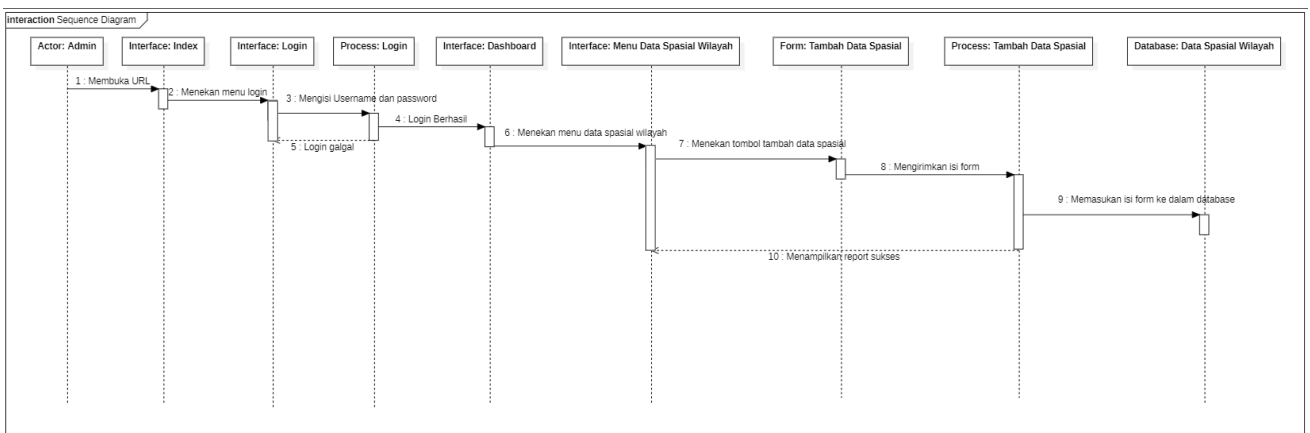
Gambar: 3.23 Sequence Diagram Ubah Admin



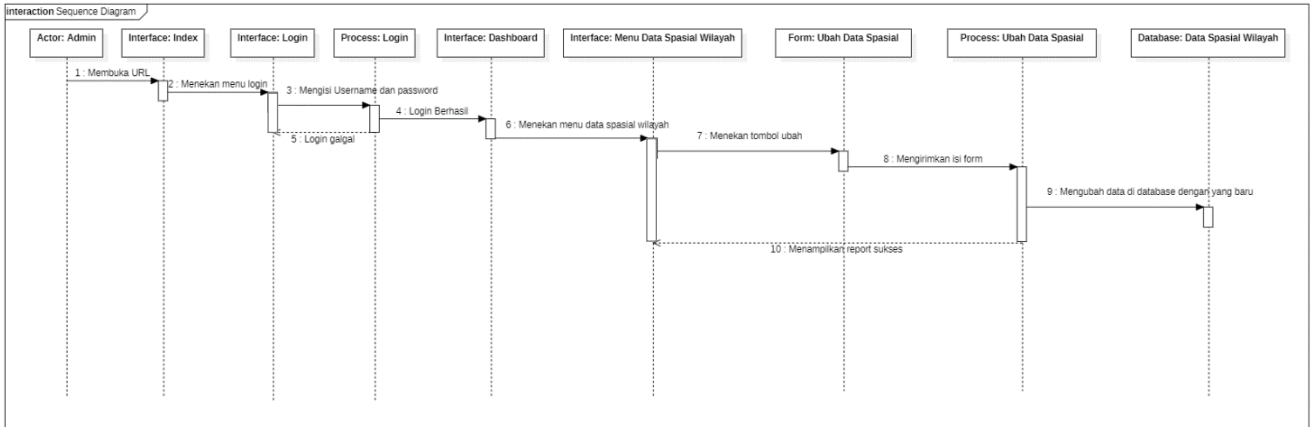
Gambar: 3.24 Sequence Diagram Hapus Admin



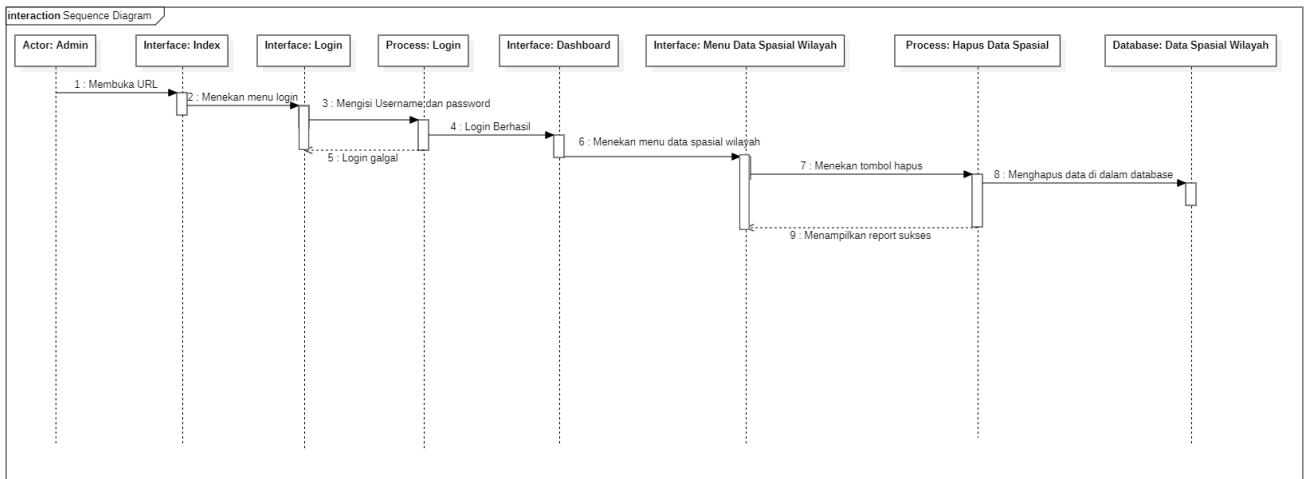
Gambar: 3.25 Sequence Diagram Lihat Admin



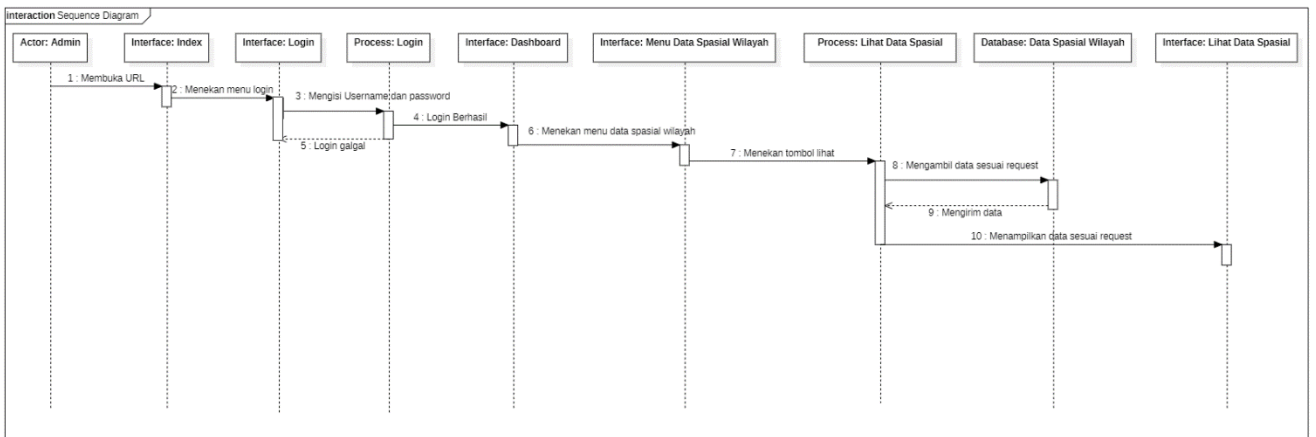
Gambar: 3.26 Sequence Diagram Tambah Data Spasial



Gambar: 3.27 Sequence Diagram Ubah Data Spasial

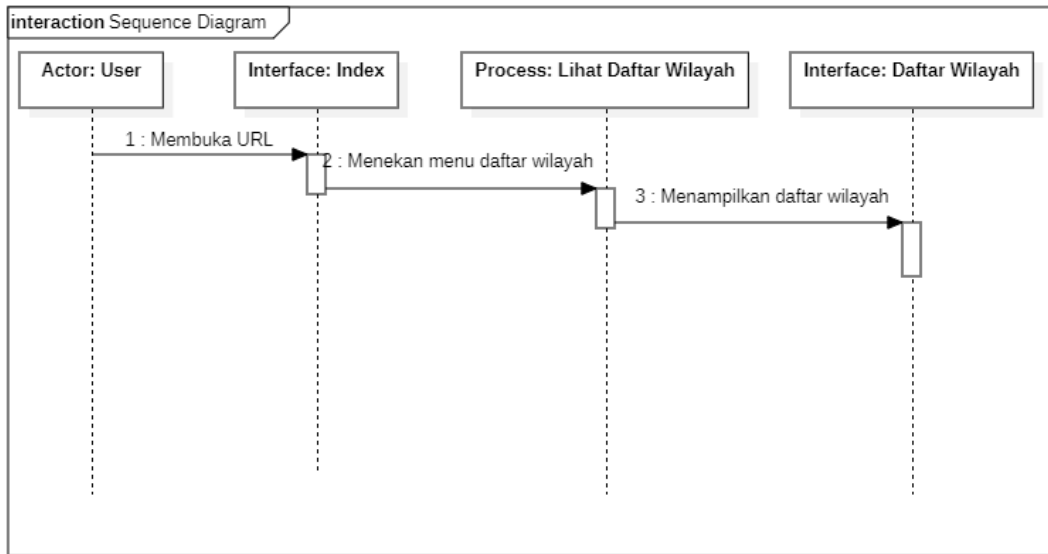


Gambar: 3.28 Sequence Diagram Hapus Data Spasial

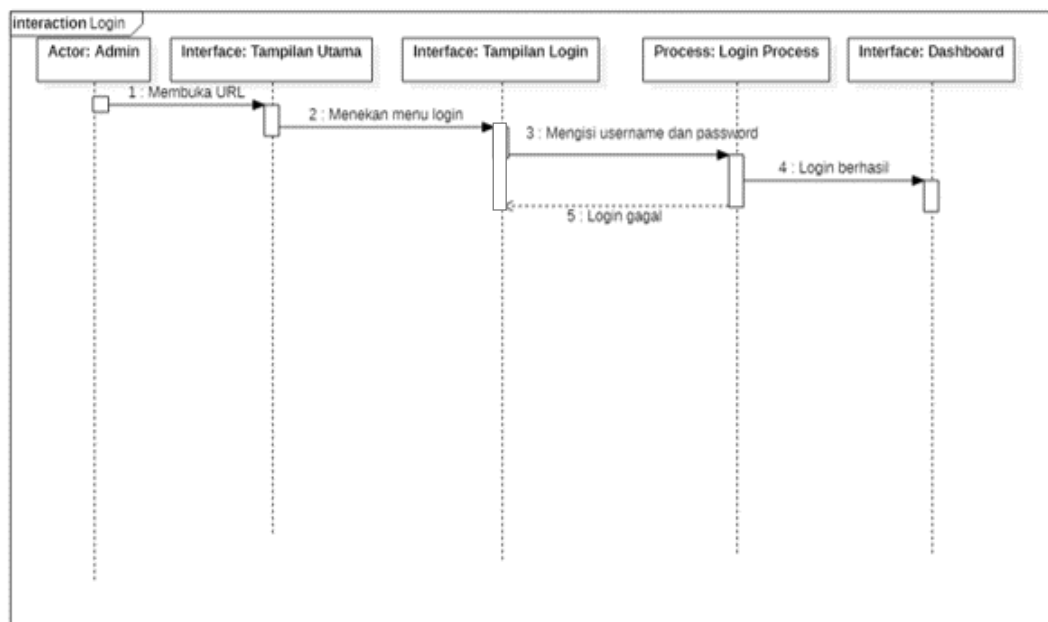


Gambar: 3.29 Sequence Diagram Lihat Data Spasial





Gambar: 3.30 Sequence Diagram User Lihat Daftar Wilayah

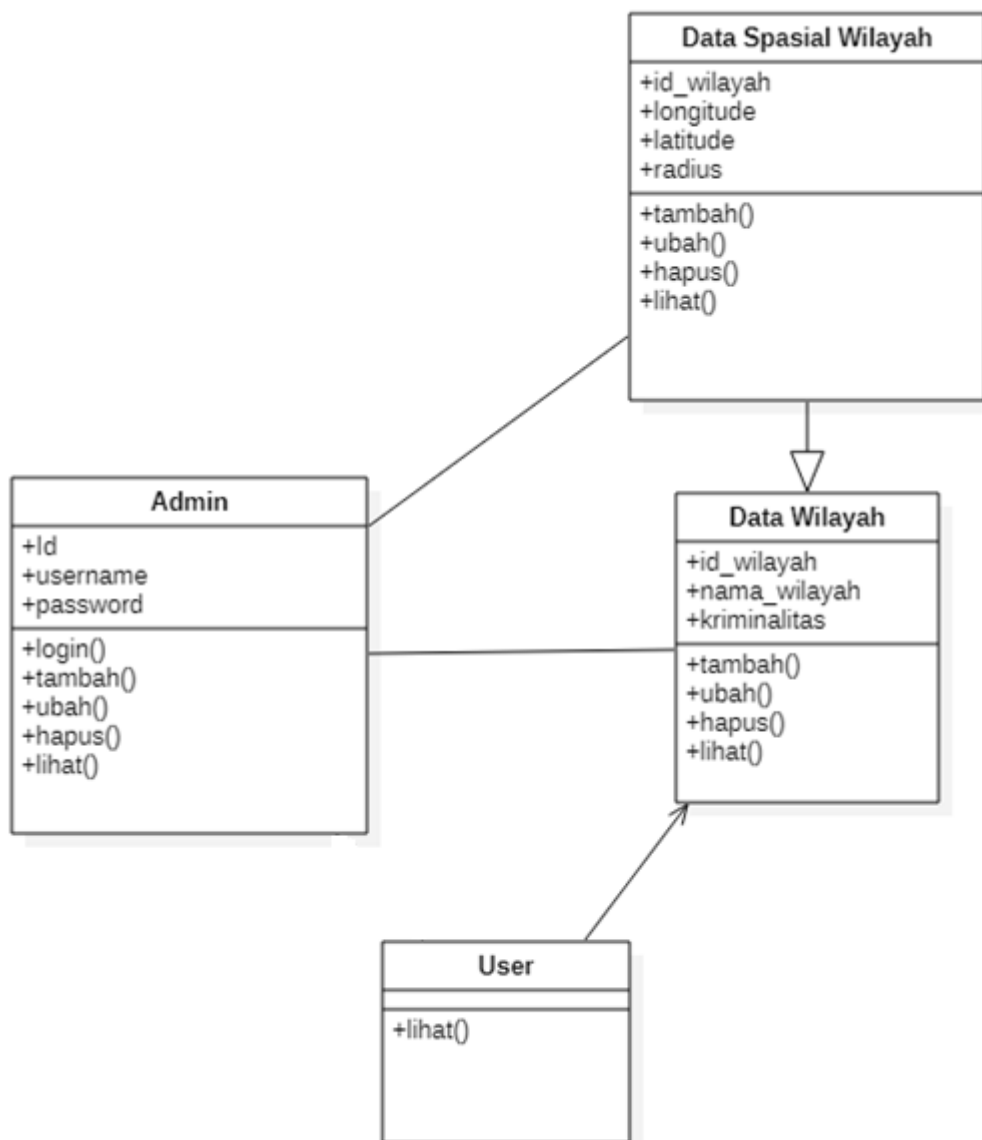


Gambar: 3.31 Sequence Diagram Login

### 3.2.1.4 Class Diagram

Diagram Class memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram Class bersifat

statis, menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan. Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.



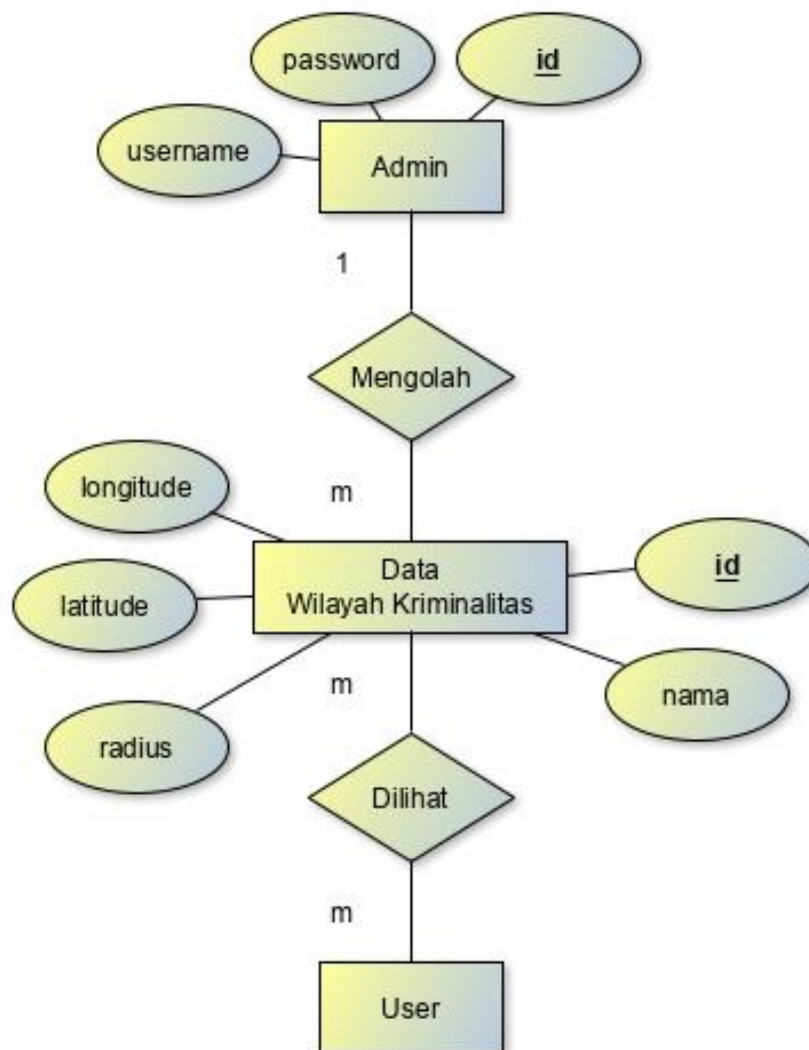
Gambar: 3.32 Perancangan Class Diagram

### **3.2.2 Perancangan Database Sistem**

Di dalam suatu organisasi yang besar, sistem database merupakan bagian penting pada sistem informasi, karena di perlukan untuk mengelola sumber informasi pada organisasi tersebut. Untuk mengelola sumber informasi tersebut yang pertama kali di lakukan adalah merancang suatu sistem basis data agar informasi yang ada pada organisasi tersebut dapat digunakan secara maksimal.

#### **3.2.2.1 Perancangan Database Dengan ERD**

**ERD** atau (*Entity Relationship Diagram*) 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. Berikut adalah perancangan database menggunakan ERD.



Gambar: 3.33 Perancangan Database Menggunakan ERD

### 3.2.2.2 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: 3.3 Data Admin

Nama Kolom	Tipe	Panjang	Null	PK / FK
id	Integer	11	No	PK
Nama	Varchar	150	Yes	
username	Varchar	150	Yes	
password	Varchar	150	Yes	
Created_at	Timestamps			
Updated_at	Timestamps			

Tabel: 3.4 Data Wilayah

Nama Kolom	Tipe	Panjang	Null	PK / FK
Id_wilayah	Integer	11	No	PK
Nama_wilayah	Varchar	150	Yes	
Kriminalitas	Varchar	150	Yes	
Jam_rawan	Varchar	150	Yes	

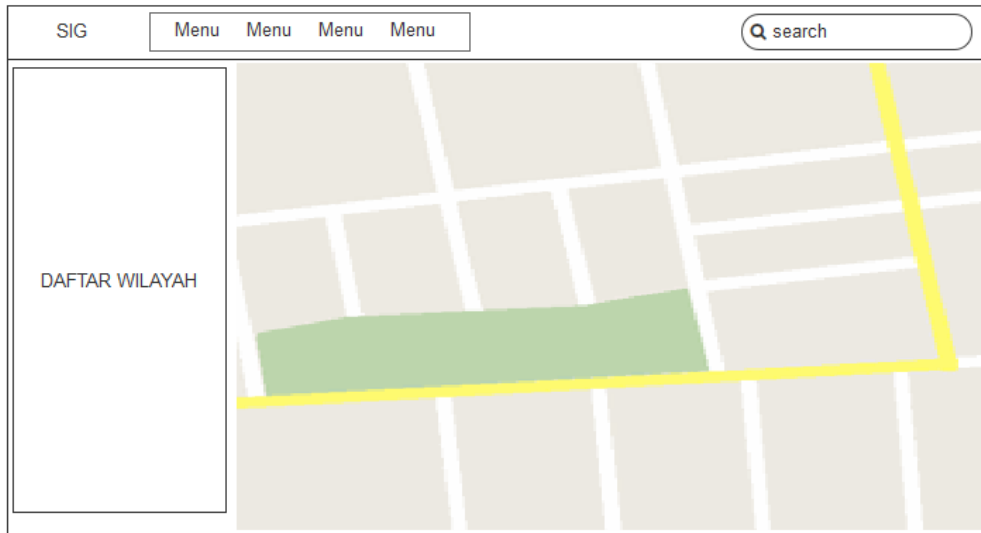
Id_spasial	Integer	11	Yes	FK
Created_at	Timestamps			
Updated_at	Timestamps			

Tabel: 3.5 Data Spasial Wilayah

Nama Kolom	Tipe	Panjang	Null	PK / FK
Id_wilayah	Integer	11	No	PK
Longitude	Varchar	150	Yes	
Latitude	Varchar	150	Yes	
Radius	Varchar	150	Yes	

### 3.2.3 Perancangan User Interface Sistem

Perancangan antarmuka disini akan menggambarkan bagaimana tampilan sistem nantinya , hal tersebut sangat dibutuhkan sebagai acuan pembuatan sistem. Selain dari pada itu dengan adanya perancangan antarmuka ini adalah untuk sebagai pengantar antara developer dengan user, sehingga menjadikan sistem yang bersifat *user friendly*. Berikut adalah perancangan terhadap antarmuka sistem.



Gambar: 3.34 Perancangan Interface Index

Gambar diatas merupakan halaman awal ketika membuka website, dijelaskan bahwa pada halaman awal akan langsung menampilkan peta lokasi, daftar wilayah, dan beberapa menu lainnya. Peta disana akan menampilkan wilayah-wilayah yang rawan terjadi kriminalitas tepatnya di kota Bandung.

Masuk

label

username

label

password

Button Button

Gambar: 3.35 Perancangan Interface Login

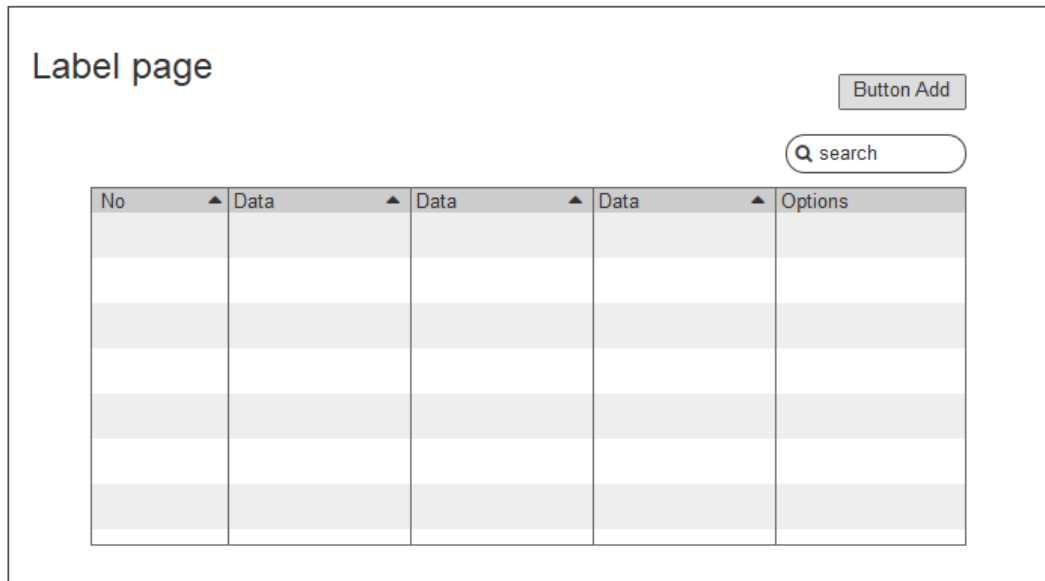
Interface login digunakan untuk mengakses *backoffice* dari sistem ini. Jadi sebelum mengakses *back-office* sistem diperlukan login terlebih dahulu dengan *username* dan juga *password* yang sudah terdaftar oleh sistem. Apabila *username* atau *password* tidak sesuai dengan yang sudah terdaftar maka sistem akan menolak akses ke *back-office* sistem.



Gambar: 3.36 Perancangan Interface Dashboard

Ketika sistem telah menerima akses ke *back-office* maka sistem akan menampilkan halaman dashboard. Halaman dashboard ini berisi beberapa analisis mengenai sistem seperti berapa jumlah banyaknya data dan data apa saja yang tersedia.





Gambar: 3.37 Perancangan Interface Datapage

Perancangan *interface datapage* merupakan tampilan *index* dari semua data yang diolah seperti data admin, spasial, dan juga data wilayah. Pada halaman ini terdapat tombol untuk melakukan tambah data dan juga tombol ubah data, lihat data dan hapus data yang berada di kolom *options*. Halaman ini akan menampilkan data secara keseluruhan yang akan disusun didalam tabel. Dan pada halaman ini tersedia *searchbox* untuk memudahkan ketika melakukan pencarian data.

Label page

Label

Label

Label

Gambar: 3.38 Perancangan Interface Tambah Data

Halaman tambah data merupakan halaman untuk mengisi kelengkapan data untuk nantinya dimasukan ke dalam *database*. Halaman tambah data ini hanya tersedia *form* untuk mengisi data serta tombol *submit* ketika data sudah selesai diisi.

Label page

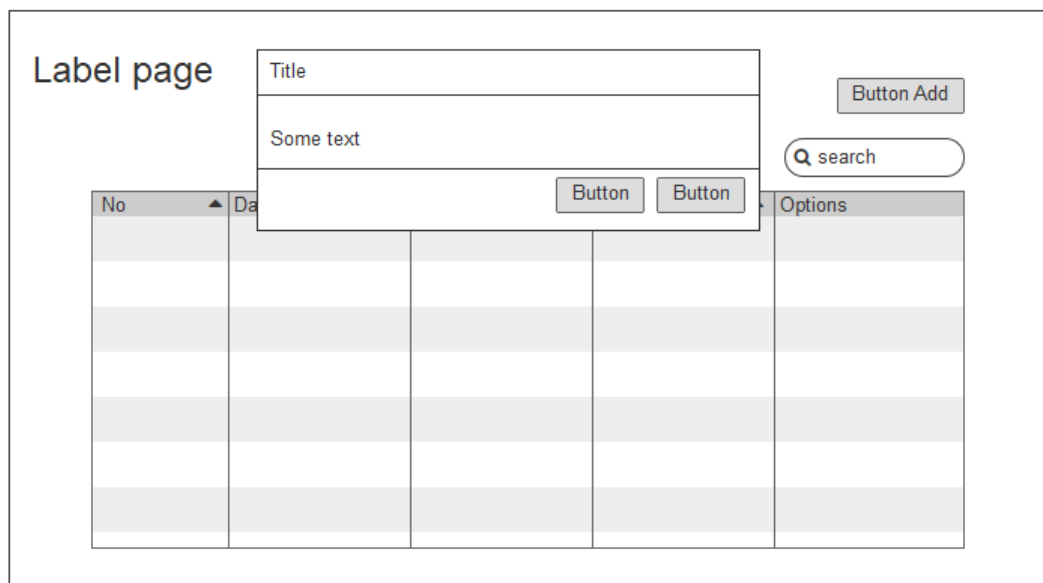
Label

Label

Label

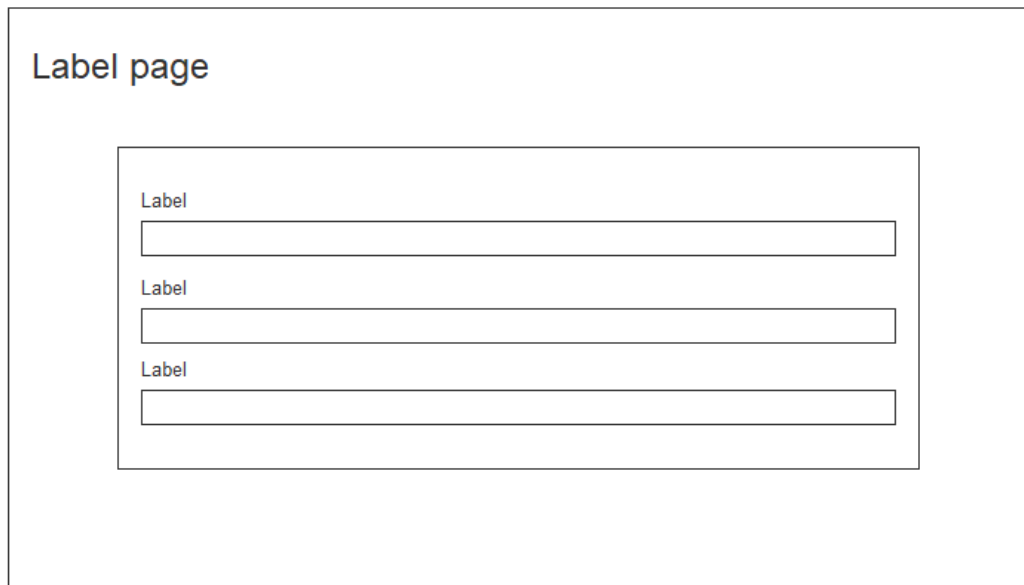
Gambar: 3.39 Perancangan Interface Ubah Data

Perancangan *interface* ubah data tidak jauh berbeda dengan *interface* tambah data. Pada halaman ini hanya berisi form untuk mengisi kelengkapan data atau mengubah data yang sudah ada serta tombol *submit* apabila telah selesai dan tombol *cancel* apabila ingin membatalkan pengubahan data.



Gambar: 3.40 Perancangan Interface Hapus Data

Untuk melakukan penghapusan terhadap data, akan diberikan validasi berupa pilihan sebelum melakukan penghapusan data. Apabila ingin menghapus data hanya perlu menekan tombol Hapus, sedangkan apabila ingin membatalkan penghapusan tekan tombol Batalkan.



Gambar: 3.41 Perancangan Interface Lihat Data

Interface lihat data hanya akan menampilkan data sesuai pilihan. Halaman ini tidak dapat digunakan untuk mengubah ataupun menambahkan data. Dan data yang ditampilkan pada halaman ini hanya satu data sesuai pilihan.

## **BAB IV**

### **IMPLEMENTATION**

#### 4.1 Implementation ( Implementasi )

Implementasi yang dilakukan oleh penulis adalah merealisasikan kelas-kelas yang dibentuk pada tahap perancangan ke dalam suatu kode bahasa pemrograman, sehingga dapat menghasilkan suatu perangkat lunak yang nyata dan dapat digunakan. Tahap implementasi dibagi menjadi 3 bagian yang diantaranya adalah implementasi pendukung sistem, implementasi database, dan implementasi *user interface*.

##### 4.1.1 Implementasi Pendukung Sistem

Pada implementasi pendukung sistem dibutuhkan beberapa perangkat *hardware* dan *software* untuk mengoperasikan sistem ini. Berikut adalah perangkat pendukung sistem yang telah penulis klasifikan sebagai berikut:

##### 1. Hardware

- PC dengan spesifikasi minimal:
  - Model Processor Core 2 Duo
  - Kecepatan Processor 2.26GHz
  - Model GPU ntel GMA 4500MHD
  - Ram 500mb
  - 20GB HDD








- atau Laptop
- Modem

## 2. Software








- Windows 10
- Google Chrome
- Xampp

### 4.1.2 Implementasi Database







Perancangan *database* pada tahap sebelumnya diimplementasikan ke dalam *database management system*. Berikut adalah implementasi *database* sistem.

Name	Type	NULL	Default	Extras	C..
<b>Keys (1)</b>					
 <b>Primary Key</b> Id				unique	
<b>Fields (6)</b>					
 <b>Id</b>	int(11)	No	<auto_increment>		
 nama	varchar(150)	Yes	<NULL>		
 username	varchar(150)	Yes	<NULL>		
 password	varchar(150)	Yes	<NULL>		
 created_at	timestamp	Yes	<NULL>		
 updated_at	timestamp	Yes	<NULL>		

Gambar: 4.1 Implementasi Database Data Admin

Name	Type	NULL	Default	Extras	Comments
Keys (2)					
 <b>Primary Key</b>	Id			unique	
 Foreign Key	id_wilayah			unique	
Fields (5)					
 <b>Id</b>	int(11)	No	<auto_increment>		
 id_wilayah	int(11)	Yes	<NULL>		
 longitude	varchar(150)	Yes	<NULL>		
 latitude	varchar(150)	Yes	<NULL>		
 radius	varchar(150)	Yes	<NULL>		

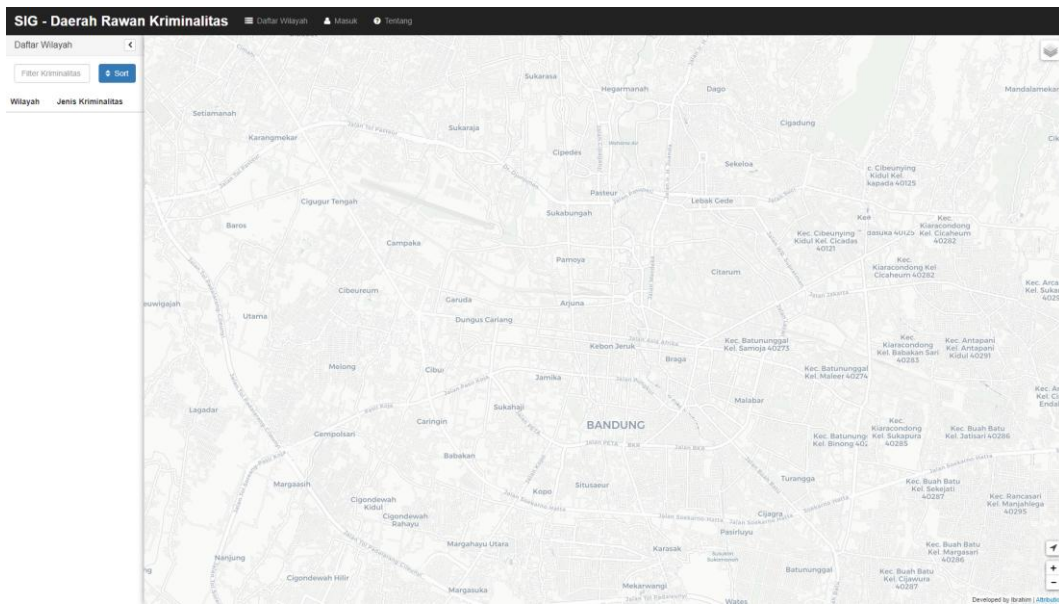
Gambar: 4.2 Implementasi Database Data Spasial Wilayah

Name	Type	NULL	Default	Extras	Comments
Keys (1)					
 <b>Primary Key</b>	Id_wilayah			unique	
Fields (5)					
 <b>Id_wilayah</b>	int(11)	No	<auto_increment>		
 nama_wilayah	varchar(150)	Yes	<NULL>		
 kriminalitas	varchar(150)	Yes	<NULL>		
 created_at	timestamp	Yes	<NULL>		
 updated_at	timestamp	Yes	<NULL>		

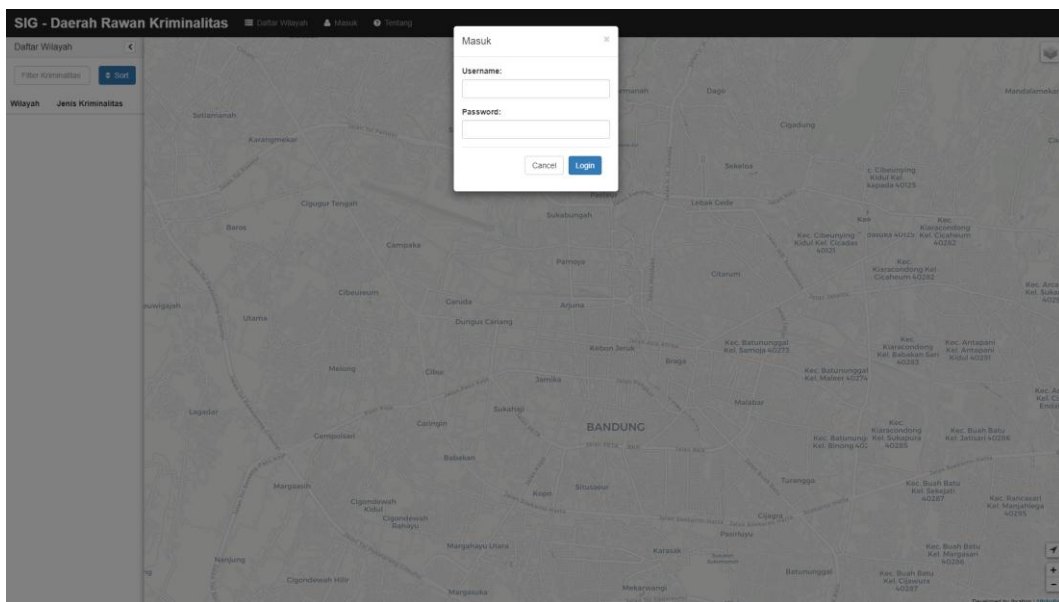
Gambar: 4.3 Implementasi Database Data Wilayah

#### 4.1.3 Implementasi User Interface

Implementasi *user interface* merupakan tampilan dari sistem yang diperoleh dengan cara melakukan *screenshot* bagaimana tampilan sistem setelah diimplementasikan kedalam Bahasa pemrograman. Implementasi *user interface* sistem sendiri berpacu dari perancangan *user interface* pada tahap sebelumnya, berikut adalah implementasi *user interface* sistem.

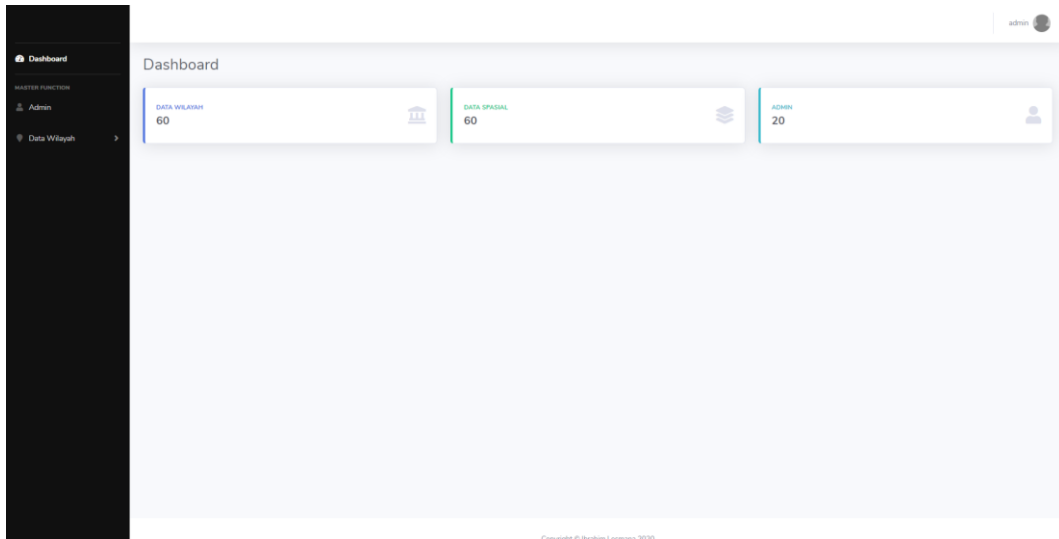


*Gambar: 4.4 Implementasi Interface Index*

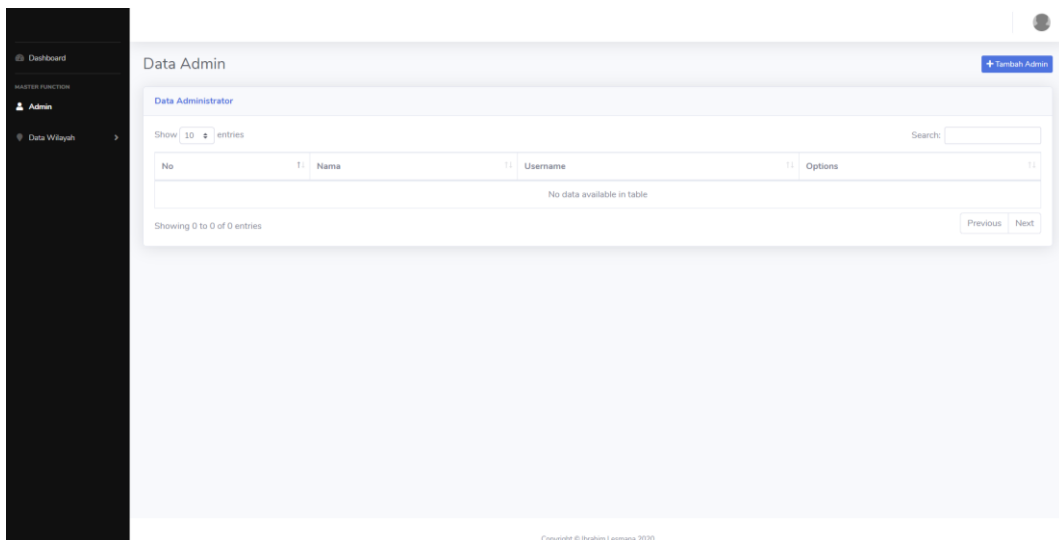


*Gambar: 4.5 Implementasi Interface Login*

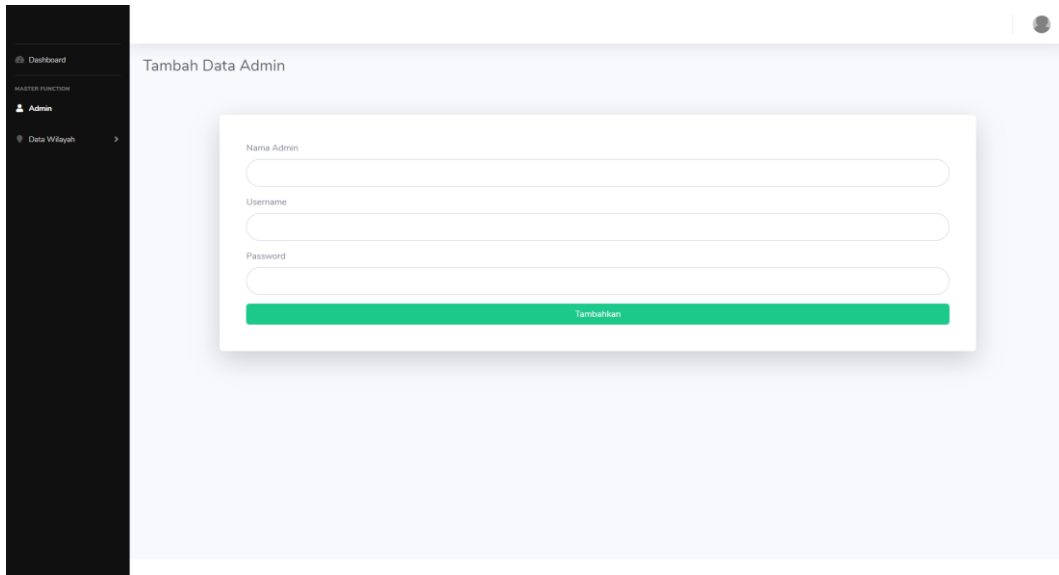




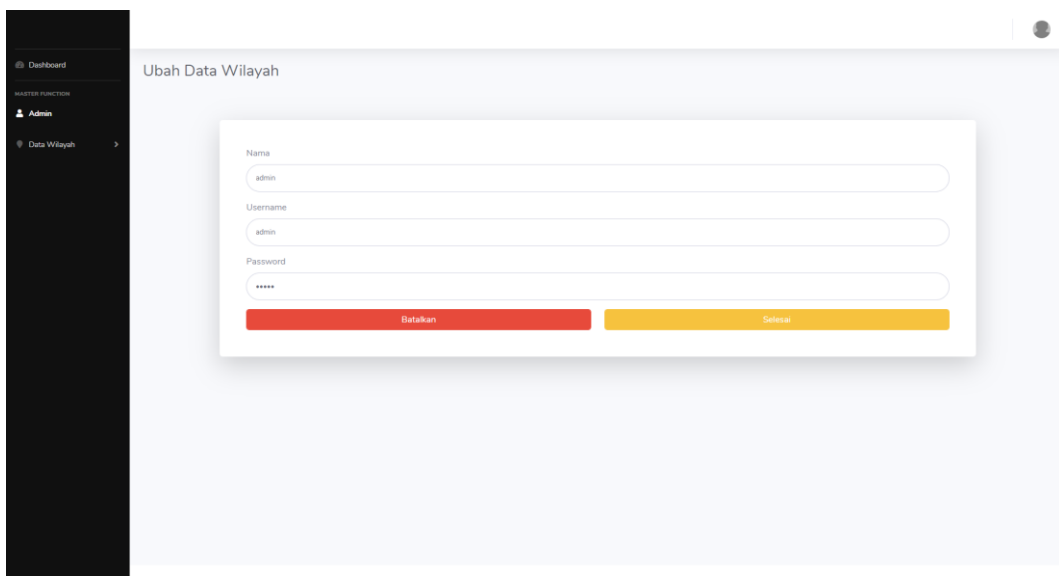
*Gambar: 4.6 Implementasi Interface Dashboard*



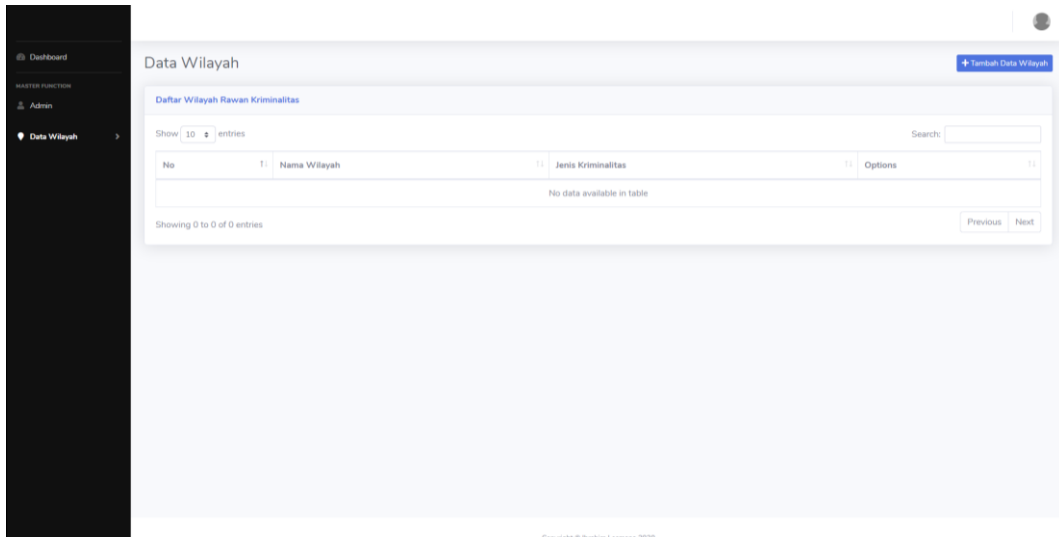
*Gambar: 4.7 Implementasi Interface Data Admin*



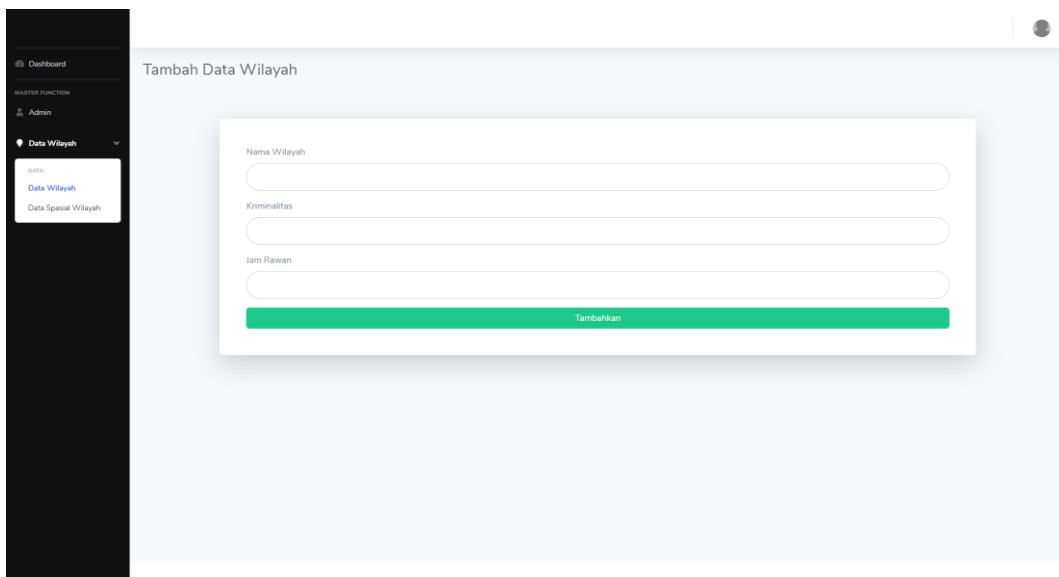
*Gambar: 4.8 Implementasi Interface Tambah Data Admin*



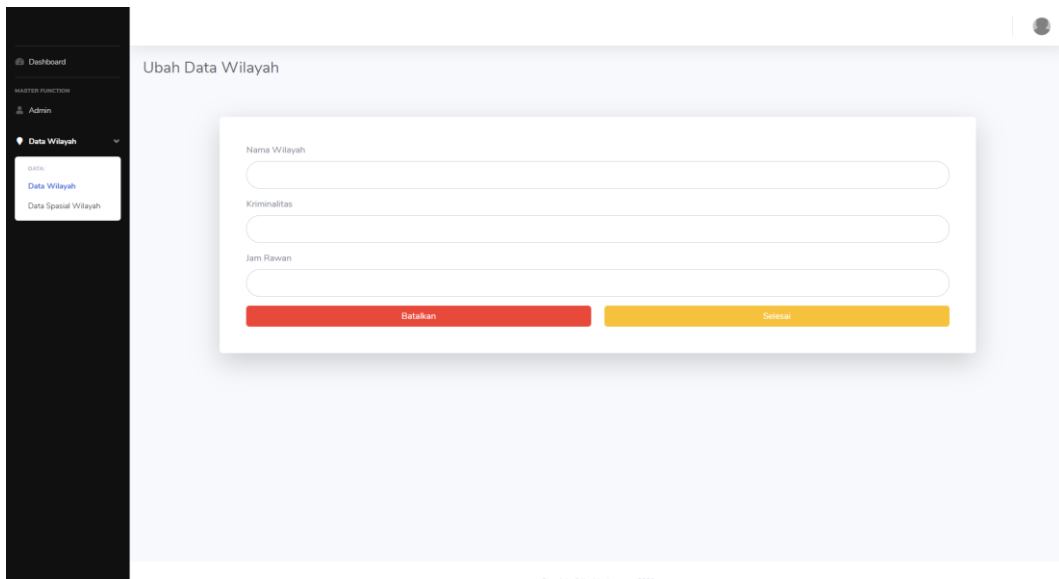
*Gambar: 4.9 Implementasi Interface Ubah Data Admin*



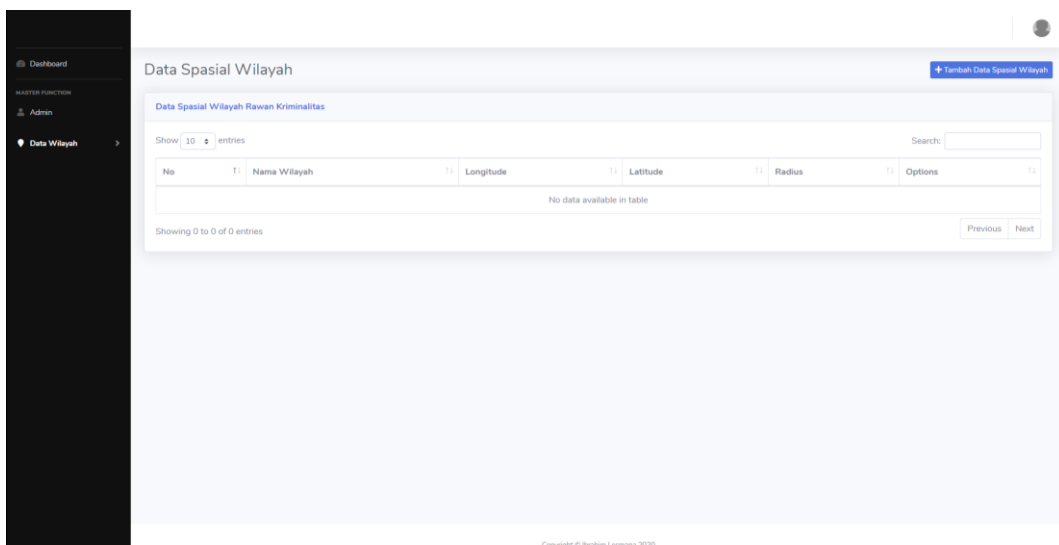
*Gambar: 4.10 Implementasi Interface Data Wilayah*



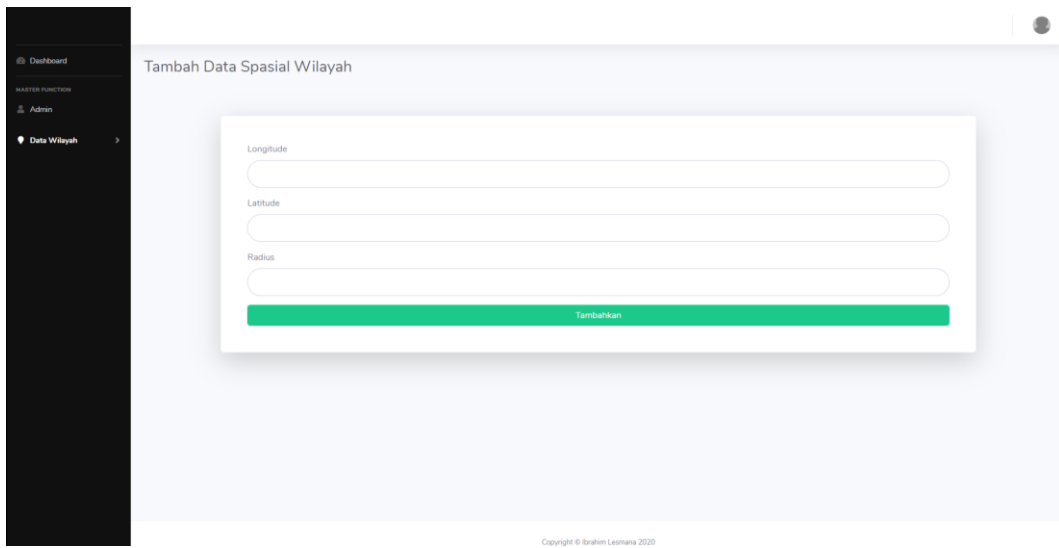
*Gambar: 4.11 Implementasi Interface Tambah Data Wilayah*



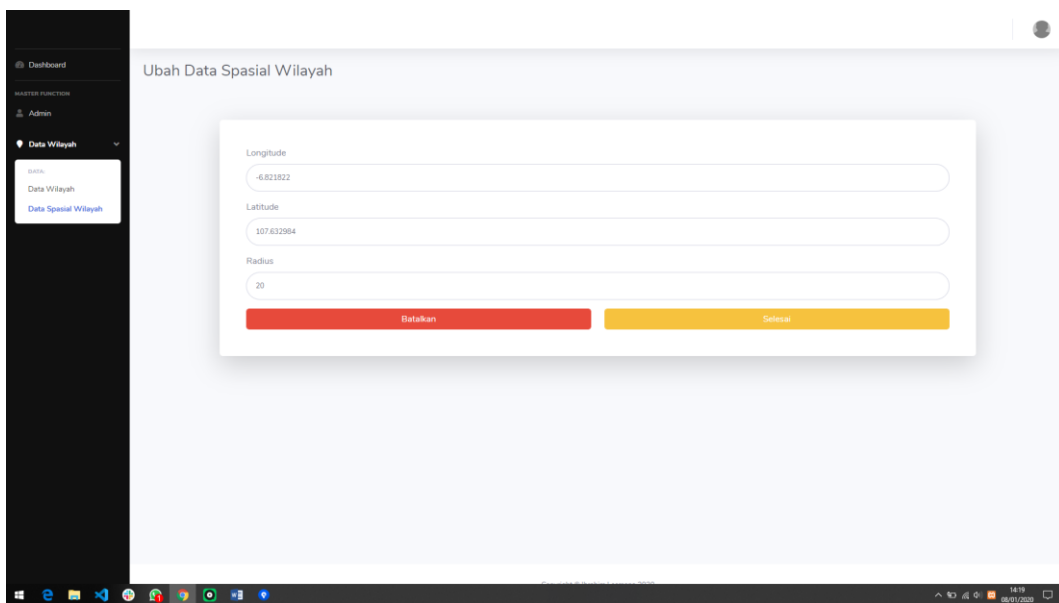
Gambar: 4.12 Implementasi Interface Ubah Data Wilayah



Gambar: 4.13 Implementasi Interface Data Spasial Wilayah



*Gambar: 4.14 Implementasi Interface Tambah Data Spasial Wilayah*



*Gambar: 4.15 Implementasi Interface Ubah Data Spasial Wilayah*

#### 4.1.4 Pengujian Sistem

Pengujian sistem adalah pengujian yang dilakukan terhadap keseluruhan sistem (secara lengkap) dan sistem yang telah terintegrasi untuk mengevaluasi apakah sistem yang dibuat telah sesuai dengan kebutuhan pengguna. Pengujian sistem merupakan bagian dari *black-box* testing, yang tidak membutuhkan pengetahuan tentang kode dan logika pemrograman. Berikut adalah pengujian terhadap sistem menggunakan *black-box*.

Tabel: 4.1 Pengujian Menu Data Admin

Fungsi	Tes yang dilakukan	Hasil yang diharapkan	Valid / Invalid
Tambah Data	Mengisi seluruh data pada form dengan benar	Menampilkan report sukses	Valid
	Tidak mengisi salah satu data pada form	Menampilkan validasi data	Valid
	Mengisi angka pada data nama	Menampilkan validasi hanya huruf saja	Valid
Ubah Data	Mengubah beberapa data pada form	Menampilkan report sukses	Valid
	Menambahkan angka pada data nama	Menampilkan validasi hanya huruf saja	Valid
	Menekan tombol selesai tanpa mengubah atau menambahkan data	Menampilkan report sukses	Valid

Hapus Data	Menekan tombol Ya	Menampilkan report sukses	Valid
	Menekan tombol Tidak	Menutup validasi hapus	Valid
Lihat Data	Menekan tombol lihat	Memunculkan data sesuai pilihan	Valid

Tabel: 4.2 Pengujian Menu Data Spasial Wilayah

Fungsi	Tes yang dilakukan	Hasil yang diharapkan	Valid / Invalid
Tambah Data	Mengisi seluruh data pada form dengan benar	Menampilkan report sukses	Valid
	Tidak mengisi salah satu data pada form	Menampilkan validasi data	Valid
	Mengisi huruf pada data longitude atau latitude	Menampilkan validasi tidak boleh huruf	Valid
Ubah Data	Mengubah beberapa data pada form	Menampilkan report sukses	Valid
	Menambahkan huruf pada data longitude atau latitude	Menampilkan validasi tidak boleh huruf	Valid
	Menekan tombol selesai tanpa mengubah atau menambahkan data	Menampilkan report sukses	Valid
Hapus Data	Menekan tombol Ya	Menampilkan report	Valid

		sukses	
	Menekan tombol Tidak	Menutup validasi hapus	Valid
Lihat Data	Menekan tombol lihat	Memunculkan data sesuai pilihan	Valid

Tabel: 4.3 Pengujian Menu Daftar Wilayah

Fungsi	Tes yang dilakukan	Hasil yang diharapkan	Valid / Invalid
Tambah Data	Mengisi seluruh data pada form dengan benar	Menampilkan report sukses	Valid
	Tidak mengisi salah satu data pada form	Menampilkan validasi data	Valid
Ubah Data	Mengubah beberapa data pada form	Menampilkan report sukses	Valid
	Menekan tombol selesai tanpa mengubah atau menambahkan data	Menampilkan report sukses	Valid
Hapus Data	Menekan tombol Ya	Menampilkan report sukses	Valid



	Menekan tombol Tidak	Menutup validasi hapus	Valid
Lihat Data	Menekan tombol lihat	Memunculkan data sesuai pilihan	Valid

Tabel: 4.4 Pengujian User

Fungsi	Tes yang dilakukan	Hasil yang diharapkan	Valid / Invalid
Lihat Daftar Admin	Menekan menu daftar admin	Menampilkan data admin	Valid
Lihat Daftar Wilayah	Menekan menu daftar wilayah	Menampilkan data wilayah	Valid

Tabel: 4.5 Pengujian Login dan Logout Admin

Fungsi	Tes yang dilakukan	Hasil yang diharapkan	Valid / Invalid
Login	Memasukan <i>username</i> benar dan <i>password</i> benar	Mengarahkan tampilan ke halaman dashboard	Valid
	Tidak memasukan <i>username</i> dan <i>password</i>	Menampilkan notifikasi field username tidak boleh kosong	Valid
	Memasukan <i>username</i> tapi tidak memasukan	Menampilkan notifikasi field password tidak	Valid

	<i>password</i>	boleh kosong	
	Memasukan <i>password</i> tapi tidak memasukan <i>username</i>	Menampilkan notifikasi field username tidak boleh kosong	Valid
	Memasukan username benar tetapi password salah	Menampilkan notifikasi username atau password salah pada halaman login	Valid
	Memasukan username salah tetapi password benar	Menampilkan notifikasi username atau password salah pada halaman login	Valid
Logout	Menekan tombol logout	Menampilkan halaman index user	Valid

Tabel: 4.6 Pengujian Role Page Admin dan User

Fungsi	Tes yang dilakukan	Hasil yang diharapkan	Valid / Invalid
Admin	Mengakses halaman dashboard	Menampilkan dashboard	Valid
	Mengakses halaman data wilayah	Menampilkan data wilayah	Valid
	Mengakses halaman data admin	Menampilkan data admin	Valid
	Mengakses halaman data spasial wilayah	Menampilkan data spasial	Valid
	Mengakses index user	Menampilkan index user	Valid
User	Mengakses halaman	Menampilkan halaman error 403 atau tidak bisa	Valid

	dashboard	mengakses halaman ini	
	Mengakses halaman data wilayah	Menampilkan halaman error 403 atau tidak bisa mengakses halaman ini	Valid
	Mengakses halaman data admin	Menampilkan data admin	Valid
	Mengakses halaman data spasial wilayah	Menampilkan halaman error 403 atau tidak bisa mengakses halaman ini	Valid
	Mengakses index user	Menampilkan index user	Valid

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Setelah dilakukan penelitian secara menyuruh penulis dapat menyimpulkan beberapa kesimpulan diantaranya adalah sebagai berikut:

1. Sistem ini dapat memetakan daerah mana saja yang rawan terjadi kriminalitas.
2. Sistem ini dapat menampilkan daerah yang rawan terjadi kriminalitas dalam bentuk radius.
3. Sistem ini dapat meniginformasikan daerah rawan terjadi kriminalitas kepada masyarakat.

#### **5.2 Saran**

Adapun saran dari penulis untuk perkembangan sistem kedepannya adalah adanya kerja sama dengan jajaran Kepolisian agar data yang dikelola akan lebih menjamin terhubung data saat ini hanya di dapat dari jajaran Polrestabes melalui media internet.

## DAFTAR PUSTAKA

- Aditya, 2019, <https://www.codepolitan.com/mengenal-uml-diagram-use-case>.
- Aditya, 2019, <https://www.codepolitan.com/mengenal-uml-contoh-uml-diagram-model-activity-diagram>.
- Aditya, 2019, <https://www.codepolitan.com/belajar-uml-sequence-diagram-57fdb1a5ba777-17044>
- Brady, M., & Loonam, J. (2010). *Exploring the use of entity-relationship diagramming as a technique to support grounded theory inquiry*.
- Dr. Ir. Harijono Djojodihardjo, 2014, *Pengantar Sistem Komputer*, Erlangga, Bandung.
- Gelinas, J.U., Dull, Richard B., Wheeler, Patrick R. 2012. *Accounting Information Systems. South Western: Cengage Learning*.
- Hermawan, 2005, *Membangun decision support system*, Jakarta.
- Irmayani, 2019, <https://www.nesabamedia.com/pengertian-flowchart/>.
- Marshall B. Romney dan Paul John Steinbart. 2014 *Sistem Informasi Akuntansi: Accounting Information Systems (Edisi 13)*, Prentice Hall.
- Muiz, 2007, *Pengenalan Sistem Basisdata Jilid 2. Indeks*. Jakarta.
- Pablo Pastor, "MVC for Noobs", <https://code.tutsplus.com/tutorials/mvc-for-noobs--net-10488>, diakses pada tanggal 27 Oktober 2019
- Rahmad, 2016, <https://www.geodose.com/2016/06/mengenal-google-maps-api-3.html>.
- Seidl, M., Scholz, M., Huemer, C., & Kappel, G. 2015. *UML@ classroom: An introduction to object-oriented modeling*. Springer.
- Soesilo, R. 1988, *Kitab Undang-Undang Acara Pidana*, Politeia, Bogor.
- Sutabri Tata, 2016, *Sistem Informasi Manajemen*, Andi Offset, Yogyakarta

# LAMPIRAN A

## SOURCE CODE PROGRAM

### 1. View

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="initial-scale=1,user-
scalable=no,maximum-scale=1,width=device-width">
    <meta name="mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="theme-color" content="#000000">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>SIG - Daerah Rawan Kriminalitas</title>

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bo
otstrap.min.css">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.4.0/css/font-awesome.min.css">
```

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/lea
flet.css">

<link rel="stylesheet"
href="https://api.tiles.mapbox.com/mapbox.js/plugins/leaflet
-markercluster/v0.4.0/MarkerCluster.css">

<link rel="stylesheet"
href="https://api.tiles.mapbox.com/mapbox.js/plugins/leaflet
-markercluster/v0.4.0/MarkerCluster.Default.css">

<link rel="stylesheet"
href="https://api.tiles.mapbox.com/mapbox.js/plugins/leaflet
-locatecontrol/v0.43.0/L.Control.Locate.css">

<link rel="stylesheet" href="{!!
asset('frontend/assets/leaflet-
groupedlayercontrol/leaflet.groupedlayercontrol.css') !!}">

<link rel="stylesheet" href="{!!
asset('frontend/assets/css/app.css') !!}">

</head>

<body>

<div class="navbar navbar-inverse navbar-fixed-top"
role="navigation">

<div class="container-fluid">

<div class="navbar-header">

<div class="navbar-icon-container">
```

```

        <a href="#" class="navbar-icon pull-right
visible-xs" id="nav-btn"><i class="fa fa-bars fa-lg
white"></i></a>

        <a href="#" class="navbar-icon pull-right
visible-xs" id="sidebar-toggle-btn"><i class="fa fa-search
fa-lg white"></i></a>

    </div>

    <a class="navbar-brand" href="#">SIG - Daerah
Rawan Kriminalitas</a>

</div>

<div class="navbar-collapse collapse">
    <form class="navbar-form navbar-right"
role="search">

        <div class="form-group has-feedback">
            <input id="searchbox" type="text"
placeholder="Search" class="form-control">

            <span id="searchicon" class="fa fa-search
form-control-feedback"></span>

        </div>

    </form>

    <ul class="nav navbar-nav">

        <li class="hidden-xs"><a href="#" data-
toggle="collapse" data-target=".navbar-collapse.in"
id="list-btn"><i class="fa fa-list
white"></i>&nbsp;&nbsp;&nbsp;Daftar Wilayah</a></li>

        <li><a href="#" data-toggle="collapse" data-
target=".navbar-collapse.in" id="legend-btn"><i class="fa
fa-picture-o white"></i>&nbsp;&nbsp;&nbsp;Legenda</a></li>

```



```
        <li><a href="#" data-toggle="collapse" data-
target=".navbar-collapse.in" id="login-btn"><i class="fa fa-
user white"></i>&nbsp;&nbsp;&nbsp;Masuk</a></li>
```

```
        <li><a href="#" data-toggle="collapse" data-
target=".navbar-collapse.in" id="about-btn"><i class="fa fa-
question-circle white"></i>&nbsp;&nbsp;&nbsp;Tentang</a></li>
```

```
    </ul>
```

```
</div><!--/.navbar-collapse -->
```

```
</div>
```

```
</div>
```

```
<div id="container">
```

```
  <div id="sidebar">
```

```
    <div class="sidebar-wrapper">
```

```
      <div class="panel panel-default" id="features">
```

```
        <div class="panel-heading">
```

```
          <h3 class="panel-title">Daftar Wilayah
```

```
          <button type="button" class="btn btn-xs btn-
default pull-right" id="sidebar-hide-btn"><i class="fa fa-
chevron-left"></i></button></h3>
```

```
        </div>
```

```
        <div class="panel-body">
```

```
          <div class="row">
```

```
            <div class="col-xs-8 col-md-8">
```

```
              <input type="text" class="form-control
search" placeholder="Filter" />
```

```
            </div>
```

```
            <div class="col-xs-4 col-md-4">
```

```
        <button type="button" class="btn btn-
primary pull-right sort" data-sort="feature-name" id="sort-
btn"><i class="fa fa-sort"></i>&nbsp;&nbsp;&nbsp;Sort</button>
    </div>
</div>
</div>
<div class="sidebar-table">
    <table class="table table-hover" id="feature-
list">
        <thead class="hidden">
            <tr>
                <th>Icon</th>
            <tr>
            <tr>
                <th>Name</th>
            <tr>
            <tr>
                <th>Chevron</th>
            <tr>
        </thead>
        <tbody class="list"></tbody>
    </table>
</div>
</div>
</div>
</div>
<div id="map"></div>
</div>
```

```

<div id="loading">
  <div class="loading-indicator">
    <div class="progress progress-striped active">
      <div class="progress-bar progress-bar-info
progress-bar-full"></div>
    </div>
  </div>
</div>

<div class="modal fade" id="aboutModal" tabindex="-1"
role="dialog">
  <div class="modal-dialog modal-lg">
    <div class="modal-content">
      <div class="modal-header">
        <button class="close" type="button" data-
dismiss="modal" aria-hidden="true">&times;</button>
        <h4 class="modal-title">Selamat Datang !</h4>
      </div>
      <div class="modal-body">
        <ul class="nav nav-tabs nav-justified"
id="aboutTabs">
          <li class="active"><a href="#about" data-
toggle="tab"><i class="fa fa-question-
circle"></i>&nbsp;&nbsp;&nbsp;Deskripsi Project</a></li>
          <li><a href="#contact" data-toggle="tab"><i
class="fa fa-envelope"></i>&nbsp;&nbsp;&nbsp;Contact us</a></li>
        </ul>
        <div class="tab-content" id="aboutTabsContent">

```

```
<div class="tab-pane fade active in"
id="about">
    <p>SIG - Daerah Rawan Kriminalitas adalah
sebuah platform yang akan menampilkan daerah-daerah yang
rawan terjadi kriminalitas</p>
    <div class="panel panel-primary">
        <div class="panel-heading">Tujuan </div>
        <ul class="list-group">
            <li class="list-group-item">Memberikan
informasi penting kepada masyarakat terkait jenis
kriminalitas dan lokasi rawan terjadi kriminalitas</li>
            <li class="list-group-item">Mengurangi
terjadinya tindak kriminalitas</li>
            <li class="list-group-item">Memberi
peringatan kepada masyarakat agar lebih hati-hati jika
berada di dalam kawasan rawan terjadi kriminalitas</li>
        </ul>
    </div>
</div>
<div id="disclaimer" class="tab-pane fade
text-danger">
    <p>The data provided on this site is for
informational and planning purposes only.</p>
    <p>Absolutely no accuracy or completeness
guarantee is implied or intended. All information on this
map is subject to such variations and corrections as might
result from a complete title search and/or accurate field
survey.</p>
```

```
</div>
<div class="tab-pane fade" id="contact">
  <form id="contact-form">
    <div class="well well-sm">
      <div class="row">
        <div class="col-md-4">
          <div class="form-group">
            <label for="first-name">First
Name:</label>
            <input type="text" class="form-
control" id="first-name">
          </div>
          <div class="form-group">
            <label for="last-name">Last
Name:</label>
            <input type="text" class="form-
control" id="last-email">
          </div>
          <div class="form-group">
            <label for="email">Email:</label>
            <input type="text" class="form-
control" id="email">
          </div>
        </div>
      </div>
      <div class="col-md-8">
        <label
for="message">Message:</label>
```

```
        <textarea class="form-control"
rows="8" id="message"></textarea>
    </div>
    <div class="col-md-12">
        <p>
            <button type="submit" class="btn
btn-primary pull-right" data-dismiss="modal">Submit</button>
        </p>
    </div>
</div>
</div>
</form>
</div>
<div class="tab-pane fade" id="boroughs-tab">
    <p>Borough data courtesy of <a
href="http://www.nyc.gov/html/dcp/pdf/bytes/nybbwi_metadata.
pdf" target="_blank">New York City Department of City
Planning</a></p>
</div>
<div class="tab-pane fade" id="subway-lines-
tab">
    <p><a
href="http://spatialityblog.com/2010/07/08/mta-gis-data-
update/#datalinks" target="_blank">MTA Subway data</a>
courtesy of the <a
href="http://www.urbanresearch.org/about/cur-
components/cuny-mapping-service" target="_blank">CUNY
Mapping Service at the Center for Urban Research</a></p>
```

</div>

<div class="tab-pane fade" id="theaters-tab">

<p>Theater data courtesy of <a href="https://data.cityofnewyork.us/Recreation/Theaters/kdu2-865w" target="\_blank">NYC Department of Information & Telecommunications (DoITT)</a></p>

</div>

<div class="tab-pane fade" id="museums-tab">

<p>Museum data courtesy of <a href="https://data.cityofnewyork.us/Recreation/Museums-and-Galleries/sat5-adpb" target="\_blank">NYC Department of Information & Telecommunications (DoITT)</a></p>

</div>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div><!-- /.modal-content -->

</div><!-- /.modal-dialog -->

</div><!-- /.modal -->

<div class="modal fade" id="legendModal" tabindex="-1" role="dialog">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

```

        <button type="button" class="close" data-
dismiss="modal" aria-hidden="true">&times;</button>
        <h4 class="modal-title">Map Legend</h4>
    </div>
    <div class="modal-body">
        <p>Map Legend goes here...</p>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-default"
data-dismiss="modal">Close</button>
    </div>
</div><!-- /.modal-content -->
</div><!-- /.modal-dialog -->
</div><!-- /.modal -->

<div class="modal fade" id="loginModal" tabindex="-1"
role="dialog">
    <div class="modal-dialog modal-sm">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-
dismiss="modal" aria-hidden="true">&times;</button>
                <h4 class="modal-title">Masuk</h4>
            </div>
            <div class="modal-body">
                <form id="my-form" method="POST" action="{{
action('LoginController@cekLogin') }}">
                    {{ csrf_field() }}

```



```

<fieldset>
  <div class="form-group">
    @if (session('status') == 'salah')
      <p class="text-danger">Maaf,
        Username atau Password anda
salah!</p>
    @endif

    <label for="name">Username:</label>
    <input type="text" class="form-control"
name="username" id="username">
  </div>
  <div class="form-group">
    <label for="email">Password:</label>
    <input type="password" class="form-
control" name="password" id="password">
  </div>
</fieldset>

<div class="modal-footer">
  <button type="button" class="btn btn-
default" data-dismiss="modal">Cancel</button>
  <button type="submit" form="my-form"
class="btn btn-primary">Login</button>
</div>
</form>
</div>
</div><!-- /.modal-content -->
</div><!-- /.modal-dialog -->

```

```
</div><!-- /.modal -->

<div class="modal fade" id="featureModal" tabindex="-1"
role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button class="close" type="button" data-
dismiss="modal" aria-hidden="true">&times;</button>
        <h4 class="modal-title text-primary"
id="feature-title"></h4>
      </div>
      <div class="modal-body" id="feature-info"></div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default"
data-dismiss="modal">Close</button>
      </div>
    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->

<div class="modal fade" id="attributionModal"
tabindex="-1" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button class="close" type="button" data-
dismiss="modal" aria-hidden="true">&times;</button>
```

```
        <h4 class="modal-title">
            Developed by Ibrahim Lesmana
        </h4>
    </div>

    <div class="modal-body">
        <div id="attribution"></div>
    </div>

</div><!-- /.modal-content -->
</div><!-- /.modal-dialog -->
</div><!-- /.modal -->

    <script src="https://code.jquery.com/jquery-
2.1.4.min.js"></script>

    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/boot
strap.min.js"></script>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/typeahead.js/0.1
0.5/typeahead.bundle.min.js"></script>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.
0.3/handlebars.min.js"></script>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/list.js/1.1.1/li
st.min.js"></script>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/le
aflet.js"></script>
```

```
        <script
src="https://api.tiles.mapbox.com/mapbox.js/plugins/leaflet-
markercluster/v0.4.0/leaflet.markercluster.js"></script>

        <script
src="https://api.tiles.mapbox.com/mapbox.js/plugins/leaflet-
locatecontrol/v0.43.0/L.Control.Locate.min.js"></script>

        <script src="{!! asset('frontend/assets/leaflet-
groupedlayercontrol/leaflet.groupedlayercontrol.js')
!!}"></script>

        <script src="{!! asset('frontend/assets/js/app.js')
!!}"></script>

    </body>
</html>
```

## 2. Controller

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Admin;

class LoginController extends Controller
{

    function cekLogin(Request $req){

        $user= $req->username;

        $pass = $req->password;
```

```

        $check = Admin::where('username', $user)-
>where('password', $pass)->count();

        if( !($check > 0) ) {
            return redirect('/')->with('status', 'salah');
        }

        $take = Admin::where('username', $user)-
>where('password', $pass)->first();

        session(['username' => $take->username]);
        session(['nama' => $take->nama]);
        session(['password' => true]);

        return redirect('/adminpage/home');

    }

    function logout(Request $req) {

        $req->session()->flush();
        $req->session()->regenerate();

        return redirect('/');

    }
}

```

### 3. Model

```
<?php
```

```
namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;

class data_adminadmin extends Authenticatable
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];
}
```

## 4. Maps.Js

```
var map, featureList, boroughSearch = [], theaterSearch =
[], museumSearch = [];

$(window).resize(function() {
    sizeLayerControl();
});

$(document).on("click", ".feature-row", function(e) {
    $(document).off("mouseout", ".feature-row",
clearHighlight);
    sidebarClick(parseInt($(this).attr("id"), 10));
});

if ( !("ontouchstart" in window) ) {
    $(document).on("mouseover", ".feature-row", function(e) {

highlight.clearLayers().addLayer(L.circleMarker([$ (this).att
r("lat"), $(this).attr("lng")], highlightStyle));

    });
}

$(document).on("mouseout", ".feature-row", clearHighlight);

$("#about-btn").click(function() {
    $("#aboutModal").modal("show");
    $(".navbar-collapse.in").collapse("hide");
    return false;
});
```

```
});
```

```
$("#full-extent-btn").click(function() {  
    map.fitBounds(boroughs.getBounds());  
    $(".navbar-collapse.in").collapse("hide");  
    return false;  
});
```

```
$("#legend-btn").click(function() {  
    $("#legendModal").modal("show");  
    $(".navbar-collapse.in").collapse("hide");  
    return false;  
});
```

```
$("#login-btn").click(function() {  
    $("#loginModal").modal("show");  
    $(".navbar-collapse.in").collapse("hide");  
    return false;  
});
```

```
$("#list-btn").click(function() {  
    animateSidebar();  
    return false;  
});
```

```
$("#nav-btn").click(function() {  
    $(".navbar-collapse").collapse("toggle");  
    return false;  
});
```



```
});

$("#sidebar-toggle-btn").click(function() {
    animateSidebar();
    return false;
});

$("#sidebar-hide-btn").click(function() {
    animateSidebar();
    return false;
});

function animateSidebar() {
    $("#sidebar").animate({
        width: "toggle"
    }, 350, function() {
        map.invalidateSize();
    });
}

function sizeLayerControl() {
    $(".leaflet-control-layers").css("max-height",
$("#map").height() - 50);
}

function clearHighlight() {
    highlight.clearLayers();
}
```

```

function sidebarClick(id) {
    var layer = markerClusters.getLayer(id);
    map.setView([layer.getLatLng().lat,
layer.getLatLng().lng], 17);
    layer.fire("click");
    /* Hide sidebar and go to the map on small screens */
    if (document.body.clientWidth <= 767) {
        $("#sidebar").hide();
        map.invalidateSize();
    }
}

function syncSidebar() {
    /* Empty sidebar features */
    $("#feature-list tbody").empty();
    /* Loop through theaters layer and add only features which
are in the map bounds */
    theaters.eachLayer(function (layer) {
        if (map.hasLayer(theaterLayer)) {
            if (map.getBounds().contains(layer.getLatLng())) {
                $("#feature-list tbody").append('<tr class="feature-
row" id="' + L.stamp(layer) + '" lat="' +
layer.getLatLng().lat + '" lng="' + layer.getLatLng().lng +
'"><td style="vertical-align: middle;"></td><td
class="feature-name">' + layer.feature.properties.NAME +

```

```

'</td><td style="vertical-align: middle;"><i class="fa fa-
chevron-right pull-right"></i></td></tr>');

    }

}

});

/* Loop through museums layer and add only features which
are in the map bounds */

museums.eachLayer(function (layer) {

    if (map.hasLayer(museumLayer)) {

        if (map.getBounds().contains(layer.getLatLng())) {

            $("#feature-list tbody").append('<tr class="feature-
row" id="' + L.stamp(layer) + '" lat="' +
layer.getLatLng().lat + '" lng="' + layer.getLatLng().lng +
'"><td style="vertical-align: middle;"></td><td
class="feature-name">' + layer.feature.properties.NAME +
'</td><td style="vertical-align: middle;"><i class="fa fa-
chevron-right pull-right"></i></td></tr>');

        }

    }

});

/* Update list.js featureList */

featureList = new List("features", {

    valueNames: ["feature-name"]

});

featureList.sort("feature-name", {

    order: "asc"

});

```

```

}

/* Basemap Layers */
var cartoLight = L.tileLayer("https://cartodb-basemaps-
{s}.global.ssl.fastly.net/light_all/{z}/{x}/{y}.png", {
  maxZoom: 19,
  attribution: '&copy; <a
href="http://www.openstreetmap.org/copyright">OpenStreetMap<
/a> contributors, &copy; <a
href="https://cartodb.com/attributions">CartoDB</a>'
});
var usgsImagery =
L.layerGroup([L.tileLayer("http://basemap.nationalmap.gov/ar
cgis/rest/services/USGSImageryOnly/MapServer/tile/{z}/{y}/{x
}", {
  maxZoom: 15,
}),
L.tileLayer.wms("http://raster.nationalmap.gov/arcgis/servic
es/Orthoimagery/USGS_EROS_Ortho_SCALE/ImageServer/WMSServer?
", {
  minZoom: 16,
  maxZoom: 19,
  layers: "0",
  format: 'image/jpeg',
  transparent: true,
  attribution: "Aerial Imagery courtesy USGS"
})]);

```

```
/* Overlay Layers */
var highlight = L.geoJson(null);
var highlightStyle = {
    stroke: false,
    fillColor: "#00FFFF",
    fillOpacity: 0.7,
    radius: 10
};

var boroughs = L.geoJson(null, {
    style: function (feature) {
        return {
            color: "black",
            fill: false,
            opacity: 1,
            clickable: false
        };
    },
    onEachFeature: function (feature, layer) {
        boroughSearch.push({
            name: layer.feature.properties.BoroName,
            source: "Boroughs",
            id: L.stamp(layer),
            bounds: layer.getBounds()
        });
    }
});

$.getJSON("data/boroughs.geojson", function (data) {
```

```

    boroughs.addData(data);
});

//Create a color dictionary based off of subway route_id
var subwayColors = {"1":"#ff3135", "2":"#ff3135",
"3":"#ff3135", "4":"#009b2e",
    "5":"#009b2e", "6":"#009b2e", "7":"#ce06cb",
"A":"#fd9a00", "C":"#fd9a00",
    "E":"#fd9a00", "SI":"#fd9a00", "H":"#fd9a00",
"Air":"#ffff00", "B":"#ffff00",
    "D":"#ffff00", "F":"#ffff00", "M":"#ffff00",
"G":"#9ace00", "FS":"#6e6e6e",
    "GS":"#6e6e6e", "J":"#976900", "Z":"#976900",
"L":"#969696", "N":"#ffff00",
    "Q":"#ffff00", "R":"#ffff00" };

var subwayLines = L.geoJson(null, {
    style: function (feature) {
        return {
            color: subwayColors[feature.properties.route_id],
            weight: 3,
            opacity: 1
        };
    },
    onEachFeature: function (feature, layer) {
        if (feature.properties) {
            var content = "<table class='table table-striped
table-bordered table-condensed'>" +

```

```

"<tr><th>Division</th><td>" + feature.properties.Division +
"</td></tr>" + "<tr><th>Line</th><td>" +
feature.properties.Line + "</td></tr>" + "<table>";

    layer.on({
        click: function (e) {
            $("#feature-title").html(feature.properties.Line);
            $("#feature-info").html(content);
            $("#featureModal").modal("show");
        }
    });
}

layer.on({
    mouseover: function (e) {
        var layer = e.target;
        layer.setStyle({
            weight: 3,
            color: "#00FFFF",
            opacity: 1
        });
        if (!L.Browser.ie && !L.Browser.opera) {
            layer.bringToFront();
        }
    },
    mouseout: function (e) {
        subwayLines.resetStyle(e.target);
    }
});

```

```

    }
  });
$.getJSON("data/subways.geojson", function (data) {
  subwayLines.addData(data);
});

/* Single marker cluster layer to hold all clusters */
var markerClusters = new L.MarkerClusterGroup({
  spiderfyOnMaxZoom: true,
  showCoverageOnHover: false,
  zoomToBoundsOnClick: true,
  disableClusteringAtZoom: 16
});

/* Empty layer placeholder to add to layer control for
listening when to add/remove theaters to markerClusters
layer */
var theaterLayer = L.geoJson(null);
var theaters = L.geoJson(null, {
  pointToLayer: function (feature, latlng) {
    return L.marker(latlng, {
      icon: L.icon({
        iconUrl: "assets/img/theater.png",
        iconSize: [24, 28],
        iconAnchor: [12, 28],
        popupAnchor: [0, -25]
      }),
      title: feature.properties.NAME,

```



```

        riseOnHover: true
    });
},
onEachFeature: function (feature, layer) {
    if (feature.properties) {
        var content = "<table class='table table-striped
table-bordered table-condensed'" + "<tr><th>Name</th><td>"
+ feature.properties.NAME + "</td></tr>" +
"<tr><th>Phone</th><td>" + feature.properties.TEL +
"</td></tr>" + "<tr><th>Address</th><td>" +
feature.properties.ADDRESS1 + "</td></tr>" +
"<tr><th>Website</th><td><a class='url-break' href='" +
feature.properties.URL + "' target='_blank'" +
feature.properties.URL + "</a></td></tr>" + "<table>";
        layer.on({
            click: function (e) {
                $("#feature-title").html(feature.properties.NAME);
                $("#feature-info").html(content);
                $("#featureModal").modal("show");

                highlight.clearLayers().addLayer(L.circleMarker([feature.geo
metry.coordinates[1], feature.geometry.coordinates[0]],
highlightStyle));
            }
        });
        $("#feature-list tbody").append('<tr class="feature-
row" id="' + L.stamp(layer) + '" lat="' +
layer.getLatLng().lat + '" lng="' + layer.getLatLng().lng +

```

```

'"><td style="vertical-align: middle;"></td><td
class="feature-name">' + layer.feature.properties.NAME +
'</td><td style="vertical-align: middle;"><i class="fa fa-
chevron-right pull-right"></i></td></tr>');

    theaterSearch.push({
        name: layer.feature.properties.NAME,
        address: layer.feature.properties.ADDRESS1,
        source: "Theaters",
        id: L.stamp(layer),
        lat: layer.feature.geometry.coordinates[1],
        lng: layer.feature.geometry.coordinates[0]
    });
}
}
});
$.getJSON("data/DOITT_THEATER_01_13SEPT2010.geojson",
function (data) {
    theaters.addData(data);
    map.addLayer(theaterLayer);
});

/* Empty layer placeholder to add to layer control for
listening when to add/remove museums to markerClusters layer
*/
var museumLayer = L.geoJson(null);
var museums = L.geoJson(null, {
    pointToLayer: function (feature, latlng) {

```

```

return L.marker(latlng, {
  icon: L.icon({
    iconUrl: "assets/img/museum.png",
    iconSize: [24, 28],
    iconAnchor: [12, 28],
    popupAnchor: [0, -25]
  }),
  title: feature.properties.NAME,
  riseOnHover: true
});
},
onEachFeature: function (feature, layer) {
  if (feature.properties) {
    var content = "<table class='table table-striped
table-bordered table-condensed'" + "<tr><th>Name</th><td>"
+ feature.properties.NAME + "</td></tr>" +
"<tr><th>Phone</th><td>" + feature.properties.TEL +
"</td></tr>" + "<tr><th>Address</th><td>" +
feature.properties.ADRRESS1 + "</td></tr>" +
"<tr><th>Website</th><td><a class='url-break' href='" +
feature.properties.URL + "' target='_blank'" +
feature.properties.URL + "</a></td></tr>" + "<table>";
    layer.on({
      click: function (e) {
        $("#feature-title").html(feature.properties.NAME);
        $("#feature-info").html(content);
        $("#featureModal").modal("show");
      }
    });
  }
}

```

```

highlight.clearLayers().addLayer(L.circleMarker([feature.geometry.coordinates[1], feature.geometry.coordinates[0]],
highlightStyle));
    }
});

$("#feature-list tbody").append('<tr class="feature-
row" id="' + L.stamp(layer) + '" lat="' +
layer.getLatLng().lat + '" lng="' + layer.getLatLng().lng +
'"><td style="vertical-align: middle;"></td><td
class="feature-name">' + layer.feature.properties.NAME +
'</td><td style="vertical-align: middle;"><i class="fa fa-
chevron-right pull-right"></i></td></tr>');

museumSearch.push({
    name: layer.feature.properties.NAME,
    address: layer.feature.properties.ADRESS1,
    source: "Museums",
    id: L.stamp(layer),
    lat: layer.feature.geometry.coordinates[1],
    lng: layer.feature.geometry.coordinates[0]
});
}
}
});

$.getJSON("data/DOITT_MUSEUM_01_13SEPT2010.geojson",
function (data) {
    museums.addData(data);

```

```

});

map = L.map("map", {
    zoom: 14,
    center: [-6.918222, 107.6033072],
    layers: [cartoLight, boroughs, markerClusters, highlight],
    zoomControl: false,
    attributionControl: false
});

/* Layer control listeners that allow for a single
markerClusters layer */
map.on("overlayadd", function(e) {
    if (e.layer === theaterLayer) {
        markerClusters.addLayer(theaters);
        syncSidebar();
    }
    if (e.layer === museumLayer) {
        markerClusters.addLayer(museums);
        syncSidebar();
    }
});

map.on("overlayremove", function(e) {
    if (e.layer === theaterLayer) {
        markerClusters.removeLayer(theaters);
        syncSidebar();
    }
});

```

```
    if (e.layer === museumLayer) {
        markerClusters.removeLayer(museums);
        syncSidebar();
    }
});

/* Filter sidebar feature list to only show features in
current map bounds */
map.on("moveend", function (e) {
    syncSidebar();
});

/* Clear feature highlight when map is clicked */
map.on("click", function(e) {
    highlight.clearLayers();
});

/* Attribution control */
function updateAttribution(e) {
    $.each(map._layers, function(index, layer) {
        if (layer.getAttribution) {
            $("#attribution").html((layer.getAttribution()));
        }
    });
}

map.on("layeradd", updateAttribution);
map.on("layerremove", updateAttribution);
```

```
var attributionControl = L.control({
    position: "bottomright"
});
attributionControl.onAdd = function (map) {
    var div = L.DomUtil.create("div", "leaflet-control-
attribution");
    div.innerHTML = "<span class='hidden-xs'>Developed by
Ibrahim | </span><a href='#'
onclick='$(\"#attributionModal\").modal(\"show\"); return
false;'>Attribution</a>";
    return div;
};
map.addControl(attributionControl);

var zoomControl = L.control.zoom({
    position: "bottomright"
}).addTo(map);

/* GPS enabled geolocation control set to follow the user's
location */
var locateControl = L.control.locate({
    position: "bottomright",
    drawCircle: true,
    follow: true,
    setView: true,
    keepCurrentZoomLevel: true,
    markerStyle: {
        weight: 1,
```

```
        opacity: 0.8,
        fillOpacity: 0.8
    },
    circleStyle: {
        weight: 1,
        clickable: false
    },
    icon: "fa fa-location-arrow",
    metric: false,
    strings: {
        title: "My location",
        popup: "You are within {distance} {unit} from this
point",
        outsideMapBoundsMsg: "You seem located outside the
boundaries of the map"
    },
    locateOptions: {
        maxZoom: 18,
        watch: true,
        enableHighAccuracy: true,
        maximumAge: 10000,
        timeout: 10000
    }
}).addTo(map);

/* Larger screens get expanded layer control and visible
sidebar */
if (document.body.clientWidth <= 767) {
```



```
        var isCollapsed = true;
    } else {
        var isCollapsed = false;
    }

var baseLayers = {
    "Street Map": cartoLight,
    "Aerial Imagery": usgsImagery
};

var layerControl = L.control.groupedLayers(baseLayers, {
    collapsed: isCollapsed
}).addTo(map);

/* Highlight search box text on click */
$("#searchbox").click(function () {
    $(this).select();
});

/* Prevent hitting enter from refreshing the page */
$("#searchbox").keypress(function (e) {
    if (e.which == 13) {
        e.preventDefault();
    }
});

$("#featureModal").on("hidden.bs.modal", function (e) {
```

```

    $(document).on("mouseout", ".feature-row",
clearHighlight);
});

/* Typeahead search functionality */
$(document).one("ajaxStop", function () {
    $("#loading").hide();
    sizeLayerControl();
    /* Fit map to boroughs bounds */
    map.fitBounds(boroughs.getBounds());
    featureList = new List("features", {valueNames: ["feature-
name"]});
    featureList.sort("feature-name", {order:"asc"});

var boroughsBH = new Bloodhound({
    name: "Boroughs",
    datumTokenizer: function (d) {
        return Bloodhound.tokenizers.whitespace(d.name);
    },
    queryTokenizer: Bloodhound.tokenizers.whitespace,
    local: boroughSearch,
    limit: 10
});

var theatersBH = new Bloodhound({
    name: "Theaters",
    datumTokenizer: function (d) {
        return Bloodhound.tokenizers.whitespace(d.name);

```

```

    },
    queryTokenizer: Bloodhound.tokenizers.whitespace,
    local: theaterSearch,
    limit: 10
  });

var museumsBH = new Bloodhound({
  name: "Museums",
  datumTokenizer: function (d) {
    return Bloodhound.tokenizers.whitespace(d.name);
  },
  queryTokenizer: Bloodhound.tokenizers.whitespace,
  local: museumSearch,
  limit: 10
});

var geonamesBH = new Bloodhound({
  name: "GeoNames",
  datumTokenizer: function (d) {
    return Bloodhound.tokenizers.whitespace(d.name);
  },
  queryTokenizer: Bloodhound.tokenizers.whitespace,
  remote: {
    url:
      "http://api.geonames.org/searchJSON?username=bootleaf&featureClass=P&maxRows=5&countryCode=US&name_startsWith=%QUERY",
    filter: function (data) {
      return $.map(data.geonames, function (result) {

```

```

        return {
            name: result.name + ", " + result.adminCode1,
            lat: result.lat,
            lng: result.lng,
            source: "GeoNames"
        };
    });
},
ajax: {
    beforeSend: function (jqXHR, settings) {
        settings.url += "&east=" +
map.getBounds().getEast() + "&west=" +
map.getBounds().getWest() + "&north=" +
map.getBounds().getNorth() + "&south=" +
map.getBounds().getSouth();

        $("#searchicon").removeClass("fa-
search").addClass("fa-refresh fa-spin");
    },
    complete: function (jqXHR, status) {
        $('#searchicon').removeClass("fa-refresh fa-
spin").addClass("fa-search");
    }
}
},
limit: 10
});
boroughsBH.initialize();
theatersBH.initialize();

```

```

museumsBH.initialize();
geonamesBH.initialize();

/* instantiate the typeahead UI */
$("#searchbox").typeahead({
  minLength: 3,
  highlight: true,
  hint: false
}, {
  name: "Boroughs",
  displayKey: "name",
  source: boroughsBH.ttAdapter(),
  templates: {
    header: "<h4 class='typeahead-header'>Boroughs</h4>"
  }
}, {
  name: "Theaters",
  displayKey: "name",
  source: theatersBH.ttAdapter(),
  templates: {
    header: "<h4 class='typeahead-header'><img
src='assets/img/theater.png' width='24'
height='28'>&nbsp;Theaters</h4>",
    suggestion:
Handlebars.compile(["{{name}}<br>&nbsp;<small>{{address}}</s
mall>"].join(""))
  }
}, {

```

```

    name: "Museums",
    displayKey: "name",
    source: museumsBH.ttAdapter(),
    templates: {
      header: "<h4 class='typeahead-header'><img
src='assets/img/museum.png' width='24'
height='28'>&nbsp;Museums</h4>",
      suggestion:
Handlebars.compile(["{{name}}<br>&nbsp;<small>{{address}}</s
mall>"].join(""))
    }
  }, {
    name: "GeoNames",
    displayKey: "name",
    source: geonamesBH.ttAdapter(),
    templates: {
      header: "<h4 class='typeahead-header'><img
src='assets/img/globe.png' width='25'
height='25'>&nbsp;GeoNames</h4>"
    }
  }).on("typeahead:selected", function (obj, datum) {
    if (datum.source === "Boroughs") {
      map.fitBounds(datum.bounds);
    }
    if (datum.source === "Theaters") {
      if (!map.hasLayer(theaterLayer)) {
        map.addLayer(theaterLayer);
      }
    }
  }

```

```

        map.setView([datum.lat, datum.lng], 17);
        if (map._layers[datum.id]) {
            map._layers[datum.id].fire("click");
        }
    }
    if (datum.source === "Museums") {
        if (!map.hasLayer(museumLayer)) {
            map.addLayer(museumLayer);
        }
        map.setView([datum.lat, datum.lng], 17);
        if (map._layers[datum.id]) {
            map._layers[datum.id].fire("click");
        }
    }
    if (datum.source === "GeoNames") {
        map.setView([datum.lat, datum.lng], 14);
    }
    if ($("#navbar-collapse").height() > 50) {
        $("#navbar-collapse").collapse("hide");
    }
}).on("typeahead:opened", function () {
    $("#navbar-collapse.in").css("max-height",
$(document).height() - $("#navbar-header").height());
    $("#navbar-collapse.in").css("height",
$(document).height() - $("#navbar-header").height());
}).on("typeahead:closed", function () {
    $("#navbar-collapse.in").css("max-height", "");
    $("#navbar-collapse.in").css("height", "");

```

```
});  
  
$(".twitter-typeahead").css("position", "static");  
$(".twitter-typeahead").css("display", "block");  
});  
  
// Leaflet patch to make layer control scrollable on touch  
browsers  
var container = $(".leaflet-control-layers")[0];  
if (!L.Browser.touch) {  
    L.DomEvent  
        .disableClickPropagation(container)  
        .disableScrollPropagation(container);  
} else {  
    L.DomEvent.disableClickPropagation(container);  
}
```



