

## KATA PENGANTAR

Dengan menyebut nama Allah Yang Maha Pengasih dan Penyayang. Puji syukur penulis panjatkan kehadiran Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul “Sistem Informasi Geografis (GIS) Mengenai Objek Wisata Di Kabuoten Bondowoso“. Shalawat dan salam atas junjungan besar kita Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Laporan tugas akhir ini disusun untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

Penyusunan laporan tugas akhir ini dapat terlaksana dengan baik karena adanya bantuan secara langsung maupun tidak langsung yang penulis dapatkan dari orang-orang yang selalu mendukung penulis, diantaranya :

1. Bapak Satrio Agung W, S.Kom.,M.kom. selaku dosen pembimbing I yang telah memberikan ilmu dan saran untuk laporan tugas akhir ini.
2. Bapak Denny Sagita R, S.Kom., M.Kom. selaku dosen pembimbing II yang juga memberikan ilmu dan saran untuk laporan tugas akhir ini.
3. Arief Andy Soebroto, S.T., M.Kom. selaku dosen pembimbing akademik saya
4. Bapak Ir. Sutrisno, M.T., Bapak Ir. Heru Nurwasito, M.Kom., Bapak Himawat Aryadita, S.T., M.Sc., dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Bapak Drs. Marji, MT. dan Bapak Issa Arwani, S.Kom., M.Sc. selaku Ketua dan Sekretaris Program Studi Informatika serta segenap Bapak/Ibu Dosen, Staff Administrasi dan Perpustakaan Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya
6. Seluruh dosen teknik Informatika Universitas Brawijaya yang telah atas segala ilmu yang telah diberikan.
7. Orang Tua, yang telah memberikan dukungan moral dan material.
8. Teman-teman yang telah membantu memberi kritik dan saran atas proposal ini.

9. Teman-teman kos yang membantu dalam dukungan moril terima kasih buat mas zaki, mas azam, mas agung, mas alfin, ando, resa dan dhanang
10. Teman-teman angkatan 2009 terutama kelas c yang telah mensupport terima kasih untuk ading, eko, toni, albiyan, robot, alfian, galih danraviqul.

Penulis menyadari bahwa laporan tugas akhir ini masih terdapat banyak kekurangan, oleh karena itu kritik dan saran yang bersifat membangun sangat diharapkan untuk menyempurnakan laporan tugas akhir ini. Saya berharap laporan tugas akhir ini dapat bermanfaat khususnya bagi diri sendiri dan bagi semua pihak.

Malang,

Penulis



## ABSTRAKSI

**Bayu Nursito Utomo. 2015. : SISTEM INFORMASI GEOGRAFIS (GIS) MENGENAI OBJEK WISATA DI KABUPATEN BONDOWOSO. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Satrio Agung W, S.Kom.,M.kom dan Denny Sagita R, S.Kom., M.Kom.**

Kabupaten Bondowoso merupakan salah satu Kabupaten di Provinsi Jawa Timur yang terletak di sebelah timur Pulau Jawa. Bondowoso mempunyai tofografi alam yang berupa dataran tinggi, menjadikan keindahan alamnya semakin menarik. Terdapat juga peninggalan prasejarah dan desa wisata. Potensi ini merupakan aset yang sangat bernilai untuk menarik para wisatawan mancanegara maupun domestik untuk berkunjung melihat keindahan alam yang dimiliki oleh Kabupaten Bondowoso.

Dalam penelitian ini akan dibuat aplikasi sistem informasi geografis mengenai wisata di Kabupaten Bondowoso. Aplikasi ini dapat digunakan sebagai alat bantu bagi para wisatawan untuk mendapatkan informasi dan rute menuju lokasi wisata yang akan dituju. Aplikasi ini menggunakan library leafletjs sebagai dasar pembuatan sistem informasi geografis, basemap yang digunakan dalam peta memakai mapbox. Bahasa pemrograman yang digunakan dalam pengembangan aplikasi ini menggunakan javascript dan php dan MySql dalam penggunaan basis data.

Hasil dari perancangan Aplikasi Sistem Informasi Geografis objek wisata di kabupaten bondowoso ini nantinya akan memberikan informasi dalam bentuk peta yang dapat digunakan sebagai referensi bagi wisatawan yang berkunjung. Lokasi yang ditampilkan pada peta adalah lokasi wisata dan lokasi hotel yang ada di Bondowoso. Penggunaan geolocation dalam aplikasi untuk menentukan keberadaan wisatawan. Selain itu terdapat rute untuk menuju objek wisata dan hotel.

**Kata Kunci:** *Sistem Informasi Geografis, Sistem Informasi Geografis Berbasis Web, Kabupaten Bondowoso, LeafletJS*

## ABSTRACTION

**Bayu Nursito Utomo. 2015 : SISTEM INFORMASI GEOGRAFIS (GIS) MENGENAI OBJEK WISATA DI KABUPATEN BONDOWOSO. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Satrio Agung W, S.Kom.,M.kom dan Denny Sagita R, S.Kom., M.Kom.**

*Bondowoso is a city in East Java which located in the east of Java Island. Bondowoso has nature topography shaped plateau, make the natural beauty more attractive. There are also prehistoric relics and tourist villages. This potential is a very valuable asset to attract foreign and domestic tourists to visit and see the beauty of nature which is owned by bondowoso.*

*In this research will be made geographical information system application about tourism in Bondowoso. This application can be used as supporting tools for tourists who are going to get information and route toward tourism. This application using library leafletjs as the basis for geographical information system, making basemap that used in a map using mapbox. Programming language used in developing applications using javascript and php, mysql in use data base.*

*Results design from application geographic information system applications tourist attraction in Bondowoso this will provide information in the form of maps that can be used as a reference for tourists. Location shown on the map is a tourist location and the location of hotels in Bondowoso. Geolocation in the application to determine the presence of tourists. In addition there is a route to get to the tourist attractions and hotels.*

**Keyword :** *geographic information systems, geographic information systems web based, Bondowoso, LeafletJS*

## DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>i</b>
<b>ABSTRAKSI</b>	<b>iii</b>
<b>ABSTRACTION</b>	<b>iv</b>
<b>DAFTAR ISI</b>	<b>v</b>
<b>DAFTAR GAMBAR</b>	<b>viii</b>
<b>DAFTAR TABEL</b>	<b>x</b>
<b>BAB I Pendahuluan</b>	<b>1</b>
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
<b>BAB II Tinjauan Pustaka</b>	<b>5</b>
2.1 Profil Kabupaten Bondowoso	5
2.2 Potensi Wisata Di Kabupaten Bondowoso	5
2.3 Sistem Informasi Geografis (SIG)	7
2.3.1 Komponen Pembentuk Sistem Informasi Geografis	8
2.3.2 Sistem Informasi Geografis Berbasis Web	9
2.3.3 Data Spasial	11
2.4 LeafletJS	13
2.5 Framework Codeigniter	15
2.6 <i>Unified Modeling Language (UML)</i>	17
2.6.1 <i>UseCase Diagram</i>	17
2.6.2 <i>Activity Diagram</i>	18
2.6.3 <i>Sequence Diagram</i>	19
2.6.4 <i>Class Diagram</i>	21
2.7 <i>Entity Relationship Model</i>	23
2.7 Pengujian Perangkat Lunak	27
2.7.1 <i>Blacbox Testing</i>	27

2.7.2 Usability Testing	28
<b>BAB III Metodologi Penelitian</b>	<b>29</b>
3.1 Studi Literatur	30
3.2 Pengumpulan Data	30
3.3 Analisis Kebutuhan Sistem	30
3.4 Perancangan Sistem	30
3.4.1 Perancangan <i>UseCase Diagram</i>	30
3.4.2 Perancangan <i>Activity Diagram</i>	31
3.4.3 Perancangan <i>Sequence Diagram</i>	31
3.4.4 Perancangan <i>Class Diagram</i>	31
3.4.5 Perancangan Basis Data	31
3.4.6 Perancangan <i>User Interface</i>	32
3.5 Implementasi Sistem	32
3.6 Pengujian Sistem	32
3.7 Kesimpulan Dan Saran	32
<b>BAB IV Analisis Dan Perancangan</b>	<b>33</b>
4.1 Analisis Kebutuhan Sistem	34
4.1.1 Analisis Data	34
4.1.2 Kebutuhan Fungsional	34
4.1.3 Kebutuhan NonFungsional	35
4.1.4 Identifikasi Aktor	35
4.2 Analisis Komponen	36
4.3 Perancangan Sistem	36
4.3.2 Perancangan <i>UseCase Diagram</i>	36
4.3.3 Perancangan <i>Activity Diagram</i>	45
4.3.4 Perancangan <i>Sequence Diagram</i>	50
4.3.5 Perancangan Kelas	61
4.3.6 Perancangan Basis Data	62
4.3.7 Perancangan <i>User Interface</i>	66
<b>BAB V Implementasi</b>	<b>69</b>
5.1 Spesifikasi Sistem	70

5.1.1 Perangkat Keras ( <i>Hardware</i> )	70
5.1.2 Perangkat Lunak ( <i>Software</i> )	70
5.1.3 Batasan Implementasi	70
5.2 Implementasi Kode Model Pada Codeigniter	71
5.3 Implementasi Kode Pada Aplikasi	80
5.4 Implementasi Database	87
5.5 Implementasi <i>User Interface</i>	90
<b>BAB VI Pengujian Dan Analisis</b>	<b>92</b>
6.1 <i>BlackBox Testing</i> (Uji Validasi)	93
6.2 <i>Usability Testing</i>	101
6.3 Analisis Pengujian Validasi	104
6.4 Analisis Pengujian Usability	104
<b>BAB VII Kesimpulan Dan Saran</b>	<b>105</b>
7.1 Kesimpulan	105
7.2 Saran	105
<b>DAFTAR PUSTAKA</b>	<b>xii</b>
<b>LAMPIRAN</b>	<b>xiii</b>

## DAFTAR GAMB

Gambar 2.1 Komponen Utama SIG	9
Gambar 2.2 Arsitektur Webgis	10
Gambar 2.3 Model Data Spasial	12
Gambar 2.4 LeafletJS dengan Basemap Openstreet Map	14
Gambar 2.5 Flow Codeigniter	16
Gambar 2.6 <i>One To One</i>	24
Gambar 2.7 <i>One To Many</i>	25
Gambar 2.8 <i>Many To Many</i>	25
Y	
Gambar 3.1 Flow Chart Metode Penelitian	29
Gambar 4.1 Pohon Perancangan	33
Gambar 4.2 <i>Class Diagram</i>	37
Gambar 4.3 Menampilkan Halaman Peta Wisata	45
Gambar 4.4 Mencari Lokasi	45
Gambar 4.5 Mendapatkan Rute	46
Gambar 4.6 Memberikan Like Pada Tempat Wisata	47
Gambar 4.7 Memeberikan Komentar Pada Objek Wisata	47
Gambar 4.8 Proses Tambah, Hapus, Ubah data	48
Gambar 4.9 <i>Sequence Diagram</i> Menambahkan Data Kecamatan	50
Gambar 4.10 <i>Sequence Diagram</i> mengubah data kecamatan	51
Gambar 4.11 <i>Sequence Diagram</i> menghapus data kecamatan	52
Gambar 4.12 <i>Sequence Diagram</i> Menambahkan Data Wisata	53
Gambar 4.13 <i>Sequence Diagram</i> Mengubah Data Wisata	54
Gambar 4.14 <i>Sequence Diagram</i> Menghapus Data Wisata	55
Gambar 4.15 <i>Sequence Diagram</i> Menambah Data Hotel	56
Gambar 4.16 <i>Sequence Diagram</i> Mengubah Data Hotel	57
Gambar 4.17 <i>Sequence Diagram</i> Menghapus Data Hotel	58
Gambar 4.18 <i>Sequence Diagram</i> Memberikan Rating Pada Objek Wisata	59
Gambar 4.19 <i>Sequence Diagram</i> Pengguna Memberikan Rating Pada Hotel	60
Gambar 4.20 <i>Sequence Diagram</i> Memberikan Komentar	61

Gambar 4.21 Hubungan Antar Kelas	62
Gambar 4.22 ER Diagram	63
Gambar 4.23 ER Mapping	63
Gambar 4.24 <i>Physical Diagram</i>	64
Gambar 4.25 Halaman Utama	66
Gambar 4.26 Halaman Map	67
Gambar 4.27 Halaman Review Wisata	6
Gambar 5. 1 Pohon Implementasi	69
Gambar 5.2 Halaman Home	90
Gambar 5.3 Halaman Map	90
Gambar 5.4 Halaman Informasi Wisata	91
Gambar 5.5 Implementasi Rute Wisata	91
Y	
Gambar 6.1 Pohon Pengujian	92

## DAFTAR TAB

Tabel 2.1 Prinsip-prinsip pengembangan SIG	10
Tabel 2.2 Contoh Representasi Data Vektor dan Atributnya	13

Tabel 2.3 Mendefinisikan Leaflet JS	14
Tabel 2.4 <i>UseCase Diagram</i>	18
Tabel 2.5 <i>Activity Diagram</i>	19
Tabel 2.6 <i>Sequence Diagram</i>	21
Tabel 2.7 <i>Class Diagram</i>	22
Tabel 2.8 Notasi ERD	27
Y	
Tabel 4.1 Daftar Kebutuhan Fungsional	35
Tabel 4.2 Kebutuhan NonFungsional	35
Tabel 4.3 Indentifikasi Aktor	36
Tabel 4.4 Skenario <i>UseCase</i> Melihat Peta	38
Tabel 4.5 Skenario <i>UseCase</i> Mencari Lokasi User	38
Tabel 4.6 Skenario <i>UseCase</i> Mencari Rute	39
Tabel 4.7 Skenario <i>UseCase</i> Memberi Rating	39
Tabel 4. 8 Skenario <i>UseCase</i> Memberi Komentar	40
Tabel 4.9 Skenario <i>UseCase</i> Tambah Data Kecamatan	40
Tabel 4.10 Skenario <i>UseCase</i> Ubah Data Kecamatan	41
Tabel 4.11 Skenario <i>UseCase</i> Hapus Data Kecamatan	41
Tabel 4.12 Skenario <i>UseCase</i> Tambah Data Wisata	42
Tabel 4.13 Skenario <i>UseCase</i> Ubah Data Wisata	42
Tabel 4.14 Skenario <i>UseCase</i> Hapus Data Wisata	43
Tabel 4.15 Skenario <i>UseCase</i> Tambah Data Hotel	43
Tabel 4.16 Skenario <i>UseCase</i> Ubah Data Hotel	44
Tabel 4.17 Skenario <i>UseCase</i> Hapus Data Hotel	44
Tabel 4.18 Struktur Tabel Admin	64
Tabel 4. 19 Struktur Tabel Kecamatan	64
Tabel 4.20 Struktur Tabel Kecamatan	64
Tabel 4.21 Struktur Tabel Wisata	65
Tabel 4.22 Struktur Tabel Hotel	65
Tabel 4.23 Struktur Tabel Komentar	65

Tabel 5.1 Spesifikasi Perangkat Keras	70
Tabel 5.2 Spesifikasi Perangkat Lunak	70
Tabel 5.3 Implementasi SItte_model	76
Tabel 5.4 Implementasi Alam_model	77
Tabel 5. 5 Implementasi Budaya_model	79
Tabel 5.6 Implementasi Hotel_Model	80
Tabel 5.7 Implementasi Geojson Data Wisata Alam	80
Tabel 5.8 Implementasi Geojson Data Budaya	81
Tabel 5.9 Implementasi Geojson Data Hotel	81
Tabel 5.10 Implementasi Basemap	81
Tabel 5.11 Implementasi Menampilkan Data Geojson Pada Sidebar dan Marker	83
Tabel 5.12 Implementasi Rute Plan Pada Peta	84
Tabel 5.13 Implementasi Komentar	86
Tabel 5.14 Implementasi Voting	87
Tabel 5.15 Implementasi Tabel Admin	87
Tabel 5.16 Implementasi Tabel Kecamatan	88
Tabel 5.17 Implementasi Tabel Wisata	88
Tabel 5. 18 Implementasi Tabel Jenis Wisata	88
Tabel 5.19 Implementasi Tabel Hotel	89
Tabel 5.20 Implementasi Tabel Komentar Wisata Alam	89
Tabel 5.21 Implementasi Tabel Komentar Hotel	89
Tabel 6.1 Pengujian Login Admin	93
Tabel 6.2 Pengujian Logout Admin	93
Tabel 6.3 Kasus Uji Validasi Tambah Data Kecamatan	94
Tabel 6.4 Kasus Uji Validasi Ubah Data Kecamatan	94
Tabel 6.5 Kasus Uji Validasi Hapus Data Kecamatan	95
Tabel 6.6 Kasus Uji Validasi Tambah Data Wisata	95
Tabel 6.7 Kasus Uji Validasi Ubah Data Wisata	96
Tabel 6.8 Kasus Uji Validasi Hapus Data Wisata	96
Tabel 6.9 Kasus Uji Validasi Tambah Data Hotel	97

Tabel 6.10 Kasus Uji Validasi Ubah Data Hotel	97
Tabel 6.11 Kasus Uji Validasi Hapus Data Hotel	98
Tabel 6.12 Kasus Uji Validasi Menampilkan Peta Digital	98
Tabel 6.13 Kasus Uji Validasi Mencari Lokasi Pengguna	99
Tabel 6.14 Kasus Uji Validasi Mencari Lokasi Pengguna	99
Tabel 6.15 Kasus Uji Validasi Mencari Rute Menuju Objek Wisata	100
Tabel 6.16 Kasus Uji Validasi Pemberian Rating Objek Wisata Dan Hotel	100
Tabel 6.17 Kasus Uji Validasi Mencari Mencari Objek Wisata Dan Hotel	101
Tabel 6.18 Kuisisioner Usability Testing	102
Tabel 6.19 Rincian Jumlah Dari Jawaban Hasil Kuesioner	103



## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Pariwisata merupakan salah satu industri besar yang dapat menampung banyak tenaga kerja dan mampu menghasilkan pendapatan besar kepada daerah. Hal itu juga tidak terlepas dengan banyaknya jumlah pengunjung atau wisatawan domestik maupun mancanegara yang datang ke Bondowoso. Kabupaten Bondowoso merupakan salah satu Kabupaten di Provinsi Jawa Timur. Bondowoso mempunyai tofografi alam yang berupa dataran tinggi, menjadikan keindahan alamnya semakin menarik. Terdapat juga peninggalan prasejarah dan desa wisata. Potensi ini merupakan aset yang sangat bernilai untuk menarik para wisatawan mancanegara maupun domestik untuk berkunjung melihat keindahan alam yang dimiliki oleh Kabupaten Bondowoso.

Informasi tentang objek wisata di Bondowoso yang disediakan pada situs resmi ([bondowosokab.go.id](http://bondowosokab.go.id)), masih kurang memenuhi kebutuhan informasi wisata yang ada di Bondowoso. Untuk mengetahui kekurangan situs resmi Dinas Pariwisata Bondowoso, peneliti melakukan wawancara kepada masyarakat yang pernah mengunjungi situs resmi Dinas Bondowoso. Wawancara difokuskan pada penyajian informasi geografis tentang lokasi wisata Bondowoso. Dari hasil wawancara yang dilakukan terdapat banyak kekurangan dalam penyajian informasi geografis pada situs Dinas Pariwisata Bondowoso. Menurut masyarakat peta yang disajikan kurang interaktif. Peta yang disediakan terlalu kecil, untuk pengunjung dari luar bondowoso merasa kesulitan mencari lokasi wisata dan rute menuju lokasi wisata. Informasi peta yang tersedia hanya berupa titik dan setiap titik tidak menjelaskan nama dari wisata dan lokasi wisata tersebut. Pengunjung tidak dapat berinteraksi langsung dengan peta karena peta yang disediakan bersifat statis. Untuk memenuhi kebutuhan masyarakat yang akan berkunjung ke Bondowoso maka perlu dibuat suatu layanan informasi yang berfungsi sebagai media pengunjung untuk mengetahui lokasi wisata yang ada di daerah Bondowoso.

Untuk mengatasi permasalahan tersebut maka dalam penelitian ini dirancang suatu sistem informasi geografis berbasis web menggunakan *library leafletjs* untuk mempermudah masyarakat mengetahui informasi dan lokasi wisata yang berada di Kabupaten Bondowoso. *Library leafletjs* digunakan sebagai dasar pembuatan sistem informasi geografis. Sistem informasi terdahulu banyak kekurangan berdasarkan wawancara yang dilakukan peneliti kepada masyarakat. Oleh sebab itu, penulis merancang suatu sistem untuk menyajikan informasi geografis. Dari uraian diatas penulis mengangkat judul “Sistem Informasi Geografis (SIG) Mengenai Objek Wisata Di Kabupaten Bondowoso”.

Dalam penelitian Sistem Informasi Geografis (SIG) Mengenai Objek Wisata Di Kabupaten Bondowoso ini peneliti mengungkapkan bahwa adanya kemungkinan besar semakin banyak turis domestik maupun mancanegara yang berkunjung ke Kabupaten Bondowoso sistem aplikasi ini diterapkan.

### **1.2 Rumusan Masalah**

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalahan pada skripsi ini yaitu sebagai berikut:

1. Bagaimana menganalisa kebutuhan data yang terkait dengan kebutuhan informasi website pariwisata Bondowoso berdasarkan website yang sudah ada sebelumnya.
2. Bagaimana merancang dan mengimplementasikan Sistem Informasi Geografis Berbasis Web menggunakan *library leafletjs*
3. Bagaimana menguji Sistem Informasi Geografis Mengenai Objek Wisata di Kabupaten Bondowoso

### **1.3 Batasan Masalah**

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi dalam hal:

1. Studi kasus yang digunakan adalah website Dinas Pariwisata Bondowoso yang sudah ada sebelumnya.
2. Pada penelitian ini peneliti menggunakan *leafletjs* sebagai *library* dalam pembuatan Sistem Informasi Geografis.
3. Sistem informasi dibangun berdasarkan kebutuhan pengguna yang diambil secara acak.

4. Sistem informasi hanya menyediakan informasi wisata alam, wisata budaya dan hotel yang ada Di Kabupaten Bondowoso.
5. Sistem yang dibuat berbasis web

#### **1.4 Tujuan Penelitian**

Tujuan dari pemanfaatan Sistem Informasi Geografis adalah untuk mempermudah masyarakat mendapatkan informasi wisata di Kabupaten Bondowoso yang telah diolah dan tersimpan sebagai atribut suatu lokasi atau obyek. Memperbaiki kekurangan sistem yang sudah ada sebelumnya.

#### **1.5 Manfaat Penelitian**

Adapun manfaat dari pengembangan perangkat lunak Sistem Informasi Geografis (GIS) Mengenai Objek Wisata Di Kabupaten Bondowoso adalah sebagai media informasi tentang wisata di Bondowoso yang berupa peta digital.

#### **1.6 Sistematika Penulisan**

Sistematika penyusunan laporan ditunjukkan untuk memberikan gambaran dan uraian dari laporan skripsi secara garis besar yang meliputi beberapa bab, sebagai berikut.

##### **BAB I : Pendahuluan**

Menguraikan latar belakang skripsi, rumusan masalah tentang pengembangan penelitian, tujuan dan manfaat skripsi serta sistematika penyusunan laporan skripsi.

##### **BAB II : Tinjauan Pustaka**

Menguraikan teori-teori yang menjadi referensi dalam pelaksanaan skripsi.

##### **BAB III : Metode Penelitian**

Menguraikan tentang metode dan langkah kerja yang dilakukan dalam proses perancangan dan implementasi pada pelaksanaan skripsi dan perancangan sistem yang menjadi objek studi kasus skripsi.

##### **BAB IV : Analisis dan Perancangan**

Menguraikan proses perancangan perangkat lunak.

##### **BAB V : Implementasi**

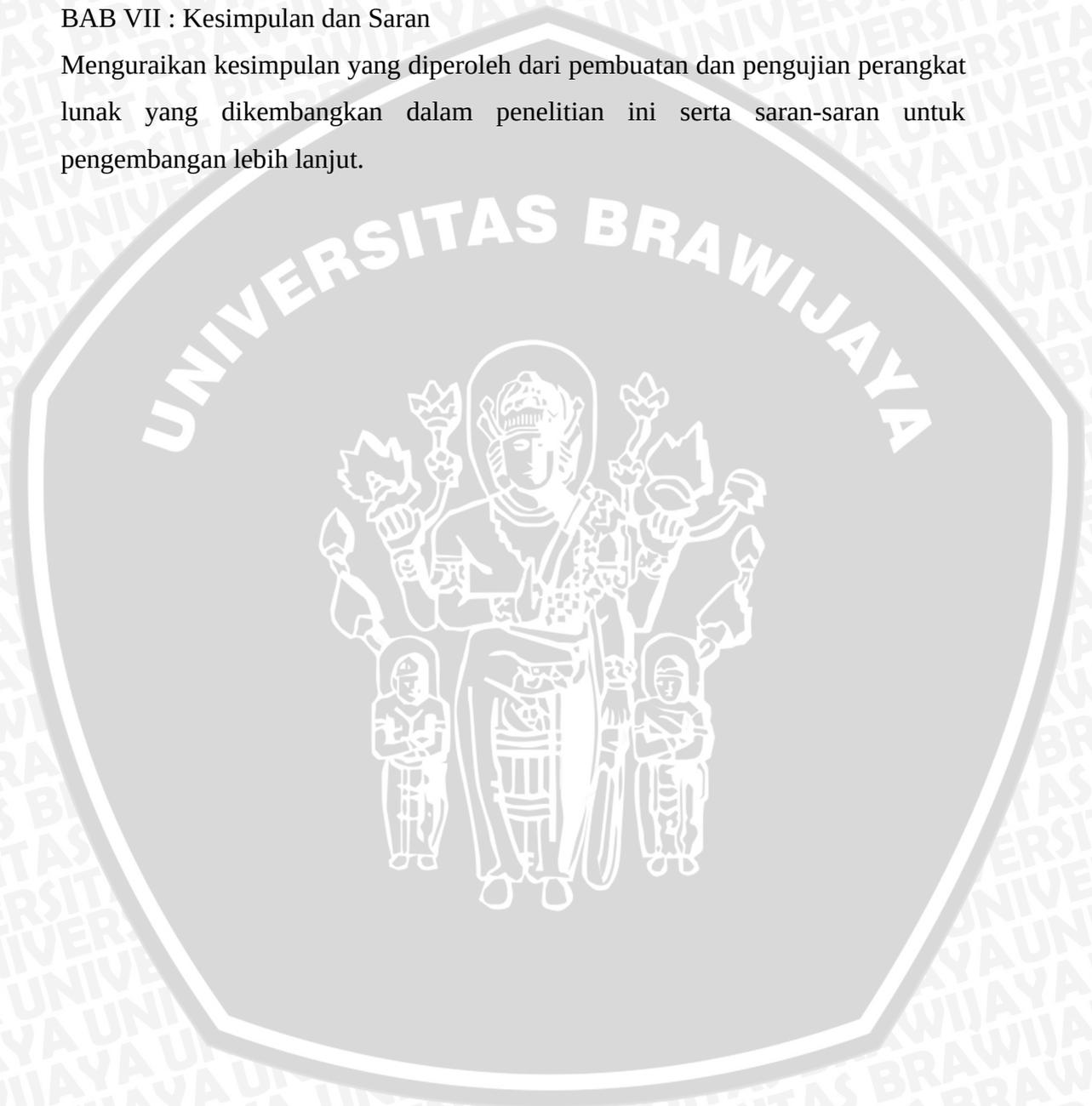
Menguraikan proses implementasi dari teori yang telah dipelajari sesuai analisis dan perancangan sistem.

## BAB VI : Pengujian dan Analisis

Menguraikan proses pengujian serta kendala yang dihadapi selama pengujian sistem.

## BAB VII : Kesimpulan dan Saran

Menguraikan kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam penelitian ini serta saran-saran untuk pengembangan lebih lanjut.



## BAB II TINJAUAN PUSTAKA

Tinjauan pustaka dalam bab ini menjelaskan tentang kajian pustaka dan teori dasar dari aplikasi yang dibuat. Tinjauan pustaka Kajian Pustaka merupakan rangkuman singkat yang komprehensif tentang semua materi terkait yang terdapat di dalam berbagai referensi tentang Sistem Informasi Geografis, Codeigniter dan Leafletjs. Dasar teori membahas teori yang dijadikan landasan pengembangan “Sistem Informasi Geografis (GIS) Mengenai Objek Wisata Di Kabupaten Bondowoso”. Dasar teori yang digunakan meliputi teori.

### **2.1 Profil Kabupaten Bondowoso**

Kabupaten Bondowoso adalah salah satu kabupaten dalam Provinsi Jawa Timur yang terletak di sebelah timur Pulau Jawa. Dikenal dengan sebutan daerah tapal kuda. Ibukotanya adalah Bondowoso. Kabupaten Bondowoso memiliki luas wilayah 1.560,10 km<sup>2</sup> yang secara geografis berada pada koordinat antara 113°48'10" - 113°48'26" BT dan 7°50'10" - 7°56'41" LS[BWS-14].

Kabupaten Bondowoso memiliki suhu udara yang cukup sejuk berkisar 15,40 OC – 25,10 OC, karena berada di antara pegunungan Kendeng Utara dengan puncaknya Gunung Raung, Gunung Ijen dan sebagainya di sebelah timur serta kaki pegunungan Hyang dengan puncak Gunung Argopuro, Gunung Krincing dan Gunung Kilap di sebelah barat. Sedangkan di sebelah utara terdapat Gunung Alas Sereh, Gunung Biser dan Gunung Bendusa[BWS-14].

Letak Kabupaten Bondowoso tidak berada pada daerah yang strategis. Meskipun berada ditengah, namun Kabupaten Bondowoso tidak dilalui jalan negara yang menghubungkan antar provinsi. Bondowoso juga tidak memiliki lautan. Ini yang menyebabkan Bondowoso sulit berkembang dibandingkan dengan kabupaten lainnya di Jawa Timur[BWS-14].

### **2.2 Potensi Wisata Di Kabupaten Bondowoso**

Pariwisata, seni dan budaya merupakan salah satu kegiatan yang diharapkan dapat menunjang pertumbuhan ekonomi masyarakat, yang berdampak pada meningkatnya pendapatan daerah. Kunjungan wisatawan baik nusantara maupun mancanegara diharapkan dapat menggerakkan perekonomian masyarakat. Berdasarkan Peraturan Daerah Nomor 8 Tahun 2007 tentang RT/RW Kabupaten Bondowoso[BWS-14], ditetapkan kawasan wisata Kabupaten Bondowoso yaitu :

1. Kawasan pariwisata budaya meliputi [BWS-14] :
  - a. Kawasan wisata sejarah dan budaya dengan obyek:
    1. Situs megalitikum di Kecamatan Bondowoso, Kecamatan Grujugan, Kecamatan Jambesari Darus Sholah, Kecamatan Maesan, Kecamatan Prajekan, Kecamatan Pujer, Kecamatan Tamanan, Kecamatan Tegalampel, Kecamatan Tlogosari, Kecamatan Wonosari, dan Kecamatan Wringin.
    2. Situs Gua Buto di Kecamatan Cerme.
    3. Wisata ziarah makam Ki Ronggo di Kecamatan Tegalampel.
    4. Wisata budaya Padepokan Gema Buana di Kecamatan Prajekan.
    5. Wisata kerajinan kuningan Cindogo di Kecamatan Tapen.
    6. Desa wisata Blimbing di Kecamatan Klabang.
    7. Desa wisata Tamanan, Kalianyar, dan Sumberkemuning di Kecamatan Tamanan.
    8. Desa wisata Gunungsari di Kecamatan Maesan.
  - b. Kawasan wisata rekreasi dengan obyek wisata Alun-alun Bondowoso di Kecamatan Bondowoso.
  - c. Monumen Gerbong Maut
2. Kawasan pariwisata alam meliputi[BWS-14] :
  - a. Kawasan wisata terpadu Kawah Ijen – Raung dengan obyek :
    1. Wisata Kawah Ijen, Kawah Telaga Weru dan Kawah Wurung di Kecamatan Sempol.
    2. Wisata air terjun Blawan dan Gua Stalagtit di Kecamatan Sempol.
    3. Wisata pemandian air panas Blawan dan pemandian Damarwulan di Kecamatan Sempol.
    4. Wisata agro di Kecamatan Sempol, Sumberwringin, Sukosari dan Tlogosari.
    5. Wisata Air Terjun Puloagung di Desa Sukorejo Kecamatan Sumberwringin.
  - b. Kawasan wisata terpadu Lereng Argopuro dengan obyek [BWS-14]:
    1. Wisata pendakian Pegunungan Hyang atau Gunung Argopuro di Kecamatan Pakem.
    2. Wisata agro Pusat Penelitian Kopi dan Kakao di Kecamatan Pakem.
    3. Wisata air terjun Tancak Kembar di Kecamatan Pakem.
    4. Wisata Pemandangan Arak-Arak di Kecamatan Wringin.
    5. Wisata megalitikum Situs Batu Labeng di Kecamatan Wringin.
  - c. Kawasan wisata rekreasi dengan obyek [BWS-14]:
    1. Wisata Pemandian Tasnan di Kecamatan Grujugan.
    2. Wisata Bendung Sampean Baru di Kecamatan Tapen.

- d. Kawasan wisata minat khusus dengan obyek [BWS-14]:
  1. Wisata arung jeram Bosamba di Kecamatan Taman Krocok dan Tapen.
  2. Kawasan wisata pendakian Gunung Raung di Kecamatan Sumberwringin.
  3. Wisata panjat tebing alam Pantirana di Kecamatan Grugujan.
  4. Wisata petualangan di Desa Solor Kecamatan Cermee.

Di Kabupaten Bondowoso juga disediakan sarana akomodasi penginapan yang memadai bagi wisatawan. Pada tahun 2008 ini jumlah hotel di Kabupaten Bondowoso terdiri dari 11 hotel. Satu hotel bintang 3 yaitu Hotel Ijen View di Jalan KIS Mangunsarkoro. Sedangkan lainnya yaitu hotel melati. Enam hotel di Kota Bondowoso yaitu Palm, Anugerah, Baru, Slamet, Kinanti dan Grand serta 4 hotel di luar Kota Bondowoso yaitu Arabica, Catimore, Jampit, dan Wisata Asri [BWS-14].

### 2.3 Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis adalah sistem informasi khusus yang mengelola

data yang memiliki informasi spasial (bereferensi keruangan), atau dalam arti yang lebih sempit adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut lokasinya, dalam sebuah database [EA-11].

Data Geografis yang dimaksud adalah data yang terdiri atas lokasi suatu geografis yang diset ke dalam bentuk koordinat yang ciri-cirinya [EA-11]:

- a. Memiliki atribut geometri seperti koordinat dan lokasi.
- b. Terkait dengan aspek ruang seperti kota dan kawasan pembangunan.
- c. Berhubungan dengan fenomena yang terdapat di bumi misal data kejadian, gejala, dan objek.

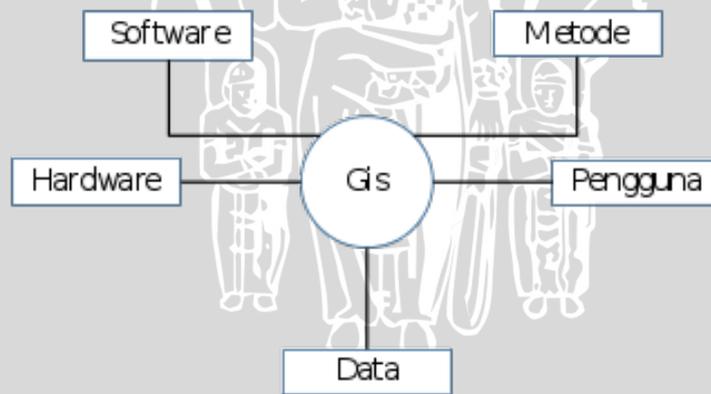
Sistem Informasi Geografis dapat diuraikan menjadi beberapa sub sistem sebagai berikut [HR-06]:

- a. Data Input: Subsistem ini bertugas untuk mengumpulkan data dan mempersiapkan data spasial dan atribut dari berbagai sumber dan bertanggung

- jawab dalam mengkonversi atau mentransfortasikan format-format data-data aslinya kedalam format yang dapat digunakan oleh GIS.
- b. Data Ouput : Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data baik dalam bentuk softcopy maupun bentuk hardcopy seperti : table, grafik dan peta.
- c. Data Management: Subsistem ini mengorganisasikan baik data spasial maupun data atribut ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil, diupdate dan di edit.
- d. Data *Manipulation and Analysis*: Subsistem ini menentukan informasi yang dapat dihasilkan oleh GIS dan melakukan manipulasi serta pemodelan data untuk menghasilkan informasi yang diharapkan.

**2.3.1 Komponen Pembentuk Sistem Informasi Geografis**

Dalam membangun sebuah Sistem Informasi Geografi, diperlukan beberapa komponen - komponen pembentuk sistem tersebut. Sehingga dalam kenyataannya, SIG memerlukan integrasi dari beberapa komponen. Karena SIG merupakan produk dari beberapa komponen. Komponen - komponen yang membentuk SIG diantaranya adalah :



Gambar 2.1 Komponen Utama SIG [EA-11]

1. Pengguna : orang yang menjalankan sistem meliputi mengoperasikan, mengembangkan bahkan memperoleh manfaat dari sistem. Kategori orang yang menjadi bagian dari SIG ini ada beragam, misalnya operator, analis, programmer, database administrator bahkan stakeholder.

2. *Software* : Dalam pembuatan GIS di perlukan software yang menyediakan fungsi tool yang mampu melakukan penyimpanan data, analisis dan menampilkan informasi geografis.
3. *Hardware* : GIS membutuhkan komputer untuk penyimpanan dan pemprosesan data. Ukuran dari sistem komputerisasi bergantung pada tipe GIS itu sendiri. GIS dengan skala yang kecil hanya membutuhkan PC (personal computer) yang kecil dan sebaliknya.
4. *Metode* : Metode yang digunakan dalam SIG akan berbeda untuk setiap permasalahan. SIG yang baik tergantung pada aspek desain dan aspek realnya.
5. *Data* : Data merupakan bagian terpenting dari SIG. data di perlukan dalam SIG meliputi data Spasial dan Non Spasial. Data terdiri dari fakta-fakta dan angka-angka yang relative.

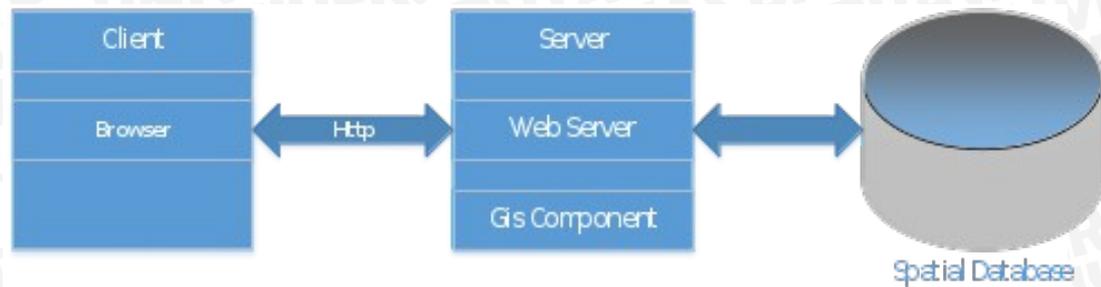
### 2.3.2 Sistem Informasi Geografis Berbasis Web

Pengembangan aplikasi GIS kedepannya mengarah kepada aplikasi berbasis Web. Hal ini disebabkan karena pengembangan aplikasi di lingkungan jaringan telah menunjukkan potensi yang besar dalam kaitannya dengan geo informasi. Sebagai contoh adalah adanya peta online sebuah kota dimana pengguna dapat dengan mudah mencari lokasi yang diinginkan secara online melalui jaringan internet tanpa mengenal batas geografi penggunanya. Di lingkungan web prinsip-prinsip tersebut di gambarkan dan di implementasikan seperti pada tabel berikut:

<b>Gis Prinsip</b>	<b>Pengembang Web</b>
Data Input	Client
Manajemen Data	DBMS Dengan Komponen Spasial
Analisis Data	GIS Library di Server
Representasi Data	Client Server

Tabel 2.1 Prinsip-prinsip pengembangan SIG [CD-08]

Untuk dapat melakukan komunikasi dengan komponen yang berbeda-beda di lingkungan web maka dibutuhkan sebuah *web server*. Karena standar dari geo data berbeda beda dan sangat spesifik maka pengembangan arsitektur system mengikuti arsitektur '*Client Server*'.



Gambar 2.2 Arsitektur Webgis [CD-08]

Gambar diatas menunjukkan arsitektur minimum sebuah sistem Web GIS. Aplikasi berada disisi client yang berkomunikasi dengan server sebagai penyedia data. Aplikasi ini bisa dikembangkan dengan berbasis web. Web Server bertanggung jawab terhadap proses permintaan dari client dan mengirimkan tanggapan terhadap respon tersebut. Dalam arsitektur web, sebuah web server juga mengatur komunikasi dengan server side SIG Komponen. Server side SIG Komponen bertanggung jawab terhadap koneksi kepada database spasial seperti menterjemahkan query kedalam SQL dan membuat representasi yang diteruskan ke server. Dalam kenyataannya Side Server SIG Komponen berupa software libraries yang menawarkan layanan khusus untuk analisis spasial pada data. Selain komponen hal lain yang juga sangat penting adalah aspek fungsional yang terletak di sisi client atau di server [CD-08].

Untuk melakukan manajemen data geografis dibutuhkan sebuah DBMS (Database Management System). Ada beberapa aplikasi yang dapat digunakan sebagai database seperti Oracle Spatial, PostgreSQL, Informix, DB2, Ingres dan yang paling populer saat ini adalah MySQL.

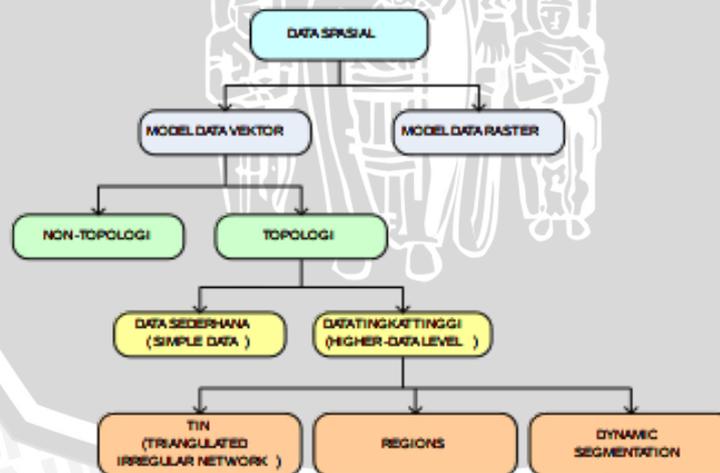
Terdapat banyak aplikasi Web Gis pada internet. Hal ini dipengaruhi oleh berkembangnya web programming dan adanya peluang komersial yang dapat dimanfaatkan. Berikut beberapa contoh Web Gis atau peta online:

1. Yahoo Maps
2. Openstreet Map
3. Mapbox Map
4. Google Map

### 2.3.3 Data Spasial

Data spasial mempunyai pengertian sebagai suatu data yang mengacu pada posisi, obyek, dan hubungan diantaranya dalam ruang bumi. Data spasial merupakan salah satu item dari informasi, dimana didalamnya terdapat informasi mengenai bumi termasuk permukaan bumi, dibawah permukaan bumi, perairan, kelautan dan bawah atmosfer[GD-07]. Perkembangan teknologi yang cepat dalam pengambilan data spasial telah membuat perekaman terhadap data berubah menjadi bentuk digital, selain itu relatif cepat dalam melakukan prosesnya. Salah satunya perkembangan teknologi yang berpengaruh terhadap perekaman data pada saat ini adalah teknologi penginderaan jauh (*remote sensing*) dan *Global Positioning System* (GPS) [GD-07].

Terdapat dua model dalam data spasial, yaitu model data raster dan model data vektor. Keduanya memiliki karakteristik yang berbeda, selain itu dalam pemanfaatannya tergantung dari masukan data dan hasil akhir yang akan dihasilkan. Model data tersebut merupakan representasi dari obyek-obyek geografi yang terekam sehingga dapat dikenali dan diproses oleh komputer[GD-07].



Gambar 2.3 Model Data Spasial [GD-07]

Model data vektor merupakan model data yang paling banyak digunakan, model ini berbasiskan pada titik (points) dengan nilai koordinat (x,y) untuk

membangun obyek spasialnya. Obyek yang dibangun terbagi menjadi tiga bagian lagi yaitu berupa titik (point), garis (line), dan area (polygon) [GD-07].

- Titik (point)

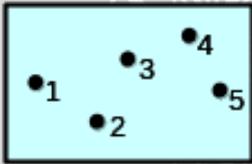
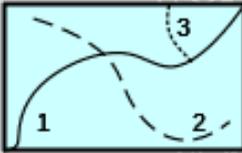
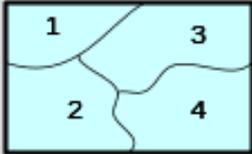
Titik merupakan representasi grafis yang paling sederhana pada suatu obyek. Titik tidak mempunyai dimensi tetapi dapat ditampilkan dalam bentuk simbol baik pada peta maupun dalam layar monitor. Contoh : Lokasi Fasilitas Kesehatan, Lokasi Fasilitas Kesehatan, dll.

- Garis (line)

Garis merupakan bentuk linear yang menghubungkan dua atau lebih titik dan merepresentasikan obyek dalam satu dimensi. Contoh : Jalan, Sungai, dll.

- Area (Poligon)

Poligon merupakan representasi obyek dalam dua dimensi. Contoh : Danau, Persil Tanah, dll.

Jenis	Contoh Representasi	Contoh Atribut		
<b>Titik</b>		id	Nama	Lokasi
		1	Wisata A	Kec A
		2	Wisata B	Kec B
<b>Line</b>		id	Status	Kondisi
		1	Jalan Nasional	Baik
		2	Jalan Provinsi	Sedang
<b>Polygon</b>		id	Guna Lahan	Status
		1	Sawah	20
		2	Permukiman	30

Tabel 2.2 Contoh Representasi Data Vektor dan Atributnya [GD-07]

## 2.4 LeafletJS

LeafletJS merupakan library atau kumpulan fungsi berbasis Javascript yang digunakan untuk menampilkan peta interaktif pada halaman web [MM-14]. Leaflet menyediakan Map API (*Application Programming Interface*) yang memudahkan web developer untuk menampilkan peta berbasis Tile pada halaman

web. Pengguna peta juga dapat berinteraksi dengan menggunakan perangkat tertentu yang telah disediakan oleh leaflet.

Dibandingkan Map API lainnya seperti OpenLayer, GMAP API, ArcGIS JS API atau Mapbox API, Leaflet memiliki keunggulan dalam kecepatan akses dan ukuran berkas yang kecil. Meskipun memiliki ukuran berkas yang lebih kecil, Leaflet memiliki fungsi web mapping yang cukup lengkap. Komponen Leaflet JS terdiri dari beberapa elemen sebagai berikut [MM-14]:

1. Tipe Raster (TileLayer dan ImageOverlay)
2. Tipe Vektor (Path, Polygon, dan tipe lain yang spesifik seperti Circle)
3. Tipe Grup (LayerGroup, FeatureGroup and GeoJSON)

Leaflet JS dapat mendukung berbagai format data untuk ditampilkan pada peta online, seperti GeoJSON, KML, CSV, GPX dan WKT. Data spasial yang disimpan pada Leaflet dapat disajikan dalam bentuk layanan WMS atau static JSON. Tidak ada dukungan untuk penyajian data spasial melalui WFS maupun GML.

Leaflet JS merupakan JavaScript Tool sehingga perlu untuk menulis JavaScript dan HTML untuk membuat peta. Leaflet JS seperti toolbox yang memegang banyak hal untuk peta di web : kartografi & citra udara , alat pengembang, peta untuk aplikasi mobile, dan banyak lagi. Menginisialisasi LeafletJS pada web browser dapat menggunakan dengan dua cara [MM-14]:

1. Mengunduh file yang berisi library LeafletJS  
<http://leafletjs.com/download.html>
2. Menggunakan CDN (Content Delivery Network) yang menyimpan kode Leaflet

```
<link rel="stylesheet"
href="http://cdn.leafletjs.com/leaflet-
0.7.3/leaflet.css" />

<scriptsrc="http://cdn.leafletjs.com/leaflet-
0.7.3/leaflet.js"></script>
```

Tabel 2.3 Mendefinisikan *Leaflet JS*



Gambar 2.4 LeafletJS dengan Basemap Openstreet Map

Tile Server merupakan sumber tile map yang digunakan leaflet untuk menyajikan

peta. Karena ada banyak kebutuhan yang unik untuk peta ada sejumlah besar variasi

server tile map yang menerapkan format yang berbeda. Pada penelitian ini peneliti menggunakan beberapa tile server yang digunakan sebagai Basemap. Untuk mendefinisikan tile server pada webgis di leafletjs menggunakan URL Template [MM-14]. Misalnya untuk openstreet map.

`http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png`

Keterangan

`http://{s}.tile.openstreetmap.org` : Nama Tile Server

`{z}` : Zoom

`{x}/{y}` : Lokasi dari Tile (koordinat).

Untuk memanggil data *leaflet JS* menggunakan geojson. GeoJSON adalah format untuk encoding berbagai struktur data geografis. Sebuah objek GeoJSON merupakan geometri, fitur, atau kumpulan fitur. GeoJSON mendukung jenis geometri berikut: *Point*, *LineString*, *Polygon*, *MultiPoint*, *MultiLineString*, *MultiPolygon*, dan *GeometryCollection*[MM-14]. Fitur di GeoJSON mengandung objek geometri dan sifat tambahan, dan koleksi fitur merupakan daftar fitur. *Leaflet JS* mendukung semua jenis GeoJSON di atas, tetapi Fitur dan *FeatureCollections* bekerja dengan baik karena memungkinkan untuk menggambarkan fitur dengan satu set properti[MM-14].

## 2.5 Framework Codeigniter

*Codeigniter* merupakan sebuah *web application* yang bersifat *open source* digunakan memebangun aplikasi PHP dinamis. Tujuan utama pengembangan

*Codeigniter* adalah membantu developer untuk mengerjakan aplikasi cepat dari pada menulis semua code dari awal [DI-11]:.

*Codeigniter* dibangun menggunakan konsep MVC (*Model-View-Controller*) *development pattern*. Kelebihan dari *Codeigniter* sangat ringan dan terstruktur. Selain itu *Codeigniter* juga memiliki fitur-fitur lainnya yang sangat bermanfaat, antara lain [DI-11]:

1. Menggunakan Pattern MVC
2. *URL Friendly*
3. Kemudahan dalam mempelajari, membuat *library* dan *helper*, memodifikasi dan mengintegrasikan antara *library* dan *helper*

MVC (*Model-View-Controller*) adalah sebuah pattern pemrograman yang memisahkan *bisnis logic* (alur pikir), *data logic* (penyimpanan data) dan *presentation logic* (antarmuka aplikasi) atau secara sederhana memindahkan antara desain, data dan proses. Komponen-komponen MVC antara lain [DI-11]::

#### 1. Model

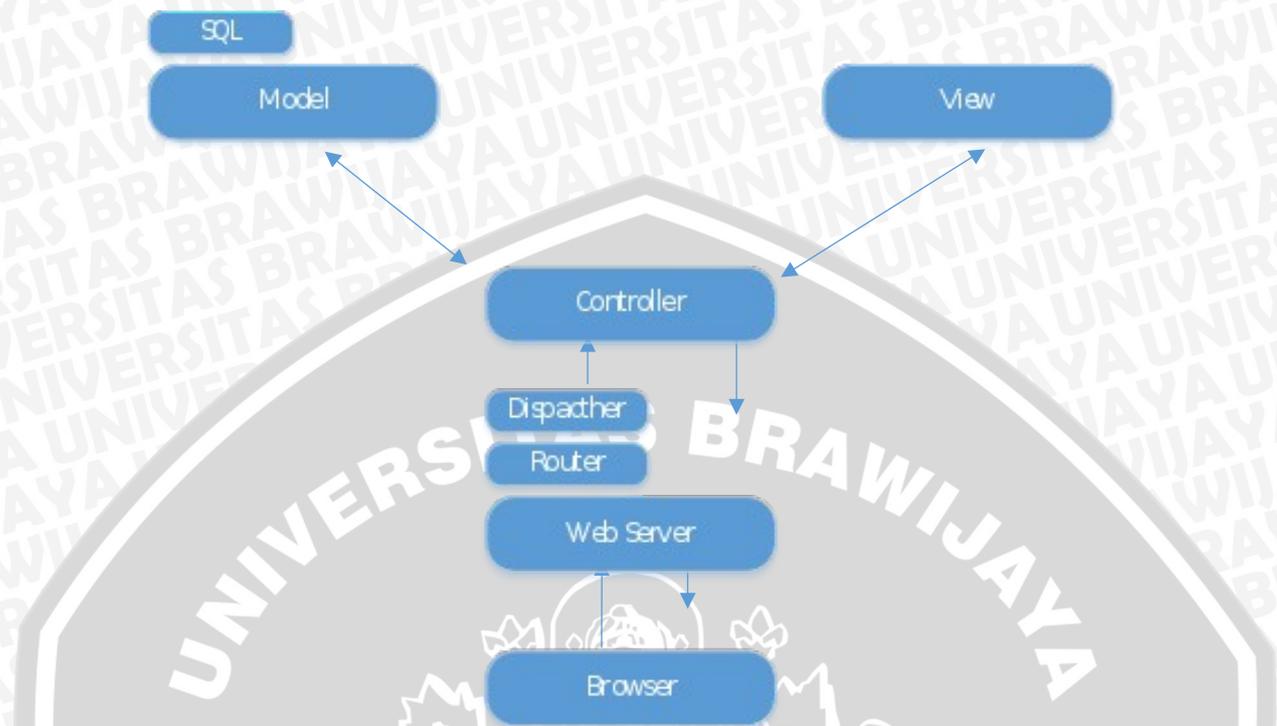
Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga mempresentasikan struktur data dari aplikasi yang berupa basis data. Biasanya didalam model berisi kelas yang berfungsi untuk mengambil, mengupdate dan menghapus data. Aplikasi website biasanya menggunakan basis data dalam penyimpanan data, maka pada bagian model berisi perintah-perintah yang berhubungan dengan SQL Query.

#### 2. View

View berhubungan dengan segala sesuatu yang ditampilkan pada *end user*. Bisa berupa halaman web, rss dan *javascript*.

#### 3. Controller

Controller berfungsi sebagai penghubung antara model dan view. Didalam controller terdapat kelas yang berfungsi memproses permintaan dari view ke dalam struktur data pada model.



Gambar 2.5 Flow Codeigniter [DI-11]

Jika dipetakan, alur kerja *Codeigniter* akan tampak seperti gambar 2.5 Browser berinteraksi melalui controller. Controller yang akan menerima dan membalas semua *request* dari browse. Untuk data maka controller akan meminta ke model dan untuk *user interface* akan meminta ke view. Jadi otak dari aplikasi berada pada controller. UI aplikasi ada di view dan data ada di model. Ketika browser meminta sebuah halaman maka router akan mencari controller mana yang harus menangani *request*. Setelah itu barulah controller menggunakan model untuk mengakses data dan view untuk menampilkan data.

### 2.6 Unified Modeling Language (UML)

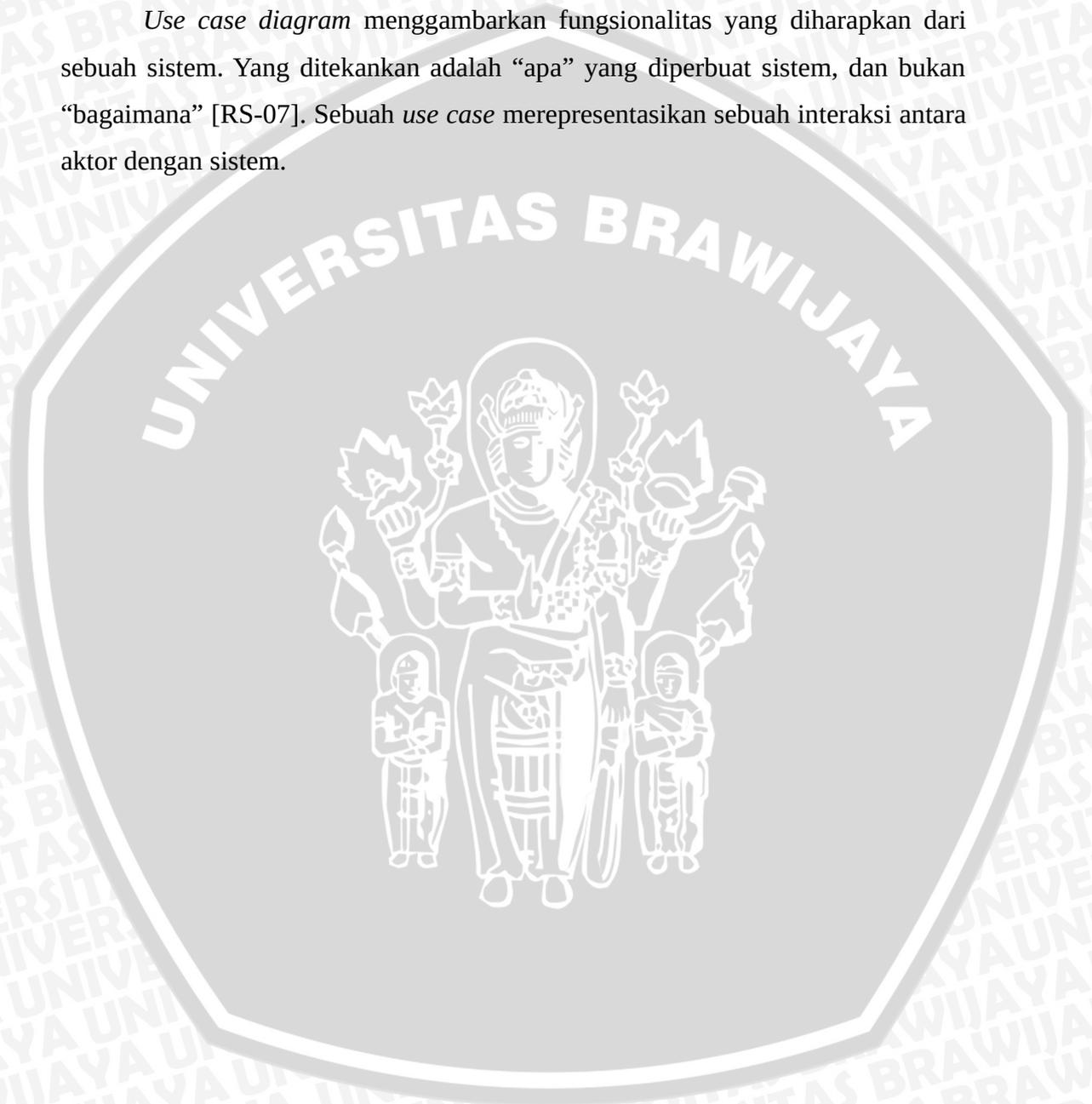
*Unified Modelling Language* (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak [RS-07]. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga

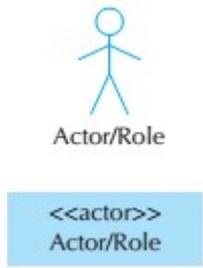
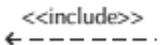
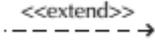


menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa – bahasa berorientasi objek [RS-07].

### 2.6.1 UseCase Diagram

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana” [RS-07]. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.



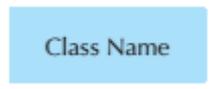
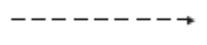
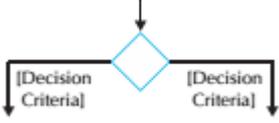
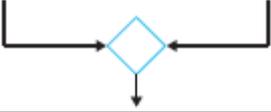
<p>Actor</p> <ol style="list-style-type: none"> <li>1. Orang atau sistem yang mendapat manfaat dari subject</li> <li>2. digambarkan sebagai tongkat (default), jika aktor bukan manusia yang terlibat, maka digambarkan persegi panjang dengan &lt;&lt; aktor &gt;&gt; di dalamnya.</li> <li>3. D diberi label dengan perannya</li> <li>4. Dapat di asosiasikan dengan actor lain menggunakan specialization/superclass</li> <li>5. Di letakkan di luar boundary</li> </ol>	
<p>Use Case</p> <ol style="list-style-type: none"> <li>1. Extend ke use case lainnya</li> <li>2. Include ke use case lainnya.</li> <li>3. Merupakan bagian utama dari fungsi sistem</li> </ol>	
<p>Association Relationship</p> <p>menghubungkan aktor dengan use case</p>	
<p>Include Relationship</p> <p>Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya</p>	
<p>Extend Relationship</p> <p>Sebuah Use-case juga dapat meng-extend Use-case lain dengan behaviour-nya sendiri</p>	
<p>Generalization Relationship</p> <p>Mewakilkkan penggunaan use case secara lebih umum</p>	

Tabel 2.4 UseCase Diagram [DWT-07]

### 2.6.2 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang [DWT-07], bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram

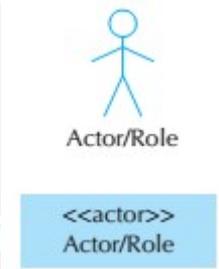
juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

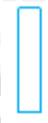
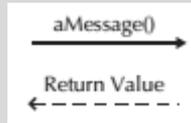
<p>Action</p> <ol style="list-style-type: none"> <li>1. langkah-langkah dalam sebuah aktivitas.</li> <li>2. Aksi bisa terjadi saat memasuki aktifitas, meninggalkan aktivitas, saat di dalam aktivitas, atau pada kejadian (event) yang spesifik.</li> </ol>	
<p>Activity</p> <ol style="list-style-type: none"> <li>1. kegiatan dalam aliran kerja</li> <li>2. digunakan untuk mewakili serangkaian tindakan</li> </ol>	
<p>Object Node</p> <ol style="list-style-type: none"> <li>1. Digunakan untuk mewakili sebuah objek yang terhubung ke satu set objek flow</li> <li>2. Entitas-entitas yang digunakan dalam aliran kerja</li> </ol>	
<p>Control Flow</p> <p>Menunjukkan urutan eksekusi.</p>	
<p>Object Flow</p> <p>Menunjukkan aliran objek dari activity (action) ke activity (action) lainnya</p>	
<p>Initial Node</p> <p>Menunjukkan dimana aliran kerja itu di mulai</p>	
<p>Final Activity Node</p> <p>Menunjukkan dimana aliran kerja itu berakhir</p>	
<p>Final Flow Node</p> <p>Menghentikan control flow atau object flow tertentu</p>	
<p>Decision Node</p> <ol style="list-style-type: none"> <li>1. Menunjukkan dimana sebuah keputusan perlu dibuat dalam aliran kerja.</li> <li>2. Diberi label dengan kriteria keputusan</li> </ol>	
<p>Merge Node</p> <p>menunjukkan dua atau lebih langkah dalam aliran kerja berjalan secara serentak.</p>	

Tabel 2.5 Activity Diagram [DWT-07]

### 2.6.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu [DWT-07]. Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu [DWT-07].

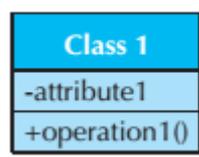
<p>Actor</p> <ol style="list-style-type: none"> <li>1. Digunakan untuk menghadirkan pemakai yang memulai alur peristiwa / kejadian.</li> <li>2. Ditempatkan di bagian atas diagram</li> <li>3. Digambarkan sebagai tongkat (default), jika aktor bukan manusia yang terlibat, maka digambarkan persegi panjang dengan &lt;&lt; aktor &gt;&gt; di dalamnya.</li> <li>4. Berpartisipasi secara berurutan dengan mengirim dan menerima pesan</li> </ol>	
<p>General</p> <p>Merepresentasikan entitas tunggal dalam sequence diagram, digambarkan dengan kotak. Entitas ini memiliki nama, stereotype atau berupa instance (menggunakan instance:class)</p>	
<p>Boundary</p> <p>Boundary biasanya berupa tepi dari system, seperti user interface, atau suatu alat yang berinteraksi dengan system lain.</p>	
<p>Control</p> <p>Control element mengatur aliran dari informasi untuk sebuah skenario. Perilaku dan perilaku bisnis umumnya diatur oleh objek ini</p>	
<p>Entity</p> <p>Entity biasanya elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa beans atau model object</p>	

<p><b>Lifetime</b></p> <ol style="list-style-type: none"> <li>Berisi X pada titik di mana kelas tidak lagi berinteraksi.</li> <li>Menggambarkan hubungan suatu elemen yang berbeda</li> <li>Elemen individu di dalam suatu interaksi</li> </ol>	
<p><b>Execution Occurrence</b></p> <ol style="list-style-type: none"> <li>Persegi panjang yang ditempatkan di atas lifeline</li> <li>Menunjukkan ketika suatu objek mengirim atau menerima pesan</li> </ol>	
<p><b>Guard Condotion</b></p> <p>Merupakan syarat yang dipenuhi untuk pesan yang akan dikirim.</p>	
<p><b>Object Destruction</b></p> <p>Sebuah X ditempatkan pada akhir objek, untuk menunjukkan akan keluar eksistensi.</p>	
<p><b>Frame</b></p> <p>Menunjukkan Kontext sequence diagram</p>	
<p><b>Message</b></p> <ol style="list-style-type: none"> <li>Menyampaikan informasi dari satu objek ke objek lain</li> <li>Panggilan operasi diberi label dengan pesan yang dikirim dan panah.</li> <li>return diberi label dengan nilai yang akan dikembalikan dan ditampilkan, digambarkan dengan panah putus-putus</li> </ol>	

Tabel 2.6 *Sequence Diagram* [DWT-07]

### 2.6.4 Class Diagram

*Class diagram* digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam system [DWT-07]. Class diagram memberikan gambaran sistem secara statis dan relasi antar mereka.

<p><b>Class</b></p> <ol style="list-style-type: none"> <li>Mewakilkkan orang, tempat, atau hal yang berhubungan dengan system yang perlu di-capture dan menyimpan informasi</li> <li>Nama diketik dalam huruf tebal dan berada di atas</li> </ol>	
---	---

<p>compartment</p> <ol style="list-style-type: none"> <li>Mempunyai list atribut ditengah compartment</li> <li>Mempunyai list operation dibawah compartment</li> <li>Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas.</li> </ol>	
<p>Attribut</p> <ol style="list-style-type: none"> <li>Menggambarkan keadaan suatu objek.</li> <li>Bisa berasal dari atribut lainnya, yang ditunjukkan oleh garis miring sebelum nama atribut itu.</li> </ol>	<p>attribute name /derived attribute name</p>
<p>Operation</p> <ol style="list-style-type: none"> <li>Merupakan tindakan atau fungsi Class</li> <li>Dapat diklasifikasikan sebagai konstruktor, query, atau operasi update</li> <li>tanda kurung yang berisi parameter atau informasi yang diperlukan untuk melakukan operasi.</li> </ol>	<p>operation name ()</p>
<p>Association</p> <ol style="list-style-type: none"> <li>Merupakan hubungan antara beberapa class atau class itu sendiri</li> <li>Bisa di antara satu atau lebih class.</li> <li>berlabel menggunakan frase kata kerja atau nama peran</li> <li>Berisi simbol, yang merupakan minimum dan maksimum waktu class</li> </ol>	<p>AssociatedWith 0..* 1</p>
<p>Generalization</p> <p>Merupakan hubungan antara beberapa kelas.</p>	
<p>Aggregation</p> <ol style="list-style-type: none"> <li>Merupakan hubungan logis antara beberapa class atau class itu sendiri</li> <li>bentuk khusus dari asosiasi.</li> </ol>	<p>0..* IsPartOf 1</p> 
<p>Composition</p> <ol style="list-style-type: none"> <li>Merupakan hubungan fisik antara beberapa class atau class itu sendiri</li> <li>bentuk khusus dari asosiasi.</li> </ol>	<p>1..* IsPartOf 1</p> 

Tabel 2.7 *Class Diagram* [DWT-07]

## 2.7 Entity Relationship Model

Diagram Hubungan Entitas atau *Entity Relationship Model* merupakan model data berupa notasi grafis dalam pemodelan data konseptual yang menggambarkan hubungan antara penyimpan [JZ-10]. Model data terdiri dari model hubungan entitas dan model relasional. Model ER mendefinisikan pandangan konseptual database. *Entity Relationship Model* bekerja di sekitar entitas dan hubungan asosiasi di antara mereka. Pada tingkat tampilan, model ER dianggap sebagai pilihan yang baik untuk merancang database.

Berikut ini bagian dari *Entity Relationship Model* [JZ-10] :

### a Entitas

Entitas adalah suatu objek yang dapat didefinisikan dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Misalnya, dalam database sekolah, siswa, guru, kelas, dan program studi yang ditawarkan dapat dianggap sebagai entitas. Semua entitas ini memiliki beberapa atribut atau properti yang memberikan identitas mereka [JZ-10].

1. *Super Key*: Satu set atribut (satu atau lebih) yang secara kolektif mengidentifikasi entitas dalam himpunan entitas.
2. *Candidate Key*: Minimal super key disebut kunci kandidat. Set entitas dapat memiliki lebih dari satu kunci kandidat.
3. *Primary Key*: salah satu *Candidate Key* yang dipilih oleh desainer database untuk secara unik mengidentifikasi himpunan entitas.

### b Atribut

Entitas mempunyai elemen yang disebut atribut, dan berfungsi mendeskripsikan karakter dari entitas. Atribut adalah properti atau karakteristik yang dimiliki oleh suatu entitas dimana properti atau karakteristik itu bermakna atau berarti bagi organisasi atau perusahaan, misalnya untuk pencatatan data pegawai di suatu instansi, entitas pegawai mungkin memiliki atribut-atribut nomor induk pegawai, nama, alamat, nomor telepon, gaji pokok dan lainnya [JZ-10]. Setiap diagram hubungan entitas bisa terdapat lebih dari satu atribut. Atribut digambarkan dalam bentuk elips. Entitas memiliki himpunan atribut yang berasosiasi dengannya. Berikut beberapa tipe attribute [JZ-10]

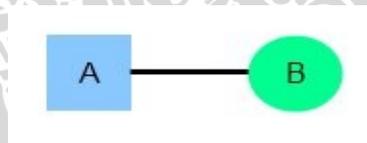
- 1 *Simple attribute* : atribut tidak dapat dibagi-bagi menjadi atribut yang sederhana
- 2 *Composite attribute* : atribut yang dapat dipecah menjadi atribut lain
- 3 *Derivide attribute* : atribut yang tidak ada dalam database fisik, tetapi nilai-nilai mereka yang berasal dari atribut lain yang hadir dalam database.
- 4 *Single-Value attribute* : atribut yang mengandung nilai tunggal
- 5 *Multi-Value attribute* : atribut yang bisa berisi lebih dari satu nilai

**c. Hubungan diantara Entitas**

Suatu hubungan adalah bagaimana data tersebut dibagi antara entitas. Ada tiga jenis hubungan antara entitas[JZ-10]:

1. One To One

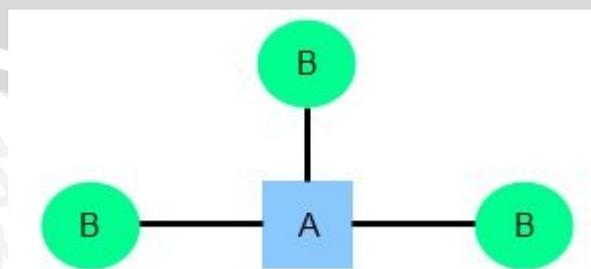
Salah satu contoh dari suatu entitas (A) dikaitkan dengan satu contoh lain dari entitas lain (B). Misalnya, dalam database karyawan, setiap nama karyawan (A) dikaitkan dengan hanya satu nomor jaminan sosial (B) [JZ-10].



Gambar 2.6 One To One [JZ-10]

2. One To Many

Salah satu contoh dari suatu entitas (A) dikaitkan dengan nol, satu atau banyak contoh entitas lain (B), tapi untuk satu contoh dari entitas B hanya ada satu contoh dari entitas A. Sebagai contoh, untuk sebuah perusahaan dengan semua karyawan yang bekerja dalam satu bangunan, nama bangunan (A) dikaitkan dengan banyak karyawan yang berbeda (B), tetapi karyawan semua berbagi hubungan tunggal yang sama dengan entitas A[JZ-10].

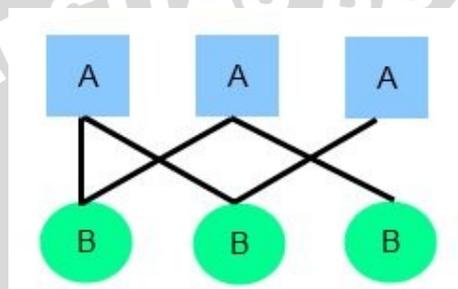


Gambar 2.7 One To Many [JZ-10]



3. Many To Many

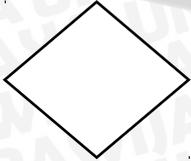
Salah satu contoh dari suatu entitas (A) dikaitkan dengan satu, nol atau banyak contoh dari entitas lain (B), dan satu contoh dari entitas B berhubungan dengan satu, nol atau banyak contoh dari entitas A. Misalnya, untuk sebuah perusahaan di yang semua karyawan bekerja pada beberapa proyek, setiap contoh dari seorang karyawan (A) dikaitkan dengan banyak contoh proyek (B), dan pada saat yang sama, setiap contoh dari proyek (B) memiliki beberapa karyawan (A) terkait dengan itu[JZ-10].

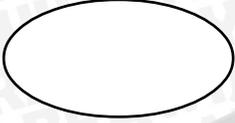
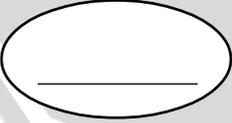
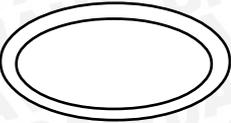


Gambar 2.8 Many To Many [JZ-10]

d. Simbol Entity Relationship Model

Simbol Simbol dalam Entity Relationship Diagram Simbol-simbol yang digunakan dalam entity relationship diagram adalah sebagai berikut ini [JZ-10]:

Notasi	Keterangan
	Entitas ialah suatu objek yang dapat dibedakan dengan objek lainnya. Entitas berfungsi untuk memberikan identitas pada entitas yang memiliki label dan nama. Entitas memiliki bentuk persegi panjang.
	Relasi ialah hubungan yang terjadi antara 1 entitas atau lebih yang tidak mempunyai fisik tetapi hanya sebagai konseptual. Dan berfungsi untuk mengetahui jenis hubungan yang ada antara 2 file. Relasi memiliki bentuk belah ketupat.

	<p>Atribut ialah karakteristik dari entitas atau relasi yang menyediakan penjelasan detil tentang entitas atau relasi tersebut. Dan berfungsi untuk memperjelas atribut yang dimiliki oleh sebuah entitas. Atribut memiliki bentuk lingkaran lebih tepatnya elips.</p>
	<p>Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi. Dan berbentuk garis.</p>
	<p>Weak Entity set, Entity set yang bergantung terhadap strong entity set. Digambarkan dengan empat persegi panjang bertumpuk</p>
	<p>Identifying relationship merupakan Relationship yang menghubungkan strong dan weak entity dan Digambarkan dengan belahketupat dengan garis ganda dan berelasi dengan</p>
	<p>Primary Key beberapa atribut yang mempunyai nilai unik sehingga dapat digunakan untuk membedakan data pada suatu baris/record dengan baris lain pada suatu entitas. Digambarkan dengan elips dan garis bertumpuk</p>
	<p>Multi-valued Attribute yang dapat terdiri dari sekumpulan harga untuk suatu entitas tertentu. Digambarkan dengan elips bertumpuk</p>

Tabel 2.8 Notasi ERD [JZ-10]

## 2.7 Pengujian Perangkat Lunak

Pengujian ini dimaksudkan untuk menunjukkan bahwa program dapat melakukan apa yang menjadi tujuan perangkat lunak tersebut. Pengujian perangkat lunak dapat dilakukan dengan mengeksekusi program dengan data percobaan. Kita dapat memeriksa hasil uji coba untuk mendapatkan kesalahan, anomali, atau informasi tentang atribut nonfungsional dalam program.

Metode desain uji kasus perangkat lunak telah berevolusi. Metode ini memberikan pendekatan yang sistematis dalam pengujian bagi para pengembang. Metode menyediakan suatu mekanisme yang dapat membantu memastikan kelengkapan tes dan memberikan kemungkinan tertinggi dalam mengungkap kesalahan-kesalahan dalam perangkat lunak. Setiap produk rekayasa perangkat lunak dapat diuji dalam dua cara [MG-12]:

### 2.7.1 *Blacbox Testing*

*Black-box testing* juga dibutuhkan pengujian perilaku berfokus pada persyaratan fungsional perangkat lunak. *Black-box testing* memungkinkan pengembang perangkat lunak untuk men-set kondisi input untuk melaksanakan semua persyaratan fungsional program. *Black-box testing* menggunakan pendekatan komplementer yang memungkinkan untuk mengungkap kesalahan [MG-12]. *Black-box testing* dapat mengungkap kesalahan dalam kategori berikut [MG-12]:

- Fungsi yang hilang atau tidak benar
- Kesalahan antarmuka
- Kesalahan dalam struktur data
- Kesalahan perilaku atau kinerja
- Kesalahan inisialisasi dan terminasi

Ada banyak keuntungan dalam penggunaan metode *black-box testing*. Berikut adalah beberapa keuntungan penggunaan metode *black-box testing* [MG-12]:

- Kemudahan dalam penggunaan. Karena penguji hanya perlu menyibukkan diri dengan uji kasus yang bekerja dalam aplikasi.

- b. Pengembangan uji kasus yang lebih cepat. Karena penguji hanya perlu menyibukkan diri dengan berbagi perilaku melalui antarmuka pengguna aplikasi.

### 2.7.2 Usability Testing

Usability adalah komponen penting dalam *User Experience*. *Usability* adalah kegunaan, bagaimana suatu produk bisa berguna, dan dapat dengan mudah (dan baik) dipergunakan [MG-12]. Menurut Jakob Nielsen, *Usability* meliputi: *Learnability* (fungsi-fungsi dasar produk mudah dioperasikan sejak awal), *Efficiency* (fungsi-fungsi produk dapat dengan cepat dipergunakan), *Memorability* (setelah pengguna meninggalkan sistem dalam waktu lama, lalu bisa dengan mudah mengerti lagi cara penggunaannya), *Errors* (seberapa banyak dan serius kesalahan penggunaan produk, dan bagaimana pengguna dapat memperbaiki kesalahan), *Satisfaction* (seberapa puaskah pengguna). Pengujian *Usability* perlu dilakukan dengan orang-orang yang termasuk dalam kelompok pengguna yang di targetkan.

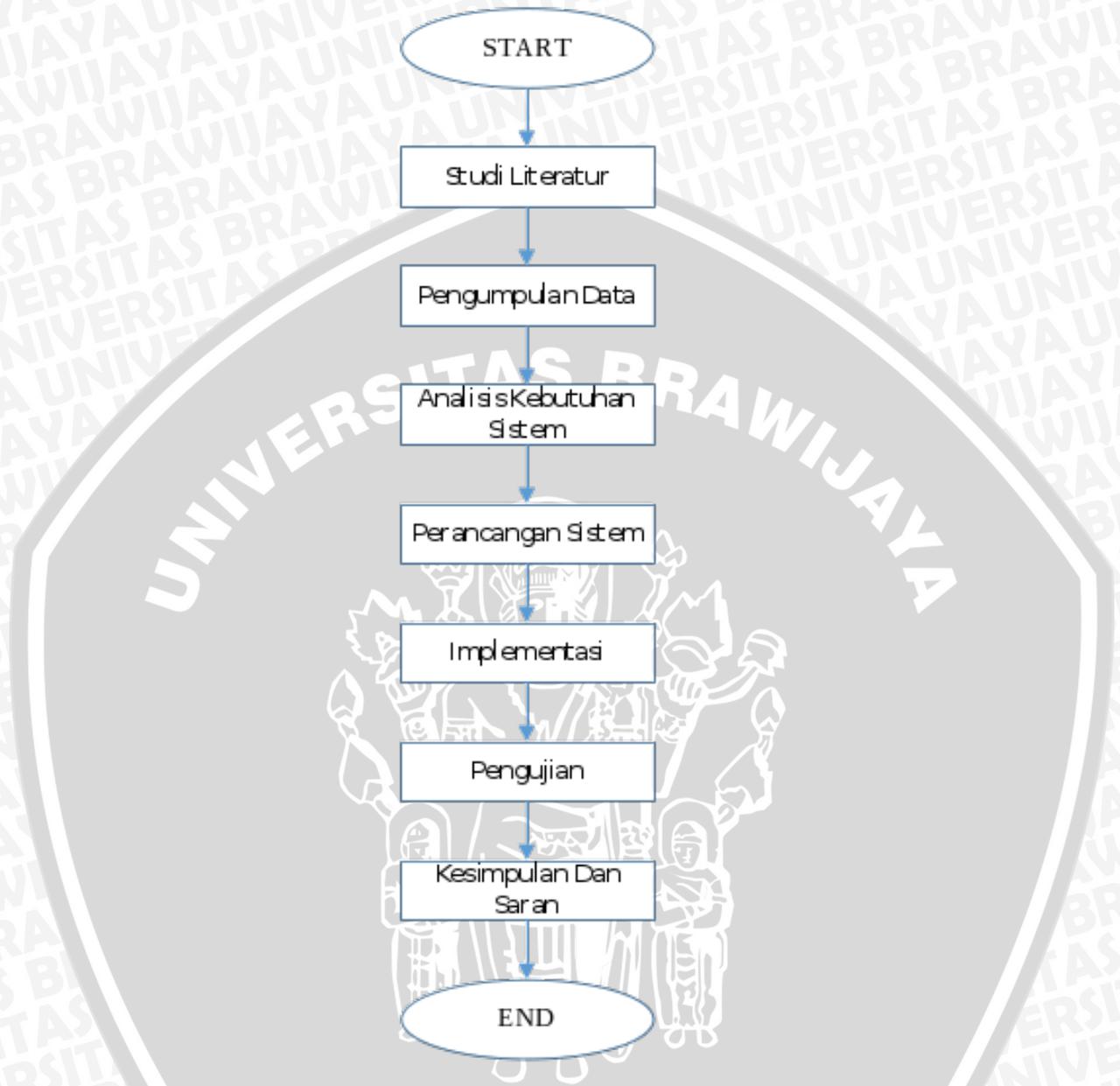


## BAB III

### METODOLOGI PENELITIAN

Metode penelitian membahas metode dan tahapan yang digunakan dalam pengembangan Sistem Informasi Geografis Objek Wisata Kabupaten Bondowoso. Metode metode yang digunakan meliputi studi literatur, analisis kebutuhan sistem, perancangan data, perancangan perangkat lunak, implementasi, pengujian dan pengambilan kesimpulan.





Gambar 3.1 *Flow Chart* Metode Penelitian

### 3.1 Studi Literatur

Studi literatur merupakan penelusuran literatur yang bertujuan dalam menyusun dasar teori yang digunakan untuk menunjang skripsi. Penelusuran literatur dapat bersumber dari buku, media, pakar maupun hasil penelitian orang lain. Teori-teori pendukung tersebut meliputi:

1. Sistem Informasi Geografis (SIG)
2. Sistem Informasi Geografis (SIG) Berbasis Web

2. *Unified Modeling Language* (UML)

3. *LeafletJs*

4. *Entity Relationship Model*

### **3.2 Pengumpulan Data**

Data yang dibutuhkan meliputi informasi wisata yang terdapat di Bondowoso. Data yang digunakan dalam pembuatan aplikasi diperoleh melalui studi pustaka dan observasi.

### **3.3 Analisis Kebutuhan Sistem**

Tujuan dari tahap analisis adalah memahami kebutuhan dari sistem. Analisis kebutuhan merupakan langkah awal untuk menentukan sistem seperti apa yang akan dihasilkan. Sistem yang baik dan sesuai dengan kebutuhan pengguna sangat bergantung kepada keberhasilan dalam analisa kebutuhan.

Untuk mempermudah dalam menganalisis sistem dibutuhkan dua jenis kebutuhan. Dua jenis kebutuhan tersebut adalah kebutuhan fungsional dan kebutuhan nonfungsional.

### **3.4 Perancangan Sistem**

Perancangan sistem perangkat lunak dilakukan setelah semua kebutuhan perangkat lunak didapatkan melalui tahap analisis kebutuhan. Kemudian dimodelkan ke dalam bentuk *Activity Diagram*, *diagram kelas*, *Sequence Diagram*, perancangan basis data, dan perancangan antar muka.

#### **3.4.1 Perancangan *UseCase Diagram***

Sistem dapat dipandang dari dua aspek, yaitu siapa pengguna sistem dan apa kegiatan yang dapat dilakukan dalam sistem. Pengguna dari sistem ini adalah pengunjung dan administrator yang ingin mencari dan mendapatkan informasi mengenai wisata di Bondowosos. Kegiatan yang dapat dilakukan dalam sistem yaitu:

1. Pengunjung mencari dan mendapatkan informasi wisata dari sistem.
2. Administrator dapat melakukan pengolahan data, dan pemeliharaan sistem.

#### **3.4.2 Perancangan *Activity Diagram***

Alur kerja sistem secara berurutan digambarkan melalui *activity diagram*. *Activity Diagram* bermanfaat untuk menganalisis *usecase* melalui penggambaran

aktivitas-aktivitas yang dibutuhkan, penggambaran algoritma berurutan yang kompleks, dan pemodelan aplikasi dengan proses paralel.

#### **3.4.3 Perancangan *Sequence Diagram***

Perancangan interaksi antar elemen (objek) dalam sistem yang telah teridentifikasi akan dimodelkan dalam *Sequence Diagram* yang menggambarkan interaksi antar objek yang disusun dalam urutan waktu. *Sequence Diagram* berupa skenario atau rangkaian langkah-langkah yang dilakukan dari sebuah kejadian untuk menghasilkan output tertentu.

#### **3.4.4 Perancangan *Class Diagram***

Perancangan kelas dan *interface* dalam sistem dimodelkan dalam *class diagram*. Diagram kelas memberi gambaran (diagram statis) tentang sistem informasi ini, relasi-relasi yang terdapat di dalamnya, dan komponen yang digunakan. Diagram kelas juga digunakan untuk mengidentifikasi kelas-kelas serta paket-paket yang terdapat dalam sistem. Kelas-kelas yang telah teridentifikasi kemudian mengidentifikasi hubungan antar kelas.

#### **3.4.5 Perancangan Basis Data**

Perancangan basis data terdiri dari dua, yaitu perancangan basis data tingkat logik dan perancangan basis data tingkat fisik. Perancangan basis data secara logik dimulai dengan penciptaan model konseptual dari sistem. Model konsep ini kemudian dipetakan menjadi model data secara logik yang telah dipengaruhi model data target basis data. Model data secara logik merupakan sumber informasi perancangan fisik.

#### **3.4.6 Perancangan *User Interface***

Kebutuhan-kebutuhan desain antarmuka untuk pembuatan sistem informasi ini sebagai berikut:

1. Sistem ini akan dibangun dengan tampilan yang mudah diterima oleh pengguna.
2. Kompatibilitas sistem harus sesuai dengan browser.

### 3.5 Implementasi Sistem

Implementasi sistem informasi geografis mengenai objek wisata di Bondowoso dilakukan dengan mengacu pada analisis kebutuhan dan perancangan sistem. Implementasi sistem informasi ini meliputi :

1. Pembuatan sistem informasi geografis pemetaan lokasi wisata sesuai dengan perancangan
2. Pembuatan basis data sesuai dengan perancangan basis data.
3. Data-data tempat wisata dan hotel yang telah diperoleh akan dimasukkan ke dalam basis data.
4. Pembuatan antarmuka sistem dibangun sesuai dengan perancangan antarmuka.

### 3.6 Pengujian Sistem

Pengujian sistem pada penelitian ini dilakukan agar dapat menunjukkan bahwa sistem telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan sistem yang telah melandasinya. Pengujian dilakukan dengan *Usability Teting*, *BlackBox Testing*.

### 3.7 Kesimpulan Dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan implementasi dan pengujian telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis setelah pembuatan aplikasi. Kemudian saran sebagai masukan untuk menambahkan fitur yang dirasa kurang pada aplikasi.

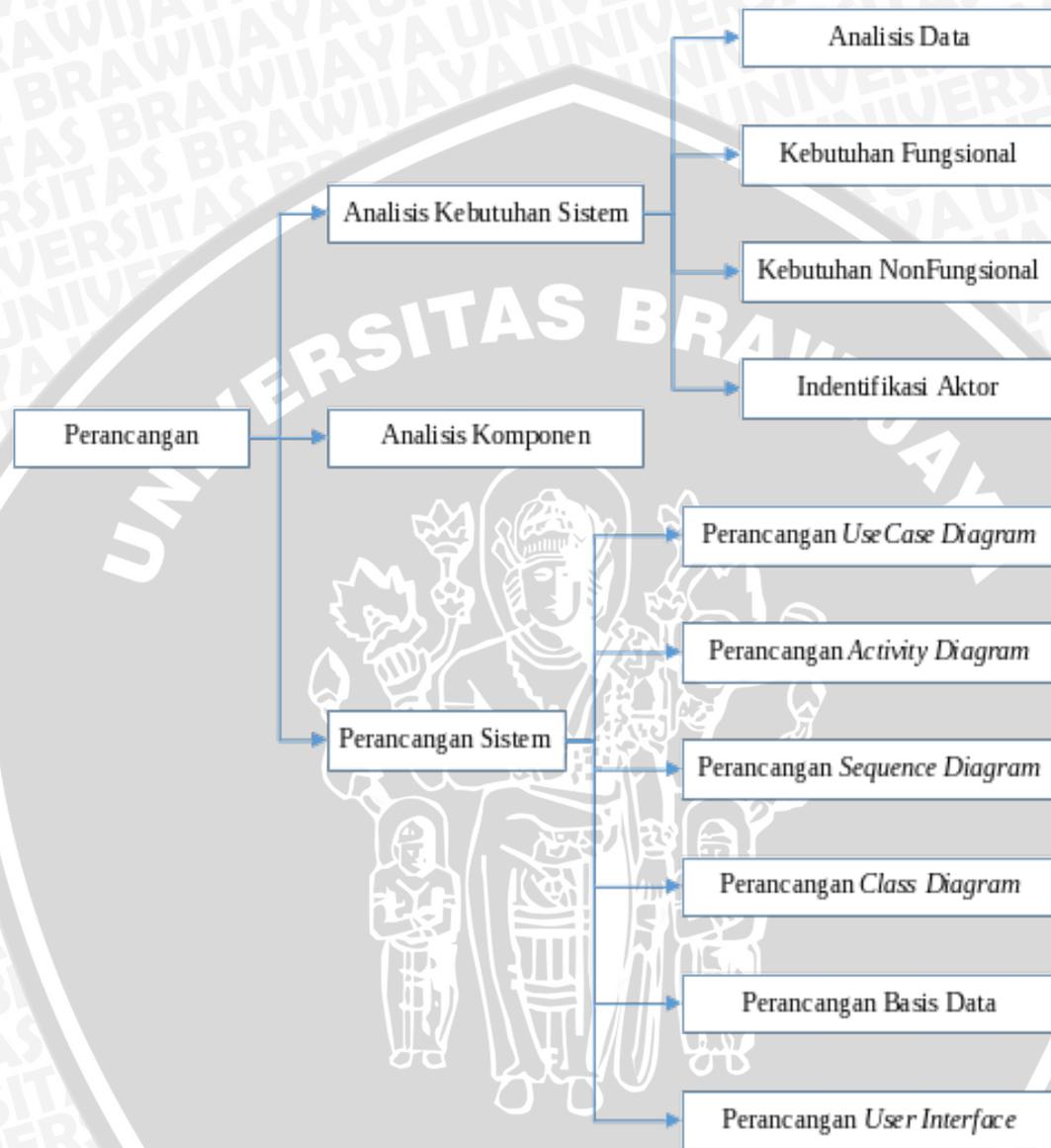
## BAB IV

### ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai analisis kebutuhan dan perancangan perangkat lunak Sistem Informasi Geografis (GIS) Mengenai Objek Wisata Di



Kabupaten Bondowoso. Perancangan yang dilakukan meliputi dua tahap yaitu proses analisis kebutuhan kemudian proses perancangan perangkat lunak.



Gambar 4.1 Pohon Perancangan

#### 4.1 Analisis Kebutuhan Sistem

Analisis kebutuhan Sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang di harapkan sehingga dapat diusulkan perbaikannya.

#### 4.1.1 Analisis Data

Analisis data bertujuan untuk mengetahui struktur penyimpanan data yang dibutuhkan perangkat lunak tersebut. Struktur data penyimpanan data disusun berdasarkan analisis data sebagai berikut:

1. Data admin terdiri dari username dan password.
2. Data Kecamatan Di Kabupaten Bondowoso.
3. Data lokasi tiap tempat wisata data berupa koordinat atau data spasial.
4. Data wisata yang berada di Kabupaten Bondowoso.
5. Data atribut setiap wisata yang berada di Kabupaten Bondowoso.

#### 4.1.2 Kebutuhan Fungsional

Kebutuhan fungsional sama artinya dengan layanan yang harus disediakan, dan bagaimana reaksi sistem terhadap input dan apa yang harus dilakukan sistem pada situasi tertentu. Berikut ini adalah kebutuhan sistem yang dilihat dari sudut pandang pengguna.

ID	Kebutuhan	Aktor
FR_01_01	Pengunjung dapat melihat peta dalam bentuk digital	Pengunjung
FR_01_02	Sistem dapat menentukan lokasi dari aktor	Pengunjung
FR_01_03	Sistem dapat menentukan rute menuju objek wisata	Pengunjung
FR_01_04	Sistem dapat memberikan rating	Pengunjung
FR_01_05	Sistem dapat melakukan komentar	Pengunjung
FR_02_01	Sistem dapat melakukan login	Administrator
FR_02_02	Sistem dapat melakukan logout	Administrator
FR_02_03	Sistem dapat menambahkan data kecamatan	Administrator
FR_02_04	Sistem dapat mengubah data kecamatan	Administrator
FR_02_05	Sistem dapat menghapus data kecamatan	Administrator
FR_02_06	Sistem dapat menambahkan data wisata	Administrator
FR_02_07	Sistem dapat mengubah data wisata	Administrator

FR_02_08	Sistem dapat menghapus data wisata	Administrator
FR_02_09	Sistem dapat menambahkan data hotel	Administrator
FR_02_010	Sistem dapat mengubah data hotel	Administrator
FR_02_011	Sistem dapat menghapus data hotel	Administrator

Tabel 4.1 Daftar Kebutuhan Fungsional

#### 4.1.3 Kebutuhan NonFungsional

Selain kebutuhan fungsional, kebutuhan non fungsional juga disediakan oleh sistem. Daftar spesifikasi kebutuhan nonfungsional menggunakan *usability* diuji menggunakan kuesioner, sampel dipilih secara acak. Berikut kebutuhan Nonfungsional dijabarkan pada table berikut:

Parameter	Deskripsi Kebutuhan
<i>Usability</i>	Perangkat lunak dapat digunakan dengan mudah oleh pengguna

Tabel 4.2 Kebutuhan NonFungsional

#### 4.1.4 Identifikasi Aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor-aktor yang akan berinteraksi dengan Sistem Informasi Geografis (GIS) Mengenai Wisata Di Kabupaten Bondowoso. Tabel 4.3 menunjukkan aktor beserta deskripsinya yang merupakan hasil dari proses identifikasi aktor

Aktor	Deskripsi
Administrator	Administrator merupakan aktor yang mengoperasikan dan memelihara Sistem Informasi Geografis Mengenao Onjek Wisata Di Kabupaten Bondowoso
Pengunjung	Pengunjung merupakan aktor pengguna yang menggunakan aplikasi tanpa melalui proses autentifikasi dahulu, mencari rute wisata, mencari lokasi wisata dan menentukan lokasi terkini dari pengunjung

Tabel 4.3 Identifikasi Aktor

## 4.2 Analisis Komponen

Komponen-komponen yang digunakan dalam Sistem Informasi Geografis Objek Wisata Di Kabupaten Bondowoso adalah sebagai berikut:

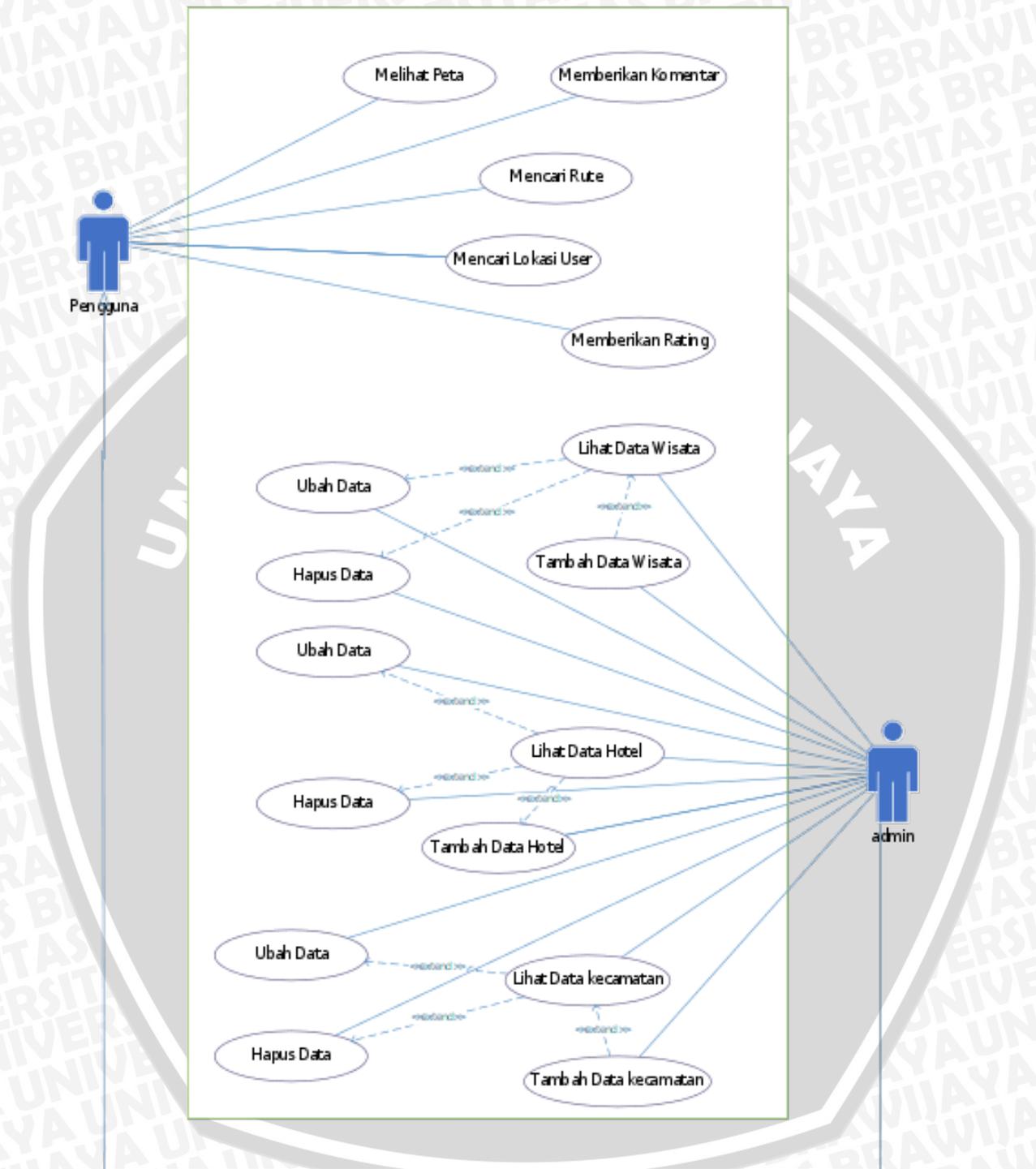
1. Codeigniter  
Codeigniter framework digunakan dalam mengimplementasikan keseluruhan Sistem Informasi Geografis Mengenai Objek Wisata di Kabupaten Bondowoso.
2. Library *LeafletJS*  
*leafletJS* digunakan untuk mengimplementasikan sistem informasi geografis pada peta.
3. MySQL  
MySQL berfungsi sebagai media penyimpanan data.

## 4.3 Perancangan Sistem

Perancangan sistem bertujuan untuk merancang komponen-komponen yang telah diperoleh sesuai dengan spesifikasi kebutuhan. Perancangan tersebut mencakup penambahan maupun perubahan pada komponen-komponen yang telah ditentukan. Perancangan dengan komponen terdiri dari enam tahapan, antara lain: perancangan use case perancangan aktivitas, perancangan kelas, perancangan interaksi, perancangan basis data, dan perancangan antarmuka.

### 4.3.2 Perancangan *UseCase Diagram*

Skenario *usecase* akan diberikan uraian nama *usecase*, aktor yang berhubungan dengan *usecase* tersebut, tujuan dari *usecase*, deskripsi global tentang *usecase*, kondisi awal yang harus dipenuhi, dan kondisi akhir yang diharapkan setelah berjalannya fungsional *usecase*. Selain itu juga akan diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas satu aksi yang diberikan oleh aktor. Gambar 4.2 merupakan hasil analisis *usecase*.



Gambar 4.2 Class Diagram

Secara lebih detail, use case pada gambar di atas akan di jelaskan pada tabel di bawah ini :



### A. Skenario Pengguna

1. *UseCase* Melihat Peta : Pada Skenario ini akan dijelaskan pengguna dapat melihat peta wisata Bondowoso.

Skenario Melihat Peta	
Nama	Melihat Peta
Tujuan	Sistem dapat menampilkan peta wisata dalam bentuk digital
Deskripsi	<i>Use case</i> ini menjelaskan tentang user dapat masuk ke halaman map dan melihat peta wisata Bondowoso.
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Pengguna memasuki halaman Home dari sistem
Main Flow	User
	Pengguna memilih menu wisata
Main Flow	Sistem
	Sistem menampilkan halaman peta wisata
Kondisi Akhir	Sistem Menampilkan Halaman map

Tabel 4.4 Skenario *UseCase* Melihat Peta

2. *UseCase* Mencari Lokasi User : Pada Skenario ini akan dijelaskan pengguna dapat mengetahui posisi terkini dari user.

Skenario Mencari Lokasi User	
Nama	Mencari Lokasi User
Tujuan	Sistem dapat menampilkan posisi pengguna
Deskripsi	<i>Use case</i> ini menjelaskan tentang user dapat mengetahui lokasi dimana user berada
Aktor	Pengguna
Skenario Utama	
Kondisi Awal	Pengguna memasuki halaman map dari sistem
Main Flow	User
	User memilih fitur find me pada peta
Main Flow	Sistem
	Sistem menampilkan posisi terkini dari user
Kondisi Akhir	Sistem dapat menampilkan posisi terkini dari user dengan marker

Tabel 4.5 Skenario *UseCase* Mencari Lokasi User

3. Skenario Mencari Rute : Pada Skenario ini akan dijelaskan pengguna mencari rute menuju lokasi objek wisata dan hotel

Skenario Mencari Rute
-----------------------

Nama	Skenario Mencari Rute	
Tujuan	Sistem dapat menampilkan rute menuju lokasi wisata atau hotel	
Deskripsi	Use case ini menjelaskan tentang user mencari rute menuju lokasi objek wisata dan hotel	
Aktor	Pengguna	
Skenario Utama		
Kondisi Awal	Pengguna memasuki halaman peta wisata	
Main Flow	User	Sistem
	User memilih fitur find me dan mengklik objek wisata yang diuju	Menampilkan garis dari posisi user menuju lokasi wisata atau hotel
Kondisi Akhir	Sistem dapat menampilkan rute menuju lokasi wisata atau hotel	

Tabel 4.6 Skenario UseCase Mencari Rute

4. Skenario Memberi Rating : Pada Skenario ini akan dijelaskan pengguna memberikan rating pada tempat wisata.

Skenario Memberi Rating		
Nama	Skenario Mencari Rute	
Tujuan	Sistem dapat memberi rating pada tempat wisata	
Deskripsi	Use case ini menjelaskan tentang user memberikan rating pada tempat wisata	
Aktor	Pengguna	
Skenario Utama		
Kondisi Awal	Pengguna memasuki halaman peta wisata	
Main Flow	User	Sistem
	- User memilih keterangan wisata	- Sistem menampilkan halaman wisata
	- User menekan tombol rating	- Sistem memberikan rating
Kondisi Akhir	Sistem dapat memaberikan rating	

Tabel 4.7 Skenario UseCase Memberi Rating

5. Skenario Memberi Komentar : Pada Skenario ini akan dijelaskan pengguna mencari rute menuju lokasi objek wisata dan hotel

Skenario Mencari Rute	
Nama	Skenario Mencari Rute
Tujuan	Sistem dapat menampilkan rute menuju lokasi wisata atau

	hotel	
Deskripsi	Use case ini menjelaskan tentang user mencari rute menuju lokasi obbjek wisata dan hotel	
Aktor	Pengguna	
Skenario Utama		
Kondisi Awal	Pengguna memasuki halaman peta wisata	
Main Flow	User	Sistem
	-pengguna mengklklik wisata -user memberikan komentar	-Menampilkan halaman wisata -sistem menyimpan komentar
Kondisi Akhir	Menampilkan komentar info wisata	

Tabel 4. 8 Skenario UseCase Memberi Komentar

B. Skenario Admin

1. Tambah Data Kecamatan : Pada Skenario ini akan menjelaskan tentang fungsi admin untuk melakukan tambah data kecamatan.

Skenario Tambah Data Kecamatan		
Nama	Tambah Data Kecamatan	
Tujuan	Sistem melakukan penambahan data kecamatan	
Deskripsi	Use case ini menjelaskan tentang administrator dapat melakukan proses penambahan data kecamatan	
Aktor	Administrator	
Skenario Utama		
Kondisi Awal	Administrator telah melalui proses autentikasi pada sistem dan memasuki halaman kecamatan	
Main Flow	Admin	Reaksi Sistem
	Admin menambahkan data kecamatan baru	Sistem menampilkan <i>form input</i> data akun kecamatan baru dan menyimpan pada sistem
Kondisi Akhir	Menampilkan Halaman Kecamatan	
Alternative Flow	Jika data error sistem meminta memasukan data yang valid	

Tabel 4.9 Skenario UseCase Tambah Data Kecamatan

2. Ubah Data Kecamatan : Pada Skenario ini akan menjelaskan tentang fungsi admin untuk melakukan ubah data kecamatan.

Skenario Ubah Data Kecamatan		
Nama	Ubah Data Kecamatan	
Tujuan	Sistem mengubah data kecamatan	
Deskripsi	Use case ini menjelaskan tentang administrator dapat melakukan proses mengubah data kecamatan	
Aktor	Administrator	

Skenario Utama	
Kondisi Awal	Administrator telah melalui proses autentikasi pada sistem dan memasuki halaman kecamatan
Main Flow	Admin
	Administrator mengubah data kecamatan yang telah dipilih
	Sistem
	Sistem menampilkan <i>form input</i> data kecamatan lalu menyimpan
Kondisi Akhir	Menampilkan Halaman Kecamatan
Alternative Flow	Jika data error sistem meminta memasukan data yang valid

Tabel 4.10 Skenario *UseCase* Ubah Data Kecamatan

3. Hapus Data Kecamatan : Pada Skenario ini akan menjelaskan tentang fungsi admin untuk melakukan hapus data kecamatan.

Skenario Hapus Data Kecamatan	
Nama	Hapus Data Kecamatan
Tujuan	Sistem melakukan penghapusan data kecamatan
Deskripsi	<i>Use case</i> ini menjelaskan tentang administrator dapat melakukan proses penghapusan data kecamatan
Aktor	Administrator
Skenario Utama	
Kondisi Awal	Administrator telah melalui proses autentikasi pada sistem
Main Flow	Aksi
	Administrator menghapus data kecamatan yang telah dipilih
	Reaksi Sistem
	Sistem melakukan penghapusan data kecamatan dalam sistem sesuai pilihan administrator
Kondisi Akhir	Menampilkan Halaman Kecamatan
Alternative	Jika data error sistem meminta memasukan data yang valid

Tabel 4.11 Skenario *UseCase* Hapus Data Kecamatan

4. Tambah Data Wisata : Pada Skenario ini akan menjelaskan tentang fungsi admin untuk melakukan tambah data wisata.

Skenario Tambah Data Wisata	
Nama	Tambah Data Wisata
Tujuan	Sistem melakukan penambahan data Wisata
Deskripsi	<i>Use case</i> ini menjelaskan tentang administrator dapat melakukan proses penambahan data wisata
Aktor	Administrator
Skenario Utama	
Kondisi Awal	Administrator telah melalui proses autentikasi pada sistem dan memasuki halaman wisata

Main Flow	Aksi	Reaksi Sistem
	Admin menambahkan data wisata baru	Sistem menampilkan <i>form input</i> data akun wisata baru dan menyimpan pada sistem
Kondisi Akhir	Menampilkan Halaman wisata	
Alternative Flow	Jika data error sistem meminta memasukan data yang valid	

Tabel 4.12 Skenario *UseCase* Tambah Data Wisata

5. Ubah Data Wisata : Pada Skenario ini akan menjelaskan tentang fungsi admin untuk melakukan ubah data wisata.

Skenario Ubah Data Wisata		
Nama	Ubah Data Wisata	
Tujuan	Sistem mengubah data wisata	
Deskripsi	<i>Use case</i> ini menjelaskan tentang administrator dapat melakukan proses mengubah data wisata	
Aktor	Administrator	
Skenario Utama		
Kondisi Awal	Administrator telah melalui proses autentikasi pada sistem dan memasuki halaman wisata	
Main Flow	Admin	Sistem
	Administrator mengubah data wisata yang telah dipilih	Sistem menampilkan <i>form input</i> data wisata lalu menyimpan
Kondisi Akhir	Menampilkan Halaman wisata	
Alternative Flow	Jika data error sistem meminta memasukan data yang valid	

Tabel 4.13 Skenario *UseCase* Ubah Data Wisata

6. Hapus Data Wisata : Pada Skenario ini akan menjelaskan tentang fungsi admin untuk melakukan hapus data wisata.

Skenario Hapus Data Wisata		
Nama	Hapus Data Wisata	
Tujuan	Sistem melakukan penghapusan data wisata	
Deskripsi	<i>Use case</i> ini menjelaskan tentang administrator dapat melakukan proses penghapusan data wisata	
Aktor	Administrator	
Skenario Utama		
Kondisi Awal	Administrator telah melalui proses autentikasi pada sistem dan memasuki halaman wisata	
Main Flow	Admin	Sistem
	Administrator menghapus data	Sistem melakukan penghapusan data wisata dalam sistem sesuai

	wisata yang telah dipilih	pilihan administrator
Kondisi Akhir	Menampilkan Halaman wisata	
Alternative Flow	Jika data error sistem meminta memasukan data yang valid	

Tabel 4.14 Skenario UseCase Hapus Data Wisata

13. Tambah Data Hotel : Pada Skenario ini akan menjelaskan tentang fungsi admin untuk melakukan tambah data hotel.

Skenario Tambah Data Hotel		
Nama	Tambah Data Hotel	
Tujuan	Sistem melakukan penambahan data Hotel	
Deskripsi	Use case ini menjelaskan tentang administrator dapat melakukan proses penambahan data Hotel	
Aktor	Administrator	
Skenario Utama		
Kondisi Awal	Administrator telah melalui proses autentikasi pada sistem dan memasuki halaman hotel	
Main Flow	Admin	Sistem
	Admin menambahkan data hotel baru	Sistem menampilkan <i>form input</i> data akun hotel baru dan menyimpan pada sistem
Kondisi Akhir	Menampilkan Halaman hotel	
Alternative Flow	Jika data error sistem meminta memasukan data yang valid	

Tabel 4.15 Skenario UseCase Tambah Data Hotel

14. Ubah Data Hotel : Pada Skenario ini akan menjelaskan tentang fungsi admin untuk melakukan ubah data hotel.

Skenario Ubah Data Hotel		
Nama	Ubah Data Hotel	
Tujuan	Sistem mengubah data hotel	
Deskripsi	Use case ini menjelaskan tentang administrator dapat melakukan proses mengubah data hotel	
Aktor	Administrator	
Skenario Utama		
Kondisi Awal	Administrator telah melalui proses autentikasi pada sistem dan memasuki halaman hotel	
Main Flow	Admin	Sistem
	Administrator mengubah data hotel yang telah	Sistem menampilkan <i>form input</i> data hotel lalu menyimpan

	dipilih
Kondisi Akhir	Menampilkan Halaman hotel
Alternative Flow	Jika data error sistem meminta memasukan data yang valid

Tabel 4.16 Skenario UseCase Ubah Data Hotel

15. Hapus Data Hotel : Pada Skenario ini akan menjelaskan tentang fungsi admin untuk melakukan hapus data hotel.

Skenario Hapus Data Hotel	
Nama	Hapus Data Hotel
Tujuan	Sistem melakukan penghapusan data Hotel
Deskripsi	<i>Use case</i> ini menjelaskan tentang administrator dapat melakukan proses penghapusan data Hotel
Aktor	Administrator
Skenario Utama	
Kondisi Awal	Administrator telah melalui proses autentikasi pada sistem
Main Flow	Admin
	Sistem
	Administrator menghapus data Hotel yang telah dipilih
	Sistem melakukan penghapusan data Hotel dalam sistem sesuai pilihan administrator
Kondisi Akhir	Menampilkan Halaman Hotel
Alternative Flow	Jika data error sistem meminta memasukan data yang valid

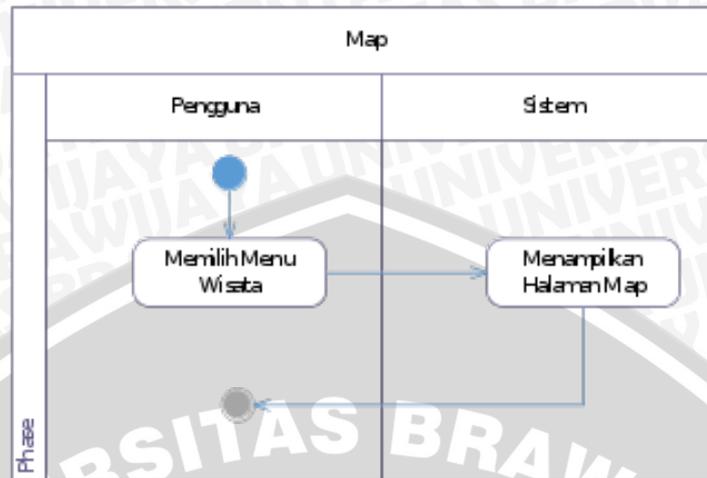
Tabel 4.17 Skenario UseCase Hapus Data Hotel

### 4.3.3 Perancangan Activity Diagram

*Activity Diagram* merupakan diagram yang menggambarkan *workflow* atau aliran kerja atau aktifitas dari sebuah sistem.

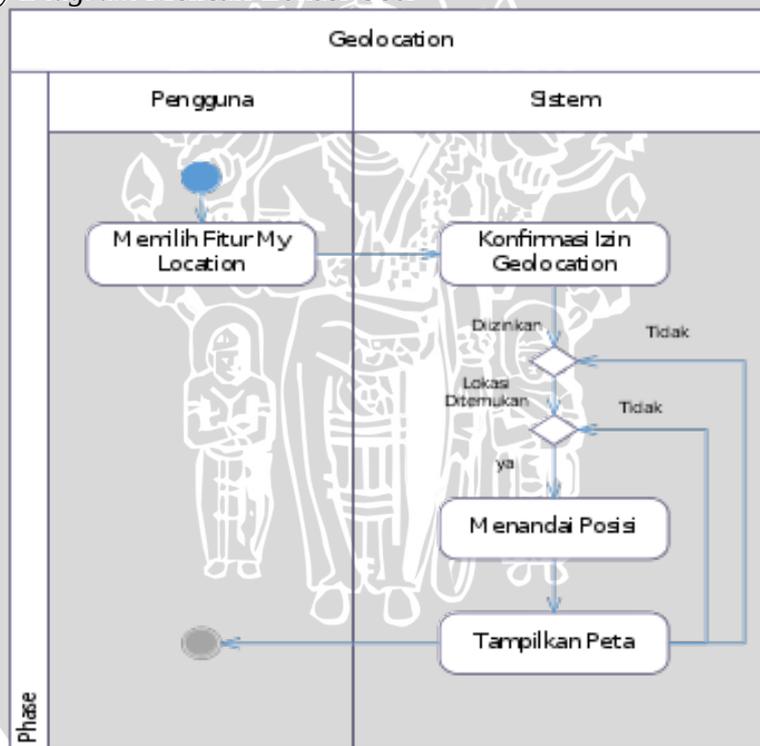
#### 1. Activity Diagram Pengguna Akhir

- a. Activity Diagram Akses Halaman Map



Gambar 4.3 Menampilkan Halaman Peta Wisata

b. Activity Diagram Mencari Lokasi User

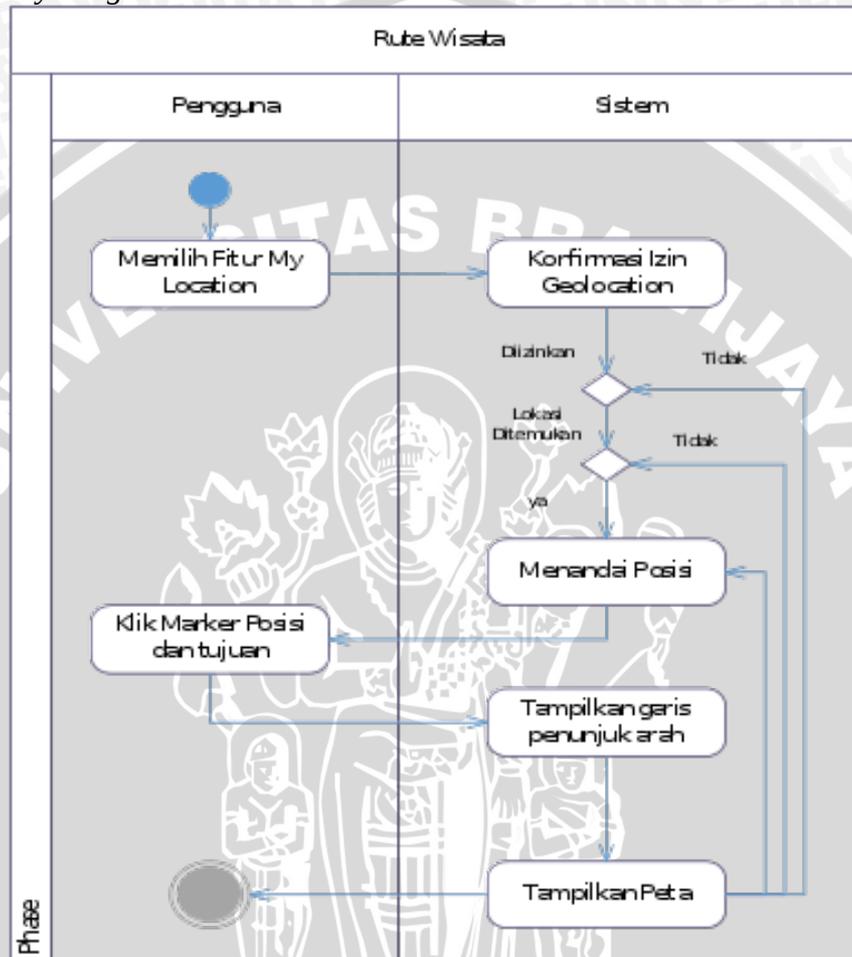


Gambar 4.4 Mencari Lokasi

Salah satu fitur yang ada dalam aplikasi ini adalah fitur geolocation atau mendeteksi keberadaan pengguna. Pertama sistem akan memberikan pertanyaan apakah diijinkan untuk mendeteksi lokasi atau tidak. Jika diijinkan, sistem akan melanjutkan pada tahap pencarian lokasi, kemudian jika

lokasi ditemukan maka sistem akan menambahkan penanda (marker). Jika tidak diijinkan atau pada saat tahap pencarian lokasi tidak ditemukan, maka sistem akan menampilkan peta tanpa penanda.

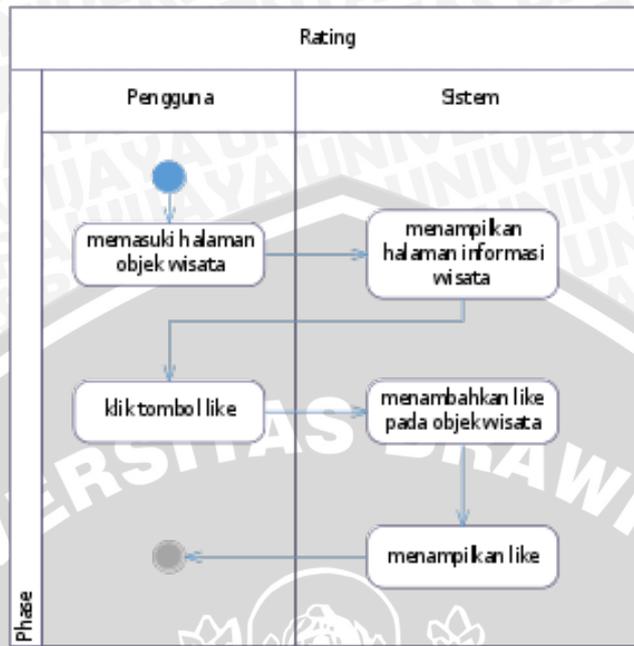
c. Activity Diagram Mencari Rute.



Gambar 4.5 Mendapatkan Rute

Jika kedua lokasi yang dimaksud ditemukan, maka sistem akan menampilkan garis petunjuk. Tetapi, jika salah satu lokasi atau keduanya tidak ditemukan, maka sistem akan menampilkan peta kosong. Setelah itu pengguna dapat memilih menu lainnya.

d. Activity Diagram Pengguna Memberikan Like Pada Tempat Wisata

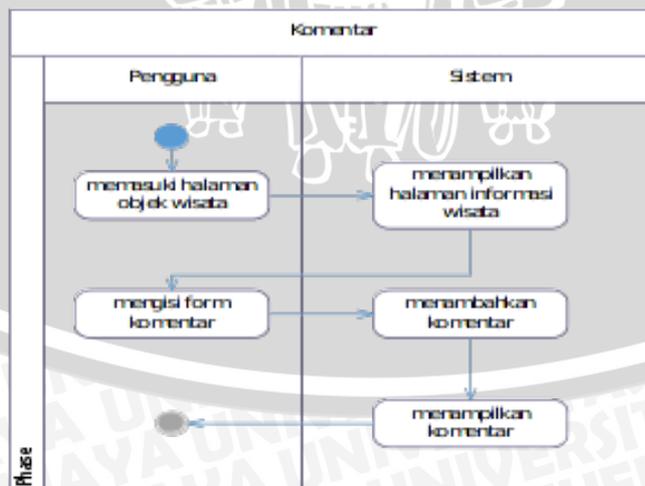


Gambar 4.6 Memberikan Like Pada Tempat Wisata

Keterangan

Untuk menambahkan like user terlebih dahulu memasuki halaman peta wisata, kemudian user mengklik deskripsi wisata. Di dalam deskripsi wisata terdapat tombol like dan dislike. User dapat like atau dislike tempat wisata tersebut.

e. Activity Diagram Pengguna Dapat Memeberikan Komentar Pada Objek Wisata



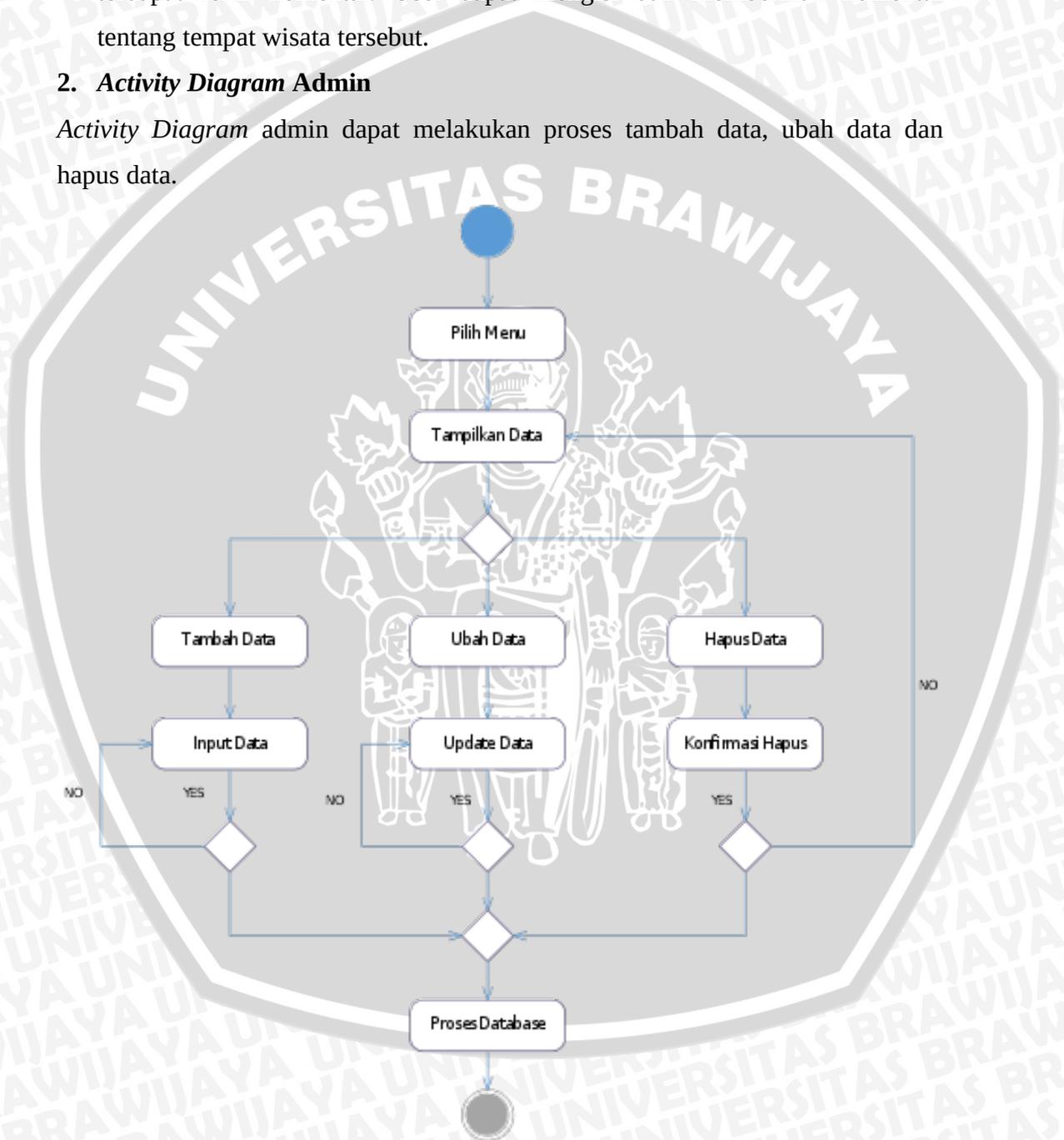
Gambar 4.7 Memeberikan Komentar Pada Objek Wisata

**Keterangan**

Untuk memberikan komentar user terlebih dahulu memasuki halaman peta wisata, kemudian user mengklik deskripsi wisata. Di dalam deskripsi wisata terdapat form komentar. User dapat mengisi dan memberikan komentar tentang tempat wisata tersebut.

**2. Activity Diagram Admin**

Activity Diagram admin dapat melakukan proses tambah data, ubah data dan hapus data.



Gambar 4.8 Proses Tambah, Hapus, Ubah data

Keterangan:

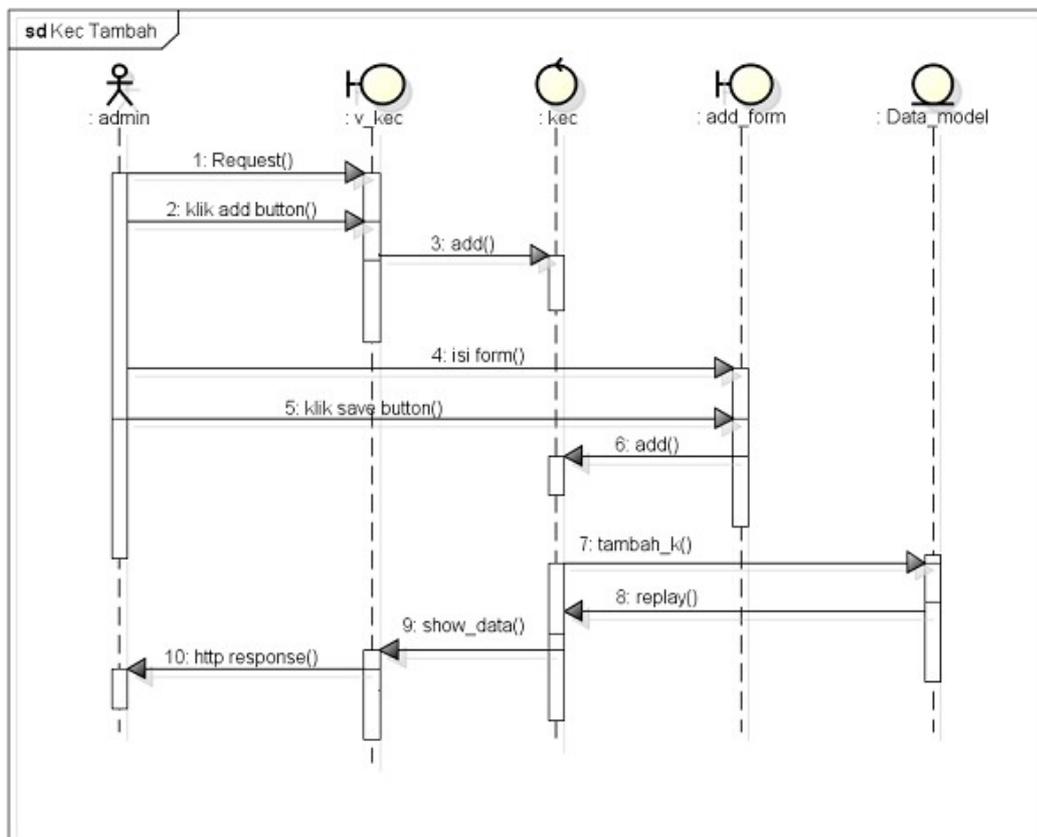
1. Mulai masuk ke sistem aplikasi, dan memilih menu.
2. Sistem secara otomatis menampilkan data berdasarkan menu yang telah dipilih tadi.
3. Di sini admin memilih salah satu dari tambah data, ubah data, atau hapus data.
4. Admin memilih pilihan proses tambah data.
  - o admin diminta untuk menginputkan data wisata baru.
  - o Jika inputan data tidak valid, maka sistem akan mengalihkan user ke halaman tambah data.
  - o Jika data inputan valid, maka sistem akan meneruskan data untuk diproses.
5. Admin memilih pilihan proses ubah data.
  - o Admin mengubah data lama dengan yang baru.
  - o Jika inputan data tidak valid, maka sistem akan mengalihkan user ke halaman ubah data.
  - o Jika data inputan valid, maka sistem akan meneruskan data untuk diproses.
6. Admin memilih pilihan proses hapus data.
  - o Setelah dipilih hapus data, maka otomatis sistem akan menampilkan konfirmasi hapus, di sini admin diminta untuk mengkonfirmasi ulang apakah data akan dihapus atau tidak.
  - o Setelah user mengkonfirmasi hapus data, maka sistem meneruskan konfirmasi untuk diproses.
  - o Jika admin mengkonfirmasi untuk tidak menghapus data, maka sistem akan mengalihkan admin ke halaman tampil data.
7. Data valid diproses ke *database*.
8. Proses selesai

### 4.3.4 Perancangan Sequence Diagram

Sequence Diagram adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek tersebut Berikut Sequence Diagram dari sistem yang akan dibuat :

1. Sequence Diagram Admin Tambah Data Kecamatan

Pada Gambar 4.10 Dapat dilihat skenario tentang mengolah data kecamatan yaitu admin dapat menambah data kecamatan.



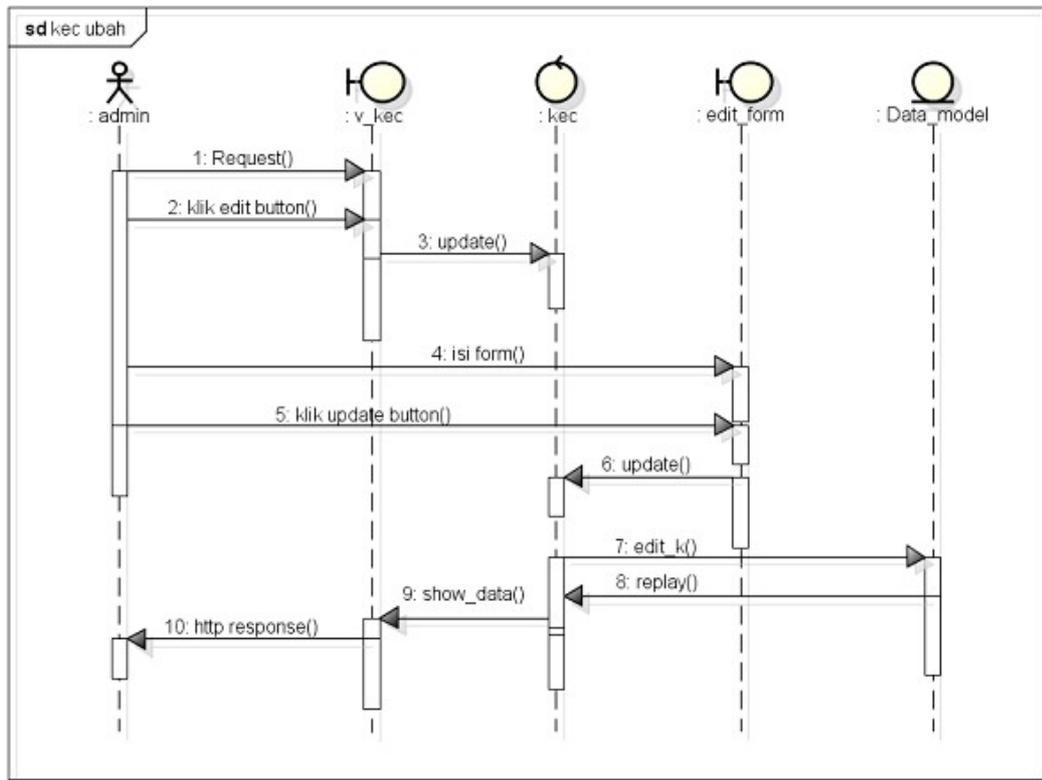
Gambar 4.9 Sequence Diagram Menambahkan Data Kecamatan

Deskripsi :

Setelah login maka admin dapat melakukan manipulasi data salah satunya adalah tambah data. Admin masuk pada view kec. Mengklik tombol add data. Di bawa ke add\_forform. Admin akan menginput data kecamatan, setelah selesai mengisi maka sistem akan akan mengirimkan data tersebut ke database untuk disimpan, dan data di ditampilkan pada v\_kec.

## 2. Sequence Diagram Ubah Data Kecamatan

Pada Gambar 4.11 Dapat dilihat skenario tentang mengolah data kecamatan yaitu admin dapat mengubah data.



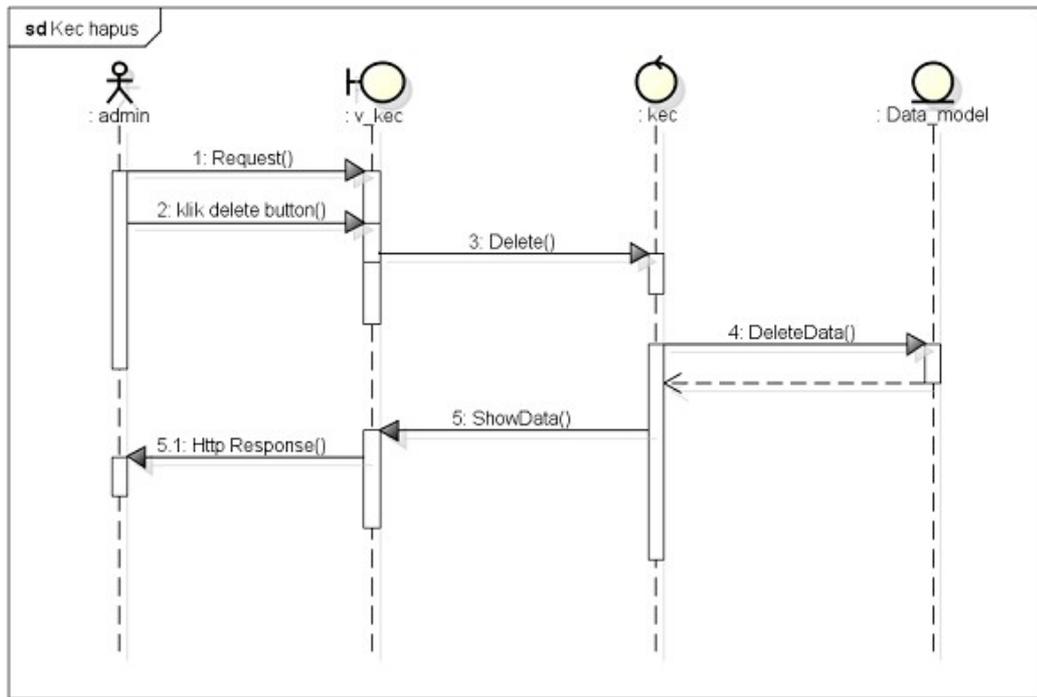
Gambar 4.10 Sequence Diagram mengubah data kecamatan

Deskripsi :

Kondisi edit ini ada jika ada data mengalami perubahan, seperti nama, alamat, info dan lain-lain. Admin akan memilih data yang akan dirubah, kemudian admin akan merubah data lama dengan data baru setelah itu controller akan mengirimkan data yang telah diubah tersebut ke database

### 3. Sequence Diagram Hapus Data Kecamatan

Pada Gambar 4.12 Dapat dilihat skenario tentang mengolah data kecamatan yaitu admin dapat menghapus data.



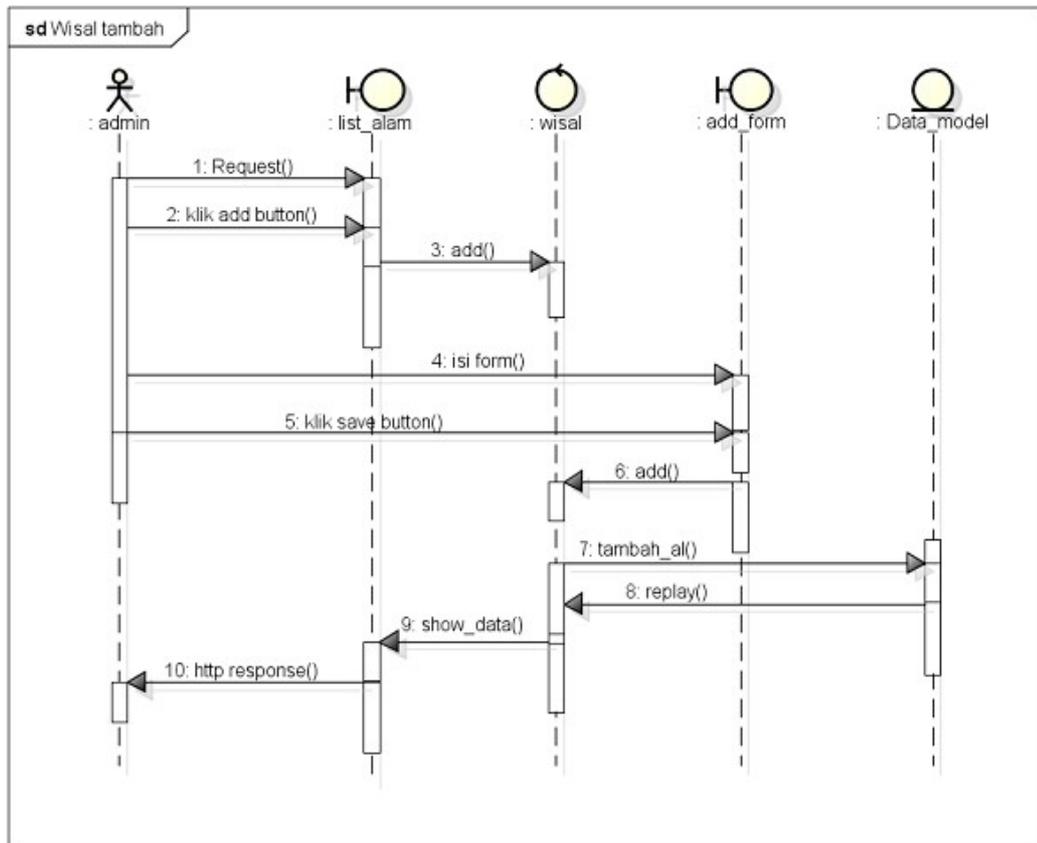
Gambar 4.11 Sequence Diagram menghapus data kecamatan

Deskripsi :

Hal ini terjadi dalam kondisi data kecamatan berubah. Admin akan memilih data yang akan dihapus dari database, kemudian admin melakukan sebuah action untuk menghapus data tersebut maka sistem akan mengirimkan perintah untuk menghapus data yang dipilih dari database.

4. *Sequence Diagram* Tambah Data Wisata

Pada Gambar 4.13 Dapat dilihat skenario tentang mengolah data yaitu admin dapat menambahkan data.



Gambar 4.12 *Sequence Diagram* Menambahkan Data Wisata

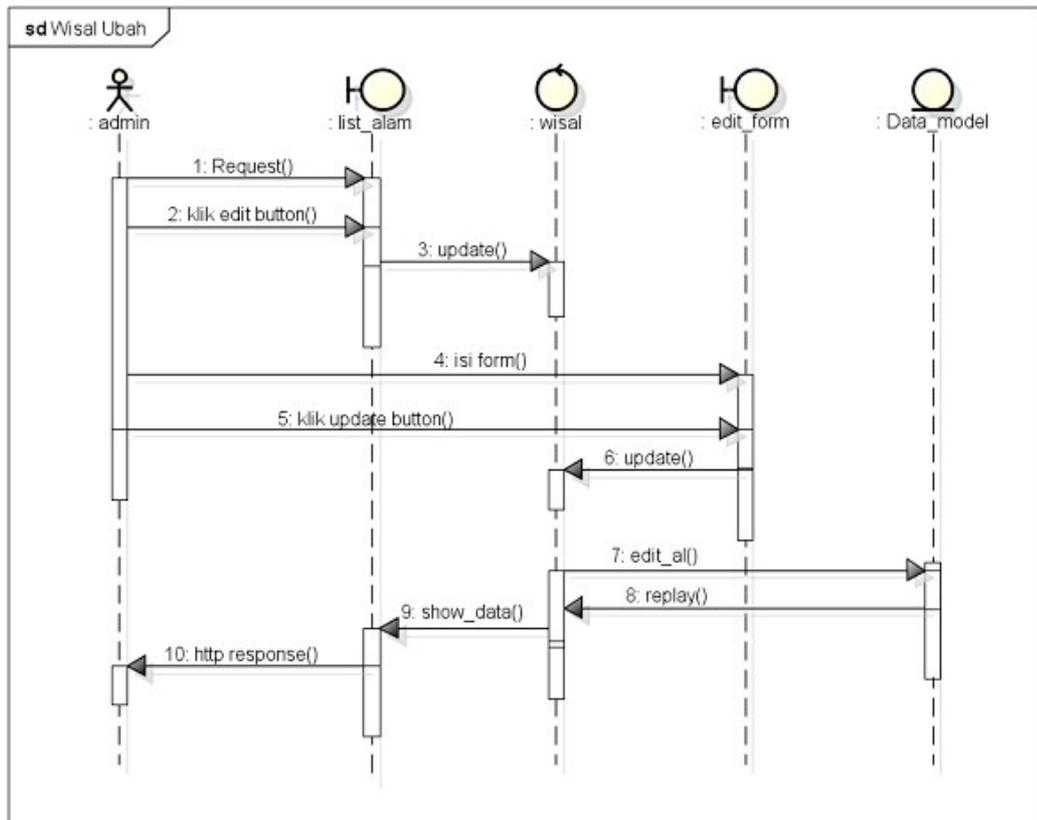
Deskripsi :

Setelah login maka admin dapat melakukan manipulasi data salah satunya adalah tambah data. Admin masuk pada menu wisata alam. Mengklik tombol add data, dan dibawa ke add\_form. Admin akan menginput data wisata, setelah selesai maka sistem akan akan mengirimkan data tersebut ke database untuk disimpan.



5. *Sequence Diagram* Ubah Data Wisata

Pada Gambar 4.14 Dapat dilihat skenario tentang mengolah data yaitu admin dapat mengubah data.



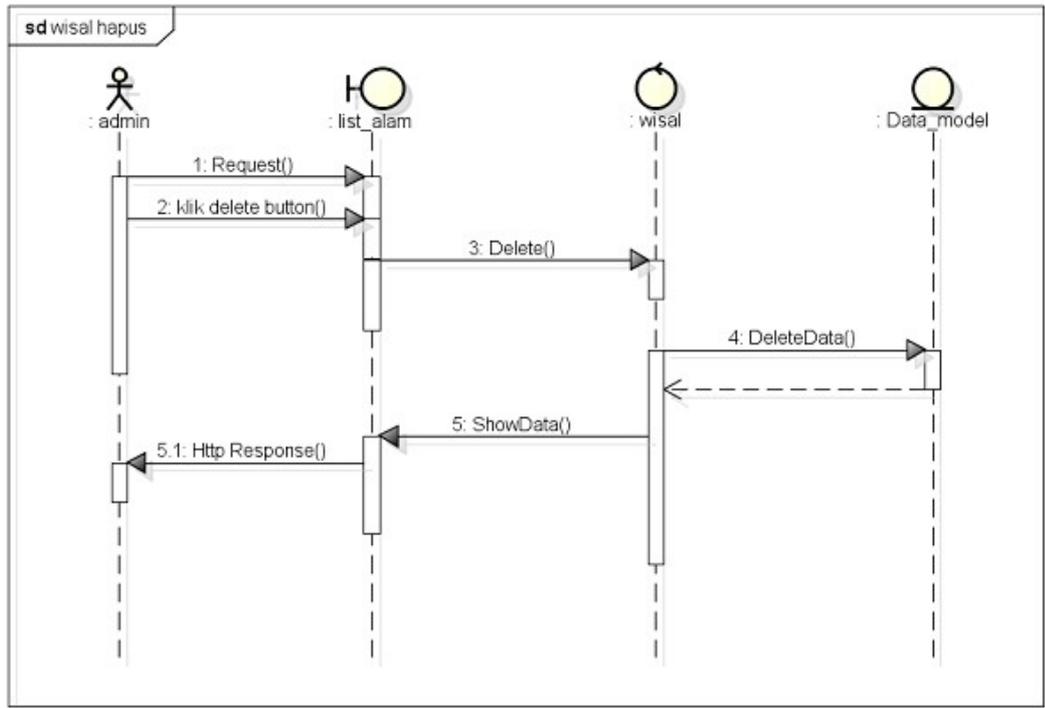
Gambar 4.13 *Sequence Diagram* Mengubah Data Wisata

Deskripsi :

Kondisi edit ini ada jika ada data wisata yang mengalami perubahan, seperti nama, alamat, info dan lain-lain. Admin akan memilih data wisata yang akan dirubah, kemudian admin akan merubah data lama dengan data baru setelah itu controller akan mengirimkan data yang telah diubah tersebut ke database.

6. *Sequence Diagram* Hapus Data Wisata

Pada Gambar 4.15 Dapat dilihat skenario tentang mengolah data yaitu admin dapat menghapus data.



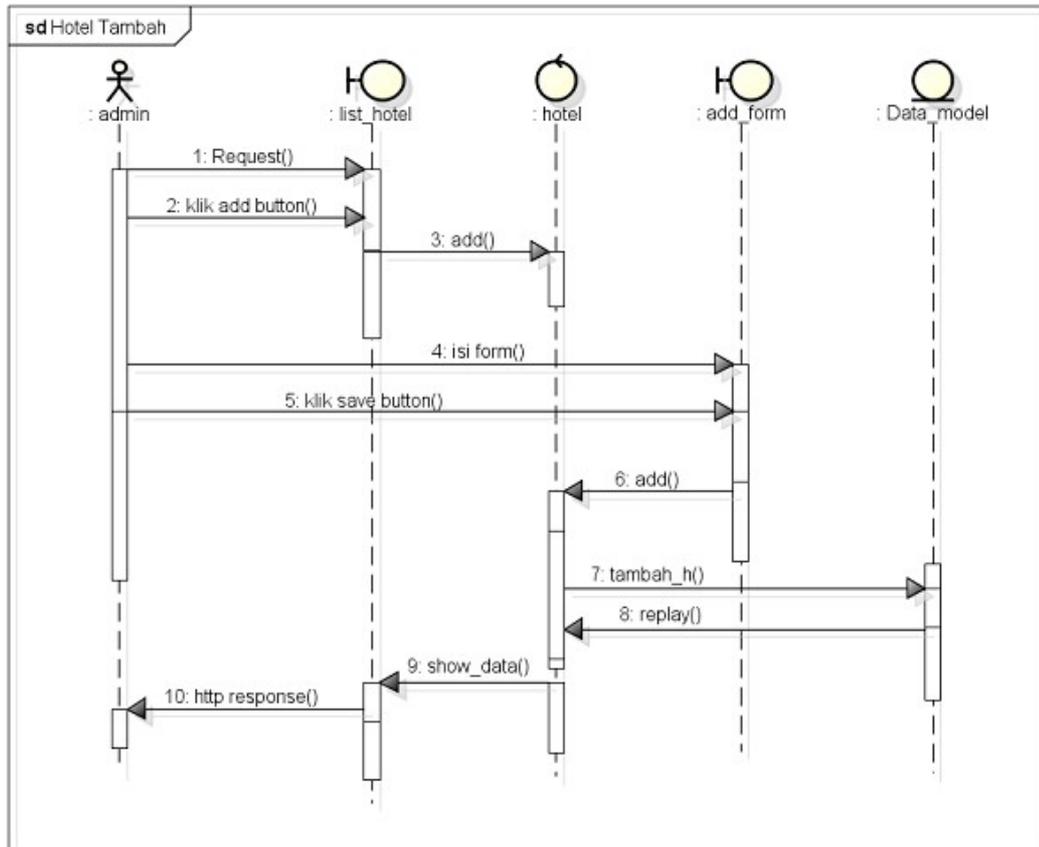
Gambar 4.14 *Sequence Diagram* Menghapus Data Wisata

Deskripsi :

Hal ini terjadi dalam kondisi data wisata berubah, seperti wisata telah di tutup. Admin akan memilih data yang akan dihapus dari database, kemudian admin melakukan sebuah action untuk menghapus data tersebut maka sistem akan mengirimkan perintah untuk menghapus data yang dipilih dari database.

7. *Sequence Diagram* Tambah Data Hotel

Pada Gambar 4.16 Dapat dilihat skenario tentang mengolah data hotel yaitu admin dapat menambah data.



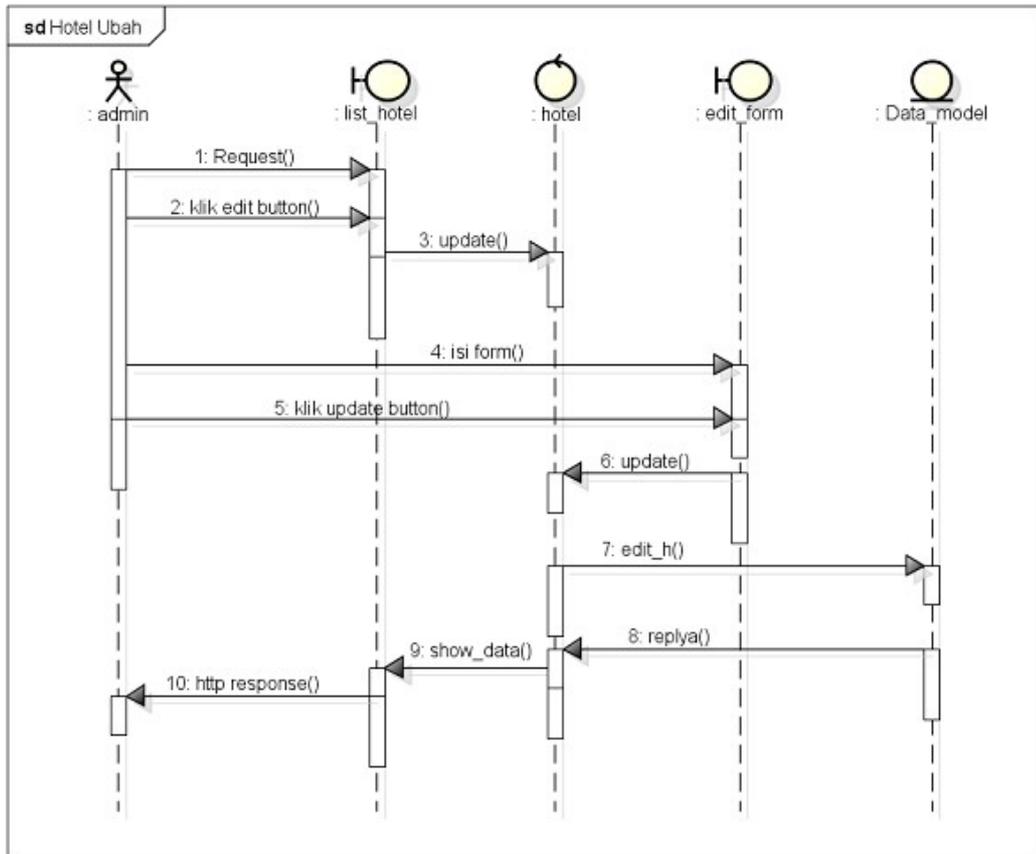
Gambar 4.15 *Sequence Diagram* Menambah Data Hotel

Deskripsi :

Setelah login maka admin dapat melakukan manipulasi data salah satunya adalah tambah data. Admin masuk pada list\_hotel, kemudian Mengklik tombol add data, Admin akan menginput data wisata, setelah selesai maka sistem akan akan mengirimkan data tersebut ke database untuk disimpan.

8. *Sequence Diagram* Ubah Data Hotel

Pada Gambar 4.17 Dapat dilihat skenario tentang mengolah data hotel yaitu admin dapat mengubah informasi data hotel.



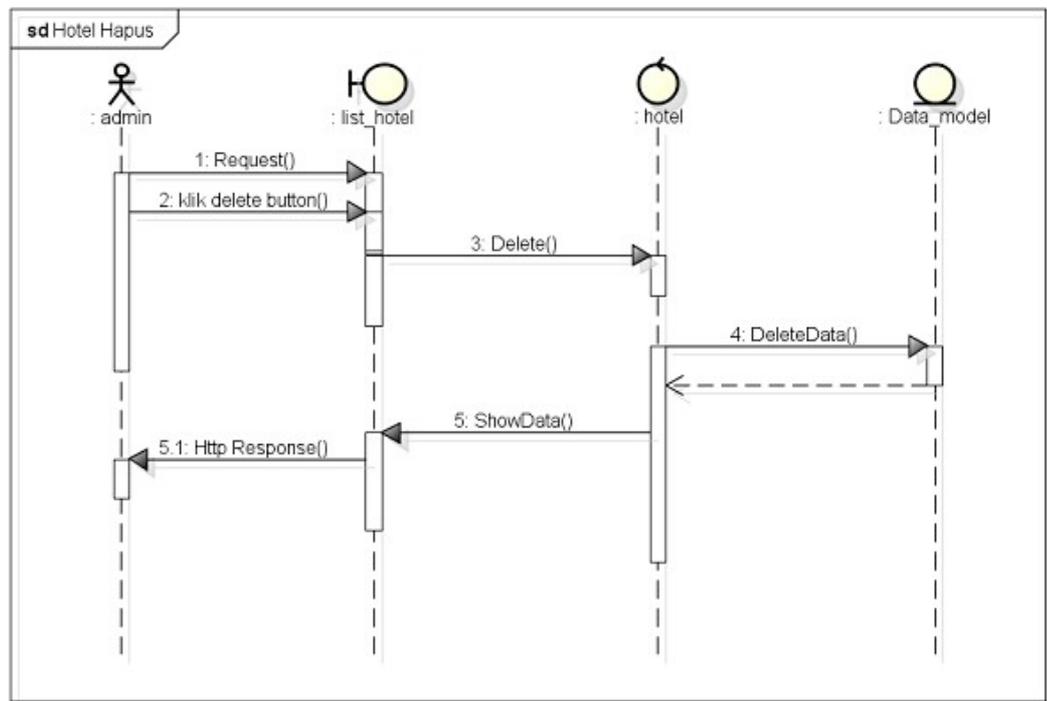
Gambar 4.16 *Sequence Diagram* Mengubah Data Hotel

Deskripsi :

Kondisi ubah data ini ada jika ada data hotel yang mengalami perubahan, seperti nama, alamat, info dan lain-lain. Admin . akan memilih data hotel yang akan dirubah, kemudian admin akan merubah data lama dengan data baru.

9. Sequence Diagram Hapus Data Hotel

Pada Gambar 4.18 Dapat dilihat skenario tentang mengolah data hotel yaitu admin dapat menghapus data hotel.



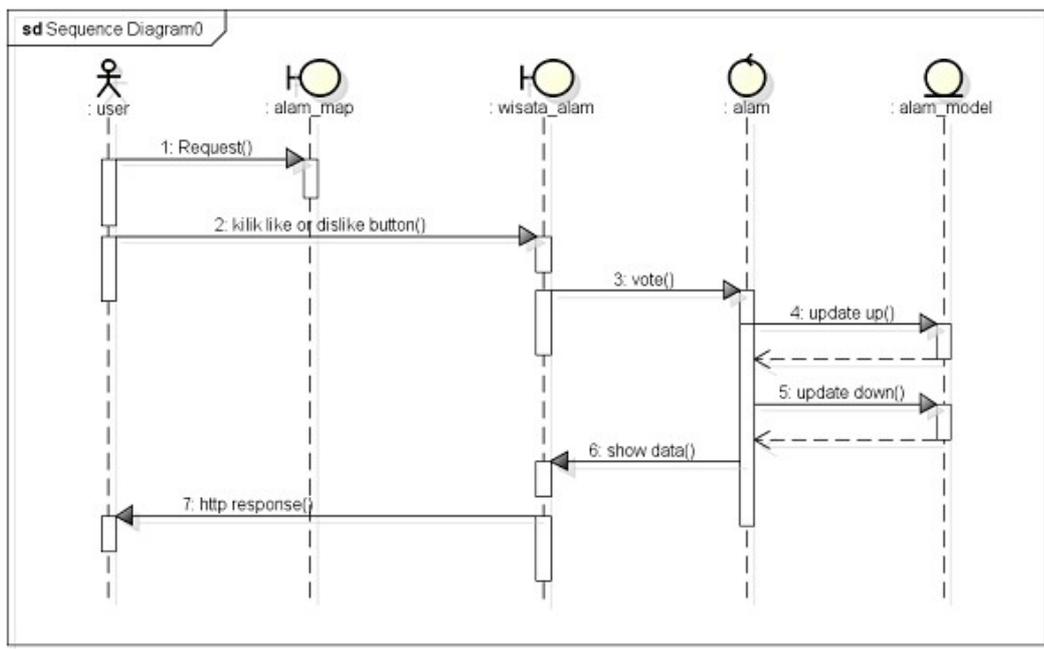
Gambar 4.17 Sequence Diagram Menghapus Data Hotel

Deskripsi :

Hal ini terjadi dalam kondisi data hotel berubah, seperti hotel telah tutup. Admin akan memilih data yang akan dihapus dari database, kemudian admin melakukan sebuah action untuk menghapus data tersebut maka sistem akan mengirimkan perintah untuk menghapus data.



10. *Sequence Diagram* Pengguna Memberikan Rating Pada Objek Wisata  
 Pada Gambar 4.19 Dapat dilihat skenario tentang user dapat memberikan rating pada objek wisata.



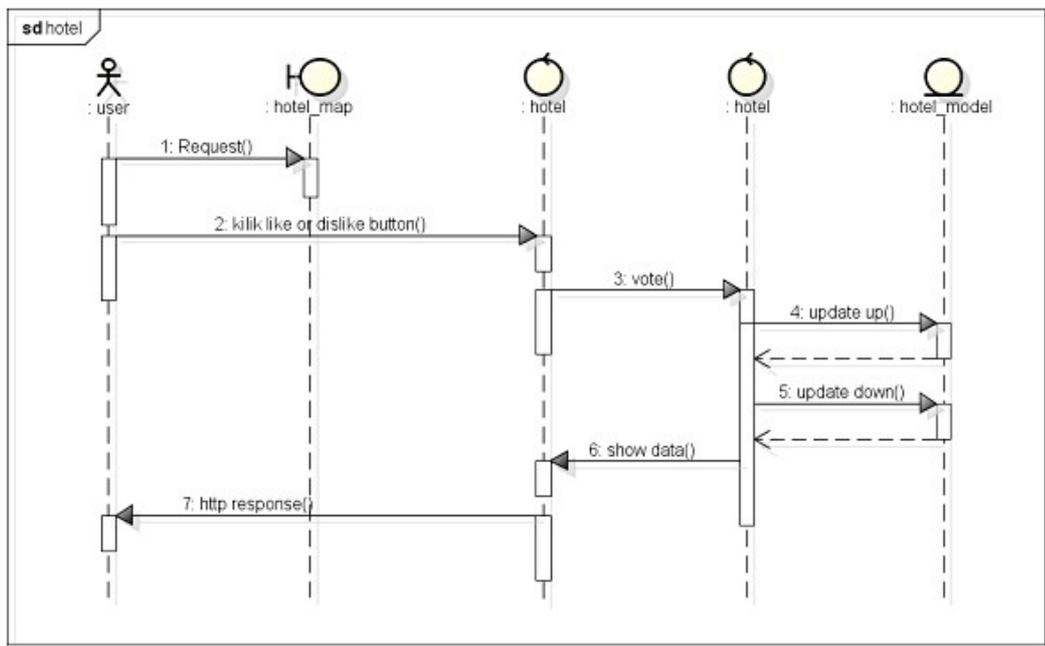
Gambar 4.18 *Sequence Diagram* Memberikan Rating Pada Objek Wisata

Deskripsi :

Sebelum melakukan *like* atau *dislike* tempat wisata. User terlebih dahulu masuk halaman peta wisata. Kemudian melihat list wisata dan memilih read more untuk mengetahui detail wisata. User dapat mengklik tombol *thumbs up* atau *thumbs down*. Data dikirim ke controller menggunakan fungsi *vote* dan menambahkan di model menggunakan fungsi *update up* dan *update down*

11. Sequence Diagram Pengguna Memberikan Rating Pada Hotel

Pada Gambar 4.20 Dapat dilihat skenario tentang pengguna dapat memberikan rating pada hotel.



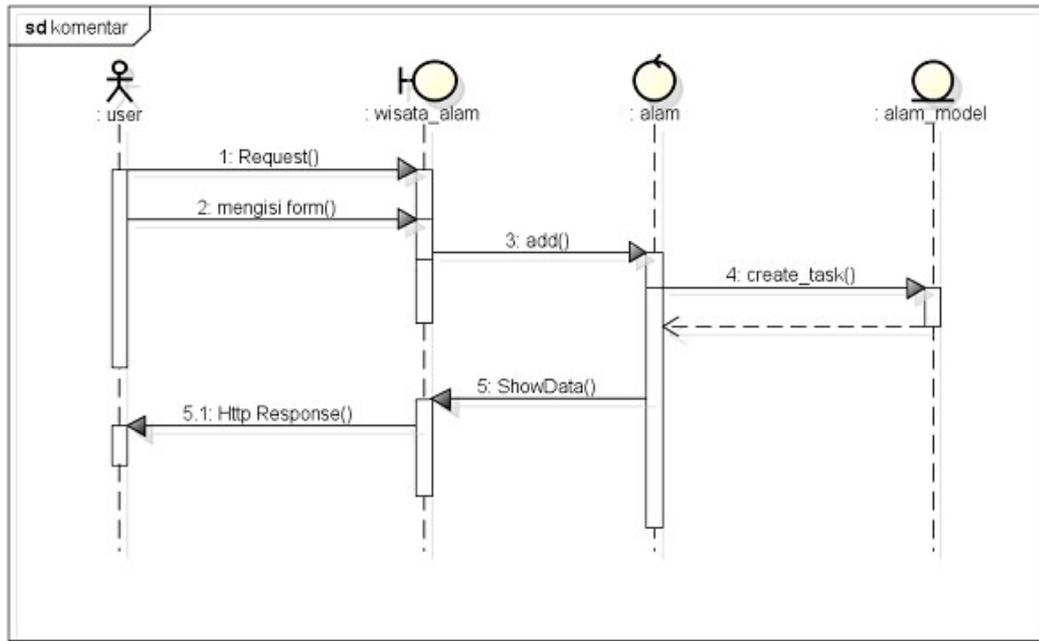
Gambar 4.19 Sequence Diagram Pengguna Memberikan Rating Pada Hotel

Deskripsi :

Sebelum melakukan *like* atau *dislike* tempat wisata. Pengguna terlebih dahulu masuk halaman peta wisata. Kemudian melihat list wisata dan memilih read more untuk mengetahui detail wisata. Pengguna dapat mengklik tombol *thumbs up* atau *thumbs down*. Data dikirim ke controller menggunakan fungsi *vote* dan menambahkan di model menggunakan fungsi *update up* dan *update down*

12. *Sequence Diagram* Pengguna Memberikan Komentar

Pada Gambar 4.21 Dapat dilihat skenario tentang pengguna dapat memberikan komentar pada objek wisata.



Gambar 4.20 *Sequence Diagram* Memberikan Komentar

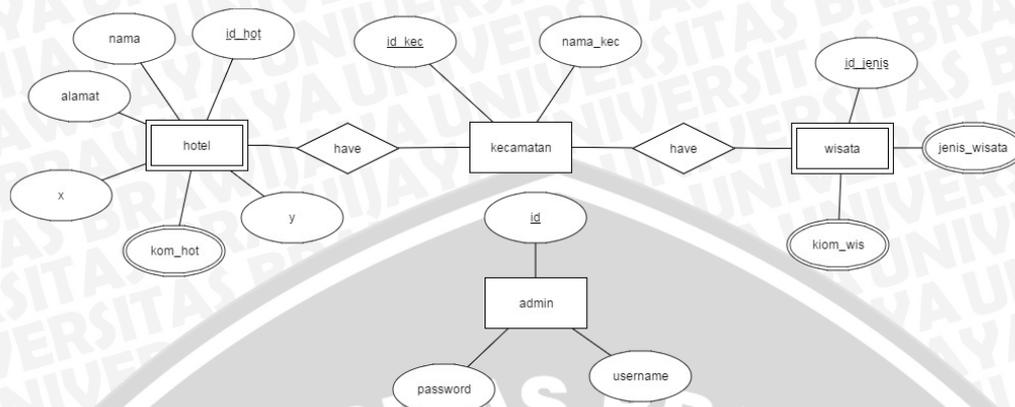
Deskripsi :

Untuk memeberikan komentar pengguna terlebih dahulu memasuki halaman informasi wisata. Kemudian mengisi form untuk memberikan komentar

**4.3.5 Perancangan Kelas**

Pada diagram kelas aplikasi terdapat kelas-kelas yang menyusun kinerja dari aplikasi ini. Kelas-kelas yang telah teridentifikasi dapat memiliki hubungan antar kelas. Kelas-kelas juga dapat memiliki pewarisan dan ketergantungan terhadap komponen-komponen yang telah ditentukan. Adapun aplikasi Sistem Informasi Geografis ini memiliki diagram kelas sebagai berikut:



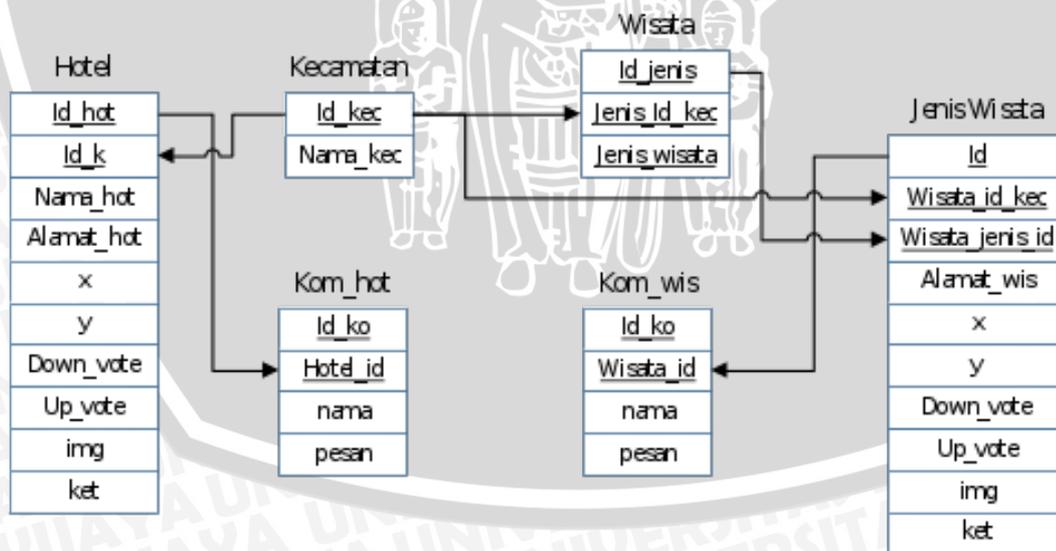


Gambar 4.22 ER Diagram

Tabel yang dibuat terdiri dari :

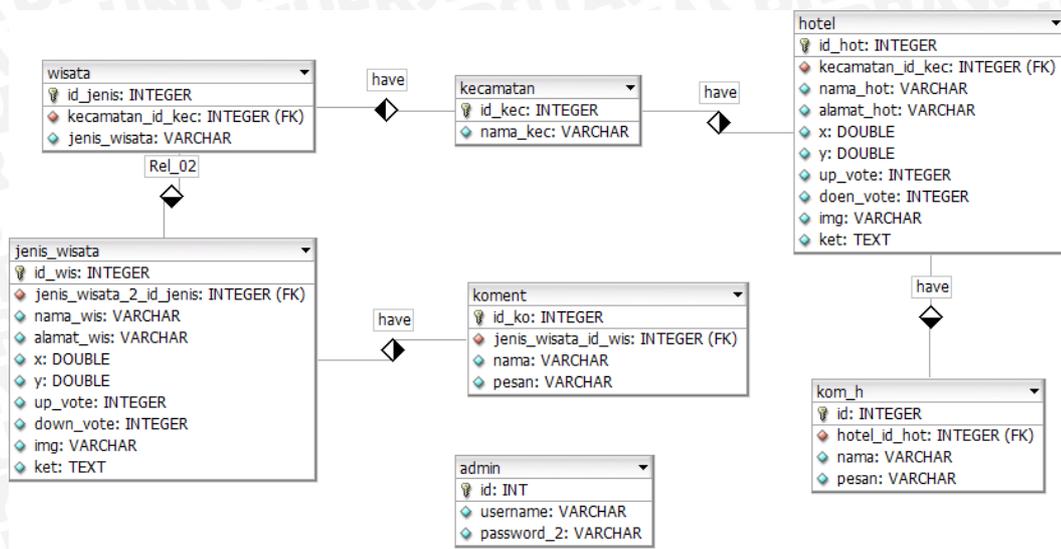
1. Table Kecamatan
2. Tabel Hotel
3. Tabel Wisata
4. Table Admin
5. Table Komentar Wisata
6. Tabel Komentar Hotel

Dari ERD di atas , kita dapat melakukan ERM mapping untuk bisa membuat normalisasi ERD dari setiap tabel , sebagai berikut:



Gambar 4.23 ER Mapping

*Physical Diagram* dirancang berdasarkan rancangan ERD pada gambar 4.23



Gambar 4.24 Physical Diagram

**B. Kamus Data**

Kamus data berisi metadata untuk menjelaskan secara detail catatan semua table di dalam suatu basis data. Katalog sistem adalah kamus data sistem yang sangat detail yang menggambarkan semua objek di dalam suatu basis data.

1. Tabel admin

Field	Tipe Data	Panjang	Keterangan
username	Varchar	20	Primary Key
password	Varchar	100	

Tabel 4.18 Struktur Tabel Admin

2. Tabel Kecamatan

Field	Tipe Data	Panjang	Keterangan
Id_kec	Int	AutoIncrement	Primary Key
Nama_kec	Varchar	255	

Tabel 4. 19 Struktur Tabel Kecamatan

3. Tabel Komentar Wisata

Field	Tipe Data	Panjang	Keterangan
Id_ko	Int	AutoIncrement	Primary Key
Nama	Varchar	255	
Pesan	Text		

Tabel 4.20 Struktur Tabel Kecamatan

4. Tabel Wisata

Field	Tipe Data	Panjang	Keterangan
-------	-----------	---------	------------

Id	Int	AutoIncrement	Primary Key
Nama_wis	Varchar	255	
Alamat_wis	Varchar	255	
x	Double		
y	Double		
Up_vote	Int		
Down_vote	Int		
Img	Varchar		
Ket	Text		

Tabel 4.21 Struktur Tabel Wisata

5. Tabel Hotel

Field	Tipe Data	Panjang	Keterangan
Id	Int	AutoIncrement	Primary Key
Nama	Varchar	255	
Alamat	Varchar	255	
x	Double		
y	Double		
Up_vote	Int		
Down_vote	Int		
Img	Varchar		
Ket	Text		

Tabel 4.22 Struktur Tabel Hotel

6. Tabel Komentar

Field	Tipe Data	Panjang	Keterangan
Id_ko	Int	AutoIncrement	Primary Key
Nama	Varchar	255	
Pesan	Text		

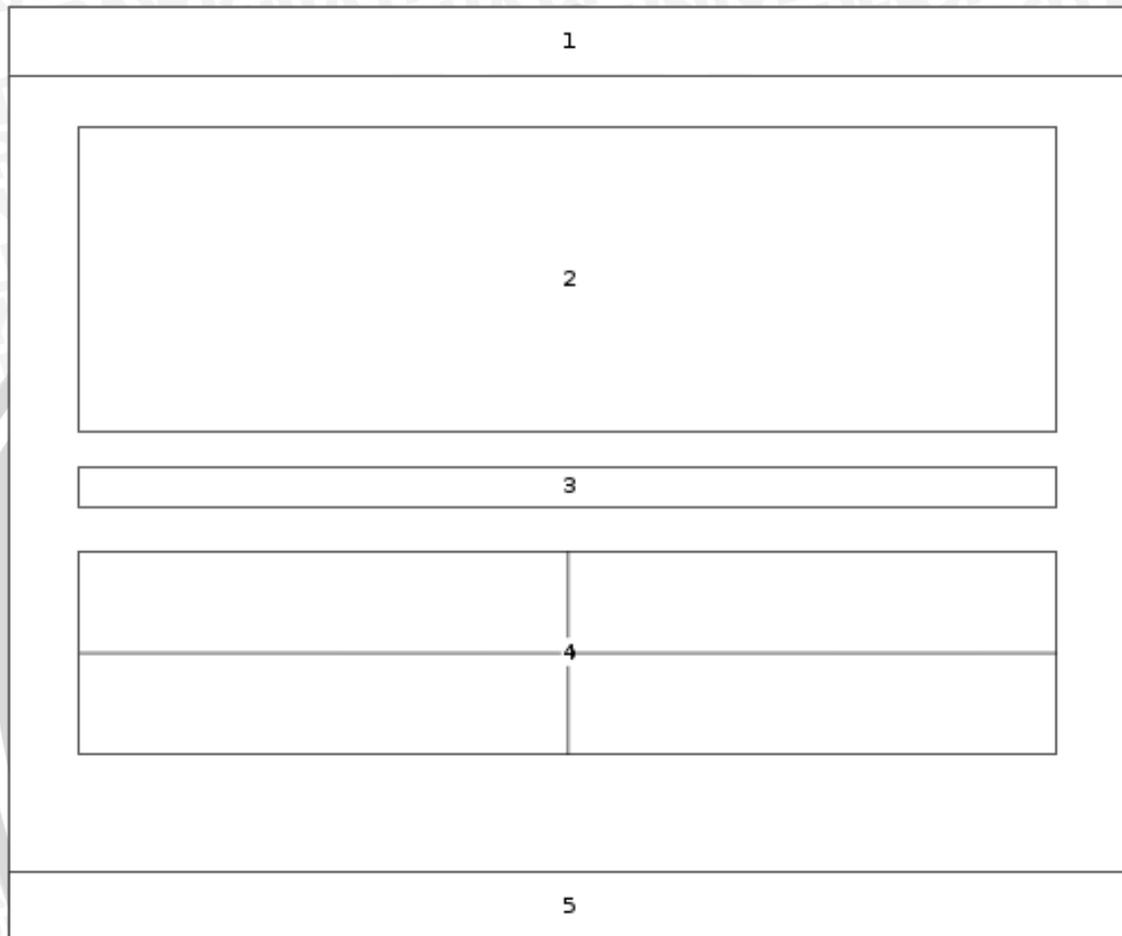
Tabel 4.23 Struktur Tabel Komentar

4.3.7 Perancangan *User Interface*

Perancangan *user interface* digunakan pengguna untuk berkomunikasi dan memberi perintah kepada system yang sedang dioperasikan. Perancangan antar muka system ini dijelaskan pada alur sitemap dan desain antarmuka tiap-tiap

halaman aplikasi. Pada bagian ini akan dijelaskan perancangan *user interface* untuk aplikasi yang ditujukan untuk client.

a. Perancangan Halaman Utama



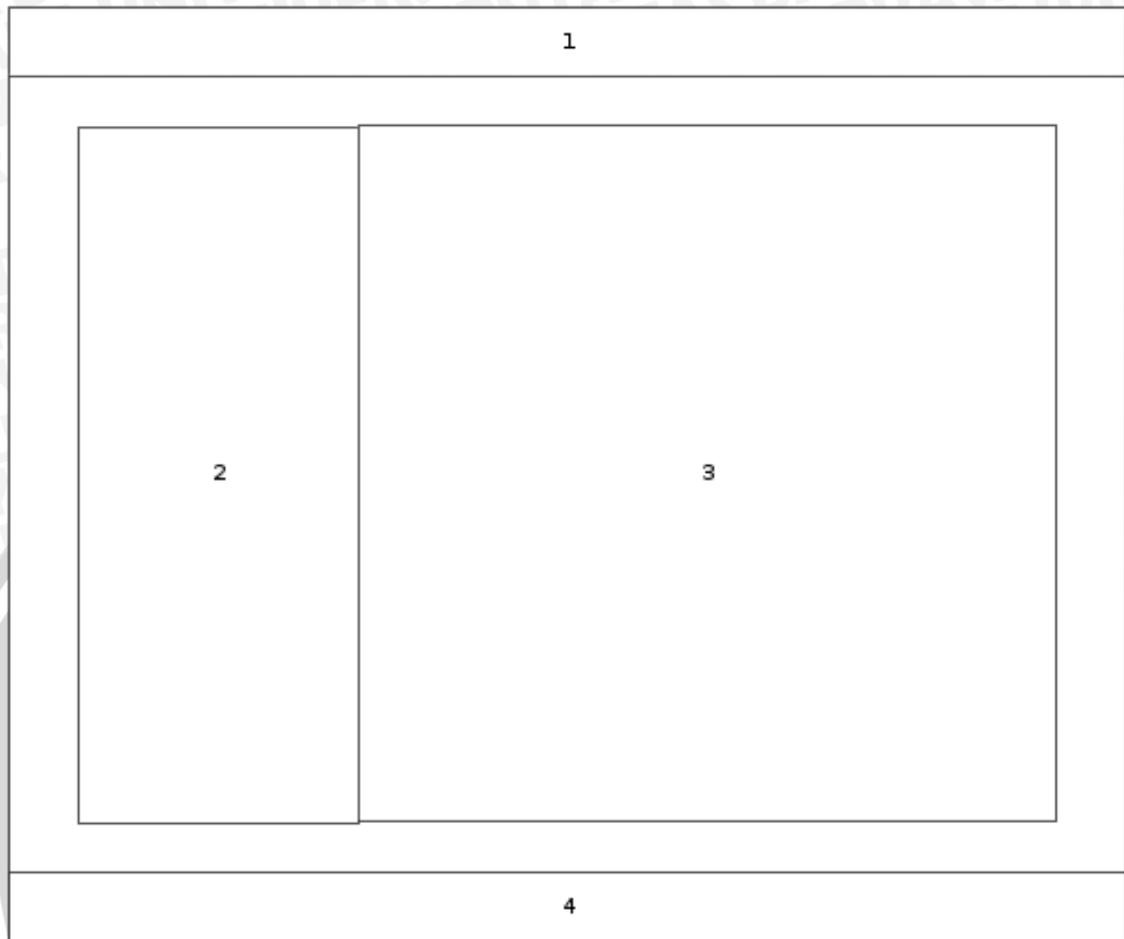
Gambar 4.25 Halaman Utama

Halaman utama menampilkan sekilas profil bondowoso dan tempat wisata

Gambar 4.27 memiliki keterangan sebagai berikut :

1. Navbar berisi tentang pilihan menu
  2. Berisikan image
  3. Berisikan tulisan pembuka
  4. Berisikan pilihan rekomendasi tempat wisata
  5. Footer berisi hak cipta dan tahun.
- b. Perancangan Halaman Map





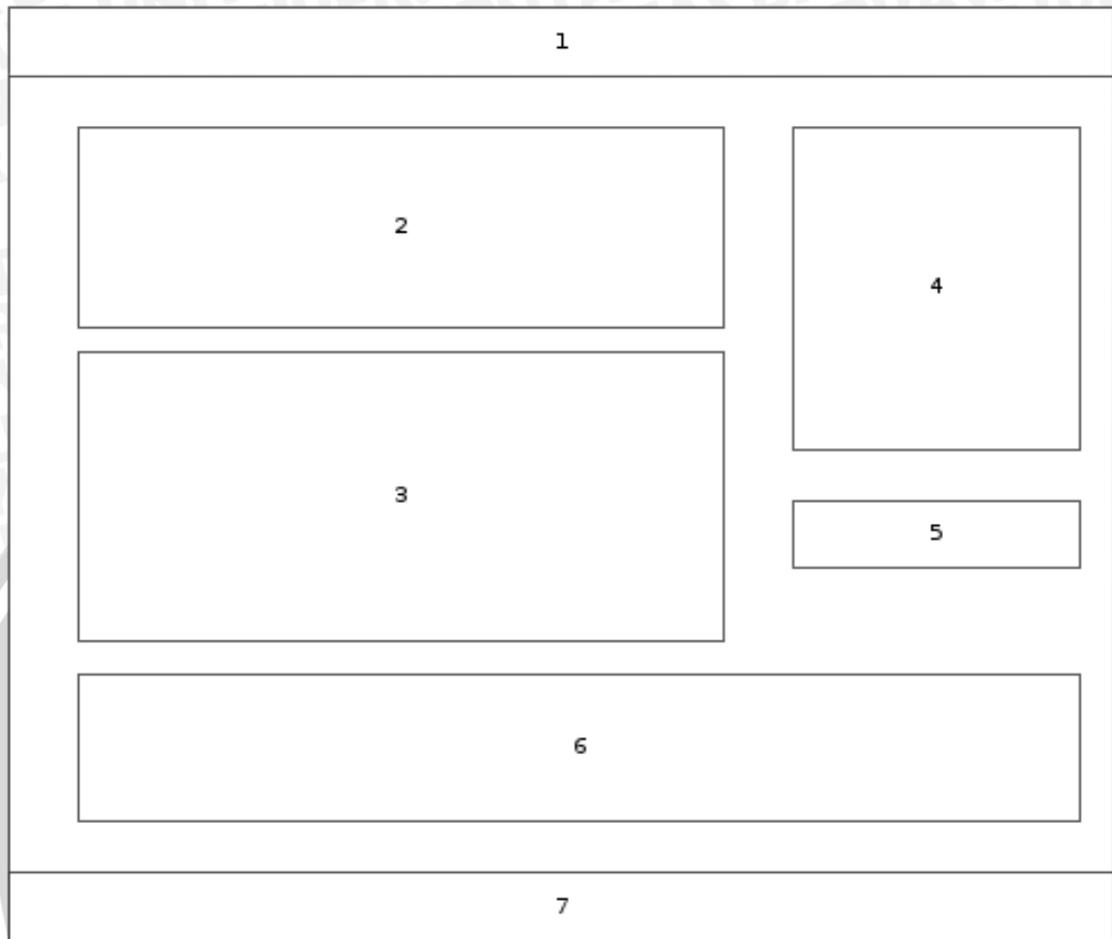
Gambar 4. 26 Halaman Map

Halaman utama menampilkan sekilas profil bondowoso dan tempat wisata

Gambar 4.28 memiliki keterangan sebagai berikut :

1. Navbar berisi tentang pilihan menu
2. Berisikan list wisata bondowoso
3. Berisikan peta wisata bondowoso
4. Footer berisi hak cipta dan tahun

c. Perancangan Halama Review, Rate Dan Komentar Wisata



Gambar 4. 27 Halaman Review Wisata

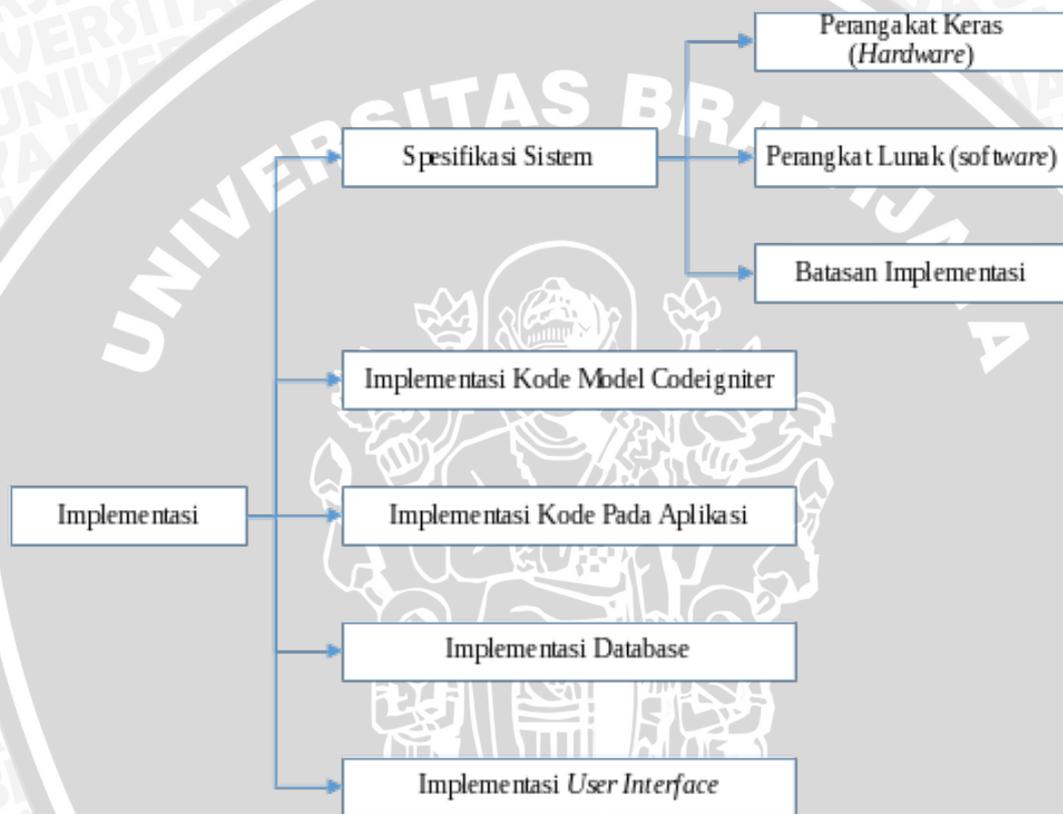
Halaman utama menampilkan sekilas profil bondowoso dan tempat wisata

Gambar 4.29 memiliki keterangan sebagai berikut :

1. Navbar berisi tentang pilihan menu
2. Berisikan image wisata
3. Berisikan informasi wisata
4. Peta wisata
5. Berisikan rate wisata
6. Berisikan komentar wisata
7. Footer berisi hak cipta dan tahun

## BAB V IMPLEMENTASI

Bab ini membahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak yang dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem dan implementasi Adaptasi komponen. Tahap-tahap pembahasan implementasi yang dikerjakan digambarkan pada Gambar 5.1 berikut ini.



Gambar 5.1 Pohon Implementasi

## 5.1 Spesifikasi Sistem

Hasil analisis kebutuhan yang telah diuraikan pada bab 4 menjadi acuan untuk melakukan implementasi menjadi sebuah sistem yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

### 5.1.1 Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras yang digunakan dalam penerapan aplikasi ini sebagai berikut :

Laptop Dell Inspiron 5420	
Processor	Processor
Memory (RAM)	Memory (RAM)
Hardisk	Hardisk
Graphic Card	Graphic Card
Monitor	Monitor

Tabel 5.1 Spesifikasi Perangkat Keras

### 5.1.2 Perangkat Lunak (*Software*)

Spesifikasi perangkat lunak yang digunakan dalam proses pengembangan aplikasi ini sebagai berikut:

Laptop Dell Inspiron 5420	
Operating System	Microsoft windows 7 ultimate 64 bit
DirectX Version	DirectX 11
Programming Language	PHP, HTML, JavaScript
Software Development	Notepad++, XAMP
DBMS	MySQL

Tabel 5.2 Spesifikasi Perangkat Lunak

### 5.1.3 Batasan Implementasi

Beberapa batasan-batasan dalam mengimplementasikan sistem adalah sebagai berikut:

1. Pembuatan aplikasi sistem informasi geografis pemetaan lokasi wisata di kabupaten bondowoso dengan bahasa pemrograman PHP dan javascript
2. Basis data yang dipakai pada sistem ini menggunakan MySQL.
3. Framework yang digunakan adalah codeigniter
4. Library Leafletjs sebagai dasar pembuatan sistem informasi geografis

## 5.2 Implementasi Kode Model Pada Codeigniter

Model mempresentasikan struktur data dari aplikasi yang berupa basis data. Biasanya didalam model berisi kelas yang berfungsi untuk mengambil, mengupdate dan menghapus data. Pada bagian model berisi perintah-perintah yang berhubungan dengan SQL Query.

### 1. Implementasi data\_Model

```
1. function getkec(){
2.     $sql = 'select id_kec, nama_kec from kecamatan';
3.     $query = $this->db->query($sql);
4.     $result = $query->result();
5.     return $result;
6. }
7. function get_kec(){
8.     $this->db->select('id_kec');
9.     $this->db->select('nama_kec');
10.    $this->db->from('kecamatan');
11.    $query = $this->db->get();
12.    $result = $query->result();
13.
14.    $id_kec = array('-SELECT-');
15.    $nama_kec = array('-SELECT-');
16.
17.    for ($i = 0; $i < count($result); $i++)
18.    {
19.        array_push($id_kec, $result[$i]->id_kec);
20.        array_push($nama_kec, $result[$i]->nama_kec);
21.    }
22.    return $kec_result = array_combine($id_kec,
23.    $nama_kec);
24. }
25. function get_jen(){
26.     $this->db->select('jenis_id');
27.     $this->db->select('jenis_wisata');
28.     $this->db->from('wisata');
29.     $query = $this->db->get();
30.     $result = $query->result();
```

```
31.
32.     //array to store department id & department name
33.     $id_jenis = array('-SELECT-');
34.     $jenis_wisata = array('-SELECT-');
35.
36.     for ($i = 0; $i < count($result); $i++)
37.     {
38.         array_push($id_jenis, $result[$i]->jenis_id);
39.         array_push($jenis_wisata, $result[$i]-
40. >jenis_wisata);
41.     }
42.     return $kec_result = array_combine($id_jenis,
43. $jenis_wisata);
44. }
45. public function get_al(){
46.     $query = 'select id, jenis_wisata, nama_kec,
47. nama_wis, alamat_wis, x, y, ket, up_vote, down_vote, img
48. from jenis_wisata
49.         INNER JOIN kecamatan ON kecamatan.id_kec =
50. jenis_wisata.wisata_id_kec
51.         INNER JOIN wisata ON wisata.jenis_id =
52. jenis_wisata.wisata_jenis_id
53.         where jenis_id = 1 ';
54.     $query = $this->db->query($query);
55.     $result = $query->result();
56.     return $result;
57. }
58. function getallim(){
59.     $sql = 'select id, jenis_wisata, nama_kec,
60. nama_wis, alamat_wis, x, y, ket, up_vote, down_vote, img
61. from jenis_wisata
62.         INNER JOIN kecamatan ON kecamatan.id_kec =
63. jenis_wisata.wisata_id_kec
64.         INNER JOIN wisata ON wisata.jenis_id =
65. jenis_wisata.wisata_jenis_id
66.         where jenis_id = 1
67.         ORDER BY jenis_wisata.up_vote DESC
68.         limit 3';
```

```
68.         $query = $this->db->query($sql);
69.         $result = $query->result();
70.         return $result;
71.     }
72.     function getal(){
73.         $sql = 'select id, jenis_wisata, nama_kec,
74. nama_wis, alamat_wis, x, y, ket, up_vote, down_vote, img
75. from jenis_wisata
76.         INNER JOIN kecamatan ON kecamatan.id_kec =
77. jenis_wisata.wisata_id_kec
78.         INNER JOIN wisata ON wisata.jenis_id =
79. jenis_wisata.wisata_jenis_id
80.         where jenis_id = 1
81.         ORDER BY jenis_wisata.up_vote DESC
82.         ';
83.         $query = $this->db->query($sql);
84.         $result = $query->result();
85.         return $result;
86.     }
87.     public function get_bud(){
88.         $query = 'select id, jenis_wisata, nama_kec,
89. nama_wis, alamat_wis, x, y, ket, up_vote, down_vote, img
90. from jenis_wisata
91.         INNER JOIN kecamatan ON kecamatan.id_kec =
92. jenis_wisata.wisata_id_kec
93.         INNER JOIN wisata ON wisata.jenis_id =
94. jenis_wisata.wisata_jenis_id
95.         where jenis_id = 2 ';
96.         $query = $this->db->query($query);
97.         $result = $query->result();
98.         return $result;
99.     }
100.     }
101.     function getbudlim(){
102.         $query = 'select id, jenis_wisata, nama_kec,
103. nama_wis, alamat_wis, x, y, ket, up_vote, down_vote, img
104. from jenis_wisata
105.         INNER JOIN kecamatan ON kecamatan.id_kec =
106. jenis_wisata.wisata_id_kec
```

```
.         INNER JOIN wisata ON wisata.jenis_id =
104 jenis_wisata.wisata_jenis_id
.         where jenis_id = 2
105         ORDER BY jenis_wisata.up_vote DESC
.         limit 3';
106         $query = $this->db->query($query);
.         $result = $query->result();
107         return $result;
.     }
108 function getbud(){
.         $query = 'select id, jenis_wisata, nama_kec,
109 nama_wis, alamat_wis, x, y, ket, up_vote, down_vote, img
.         from jenis_wisata
110         INNER JOIN kecamatan ON kecamatan.id_kec =
.         jenis_wisata.wisata_id_kec
111         INNER JOIN wisata ON wisata.jenis_id =
.         jenis_wisata.wisata_jenis_id
112         where jenis_id = 2
.         ORDER BY jenis_wisata.up_vote DESC
113         ';
.         $query = $this->db->query($query);
114         $result = $query->result();
.         return $result;
115     }
.     public function tambah($data) {
116         return $this->db->insert('jenis_wisata', $data);
.     }
117     public function detail_wisata($id = FALSE) {
.         if ($id === FALSE) {
118             $query = $this->db->get('jenis_wisata');
.             return $query->result_array();
119         }
.         $query = $this->db->get_where('jenis_wisata',
120 array('id' => $id));
.         return $query->row_array();
121     }
.     public function edit($data) {
122         $this->db->where('id', $data['id']);
```

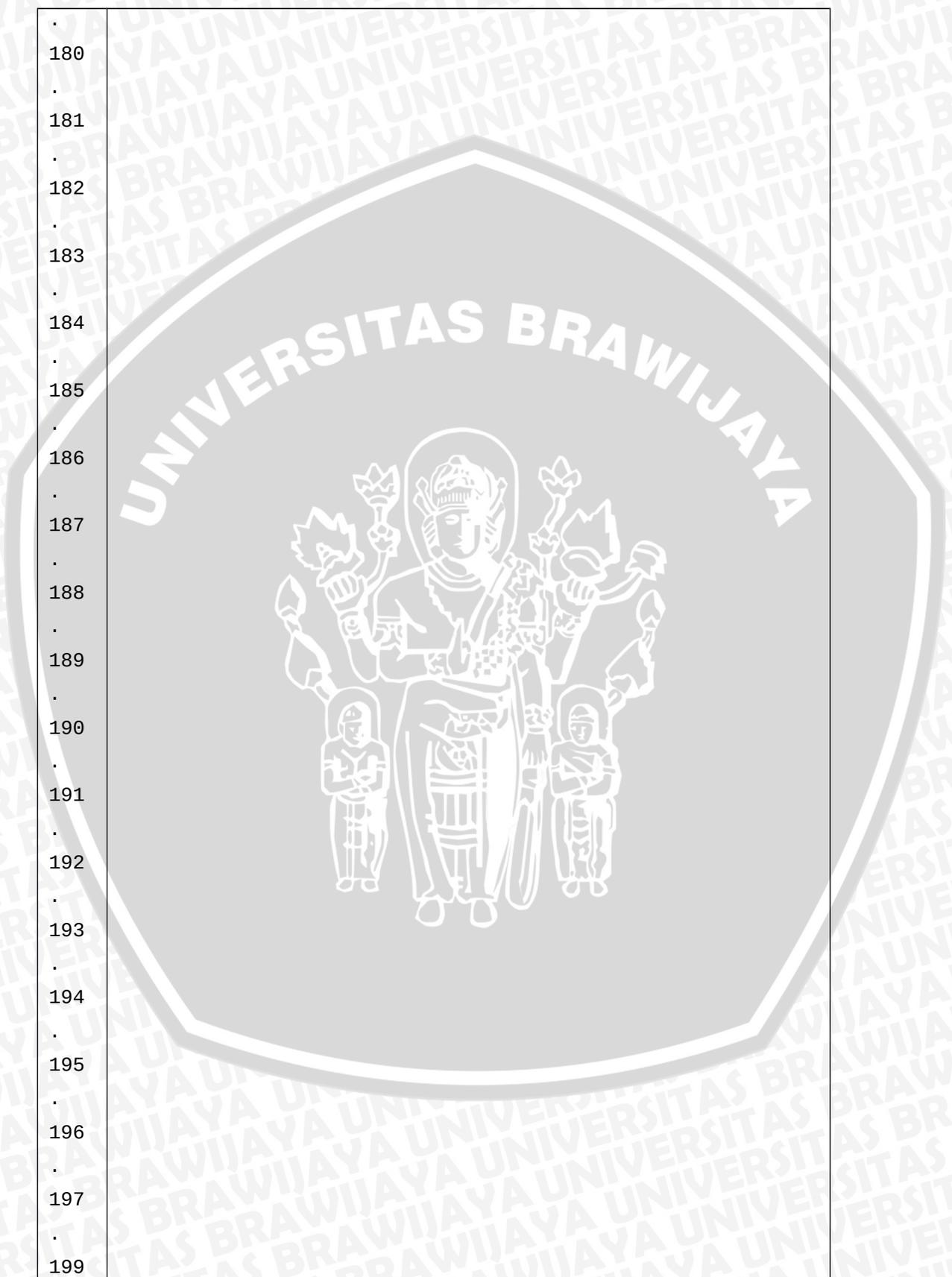
```
.         return $this->db->update('jenis_wisata', $data);
123     }
.     function get_hot(){
124         $sql = 'select id, nama_kec, nama, alamat, x, y
.         from hotel
125         INNER JOIN kecamatan ON kecamatan.id_kec =
.         hotel.kec_id
126         ';
.         $query = $this->db->query($sql);
127         $result = $query->result();
.         return $result;
128     }
.     function gethot(){
129         $sql = 'select id, nama_kec, nama, alamat, x,
.         y,img, ket, up_vote, down_vote from hotel
130         INNER JOIN kecamatan ON kecamatan.id_kec =
.         hotel.kec_id
131         ORDER BY hotel.up_vote DESC
.         limit 3
132         ';
.         $query = $this->db->query($sql);
133         $result = $query->result();
.         return $result;
134     }
.     function geth(){
135         $sql = 'select id, nama_kec, nama, alamat, x,
.         y,img, ket, up_vote, down_vote from hotel
136         INNER JOIN kecamatan ON kecamatan.id_kec =
.         hotel.kec_id
137         ORDER BY hotel.up_vote DESC';
.         $query = $this->db->query($sql);
138         $result = $query->result();
.         return $result;
139     }
.     public function tambah_h($data) {
140         return $this->db->insert('hotel', $data);
.     }
141     public function detail_hotel($id = FALSE) {
```

```
.         if ($id === FALSE) {
142         $query = $this->db->get('hotel');
.         return $query->result_array();
143     }
.         $query = $this->db->get_where('hotel', array('id'
144 => $id));
.         return $query->row_array();
145     }
.     public function edit_h($data) {
146         $this->db->where('id_wis', $data['id']);
.         return $this->db->update('hotel', $data);
147     }
.     public function tambah_k($data) {
148         return $this->db->insert('kecamatan', $data);
.     }
149     public function edit_h($data) {
.         $this->db->where('id_kec', $data['id']);
150         return $this->db->update('kecamatan', $data);
.     }
151
.
152
.
153
.
154
.
155
.
156
.
157
.
158
.
159
.
160
```

.  
161  
.  
162  
.  
163  
.  
164  
.  
165  
.  
166  
.  
167  
.  
168  
.  
169  
.  
170  
.  
171  
.  
172  
.  
173  
.  
174  
.  
175  
.  
176  
.  
177  
.  
178  
.  
179



.  
180  
.  
181  
.  
182  
.  
183  
.  
184  
.  
185  
.  
186  
.  
187  
.  
188  
.  
189  
.  
190  
.  
191  
.  
192  
.  
193  
.  
194  
.  
195  
.  
196  
.  
197  
.  
199



.	
200	
201	
.	
202	
.	

Tabel 5.3 Implementasi Site\_model

## 2. Implementasi Alam\_model

```

1. public function detail_alam($id = FALSE) {
2.     if ($id === FALSE) {
3.         $query = $this->db->get('jenis_wisata');
4.         return $query->result_array();
5.     }
6.     $this->db->where('wisata_jenis_id = 1');
7.     $query = $this->db->get_where('jenis_wisata',
8. array('id' => $id));
9.     return $query->row_array();
10 }
. public function get_koment($id){
11     $this->db->where('wisata_id', $id);
.     $this->db->order_by('id_ko', 'DESC');
12     $data = array();
.     $this->db->order_by('id_ko', 'DESC');
13     $q = $this->db->get('kom_wis', 10);
.     if ($q->num_rows() > 0) {
14         foreach ($q->result() as $row) {
.             $data[] = $row;
15         }
.     }
16     return $data;
.     $q->free_result();
17 }
. public function create_task($data){
18     $insert = $this->db->insert('kom_wis', $data);
.     return $insert;
19 }
. function updateDown($id){

```



```
20     $sql = "UPDATE jenis_wisata set DOWN_VOTE =  
    .     DOWN_VOTE+1 WHERE ID = ? ";  
21     $this->db->query($sql, array($id));  
    .     return $this->db->affected_rows();  
22 }  
    .     function updateUp($id){  
23         $sql = "UPDATE jenis_wisata set UP_VOTE = UP_VOTE+1  
    .     WHERE ID = ? ";  
24         $this->db->query($sql, array($id));  
    .     return $this->db->affected_rows();  
25     }  
    .  
26  
    .  
27  
    .  
28  
    .  
29  
    .  
30  
    .  
31  
    .  
32  
    .  
33  
    .  
34  
    .  
35  
    .  
36  
    .  
37  
    .  
38  
    .
```

39	
.	
40	
.	

Tabel 5.4 Implementasi Alam\_model

### 3. Implementasi Budaya\_Model

```

1. public function detail_budaya($id = FALSE) {
2.     if ($id === FALSE) {
3.         $query = $this->db->get('jenis_wisata');
4.         return $query->result_array();
5.     }
6.     $this->db->where('wisata_jenis_id = 2');
7.     $query = $this->db->get_where('jenis_wisata',
8. array('id' => $id));
9.     return $query->row_array();
10. }
. public function get_koment($id){
11.     $this->db->where('wisata_id', $id);
.     $this->db->order_by('id_ko', 'DESC');
12.     $data = array();
.     $this->db->order_by('id_ko', 'DESC');
13.     $q = $this->db->get('kom_wis', 10);
.     if ($q->num_rows() > 0) {
14.         foreach ($q->result() as $row) {
.             $data[] = $row;
15.         }
.     }
16.
.     return $data;
17.     $q->free_result();
. }
18. public function create_task($data){
.     $insert = $this->db->insert('kom_wis', $data);

```

```
19         return $insert;
20     }
21     function updateDown($voteId){
22         $sql = "UPDATE jenis_wisata set DOWN_VOTE =
23         DOWN_VOTE+1 WHERE ID = '$voteId' ";
24         $this->db->query($sql, array('id',$voteId));
25         return $this->db->affected_rows();
26     }
27
28     function updateUp($voteId){
29         $sql = "UPDATE jenis_wisata set UP_VOTE = UP_VOTE+1
30         WHERE ID = '$voteId' ";
31         $this->db->query($sql, array('id',$voteId));
32         return $this->db->affected_rows();
33     }
34
35
36
37
```

38
.
39
.
40
.
41
.
42
.

Tabel 5. 5 Implementasi Budaya\_model

#### 4. Implementasi Hotel\_model

```

1. public function detail_hotel($id = FALSE) {
2.     if ($id === FALSE) {
3.         $query = $this->db->get('hotel');
4.         return $query->result_array();
5.     }
6.     $query = $this->db->get_where('hotel', array('id'
7. => $id));
8.     return $query->row_array();
9. }
10 public function get_koment($id){
.     $this->db->where('wisata_id', $id);
11     $this->db->order_by('id_ko', 'DESC');
.     $data = array();
12     $this->db->order_by('id_ko', 'DESC');
.     $q = $this->db->get('kom_hot', 10);
13     if ($q->num_rows() > 0) {
.         foreach ($q->result() as $row) {
14             $data[] = $row;
.         }
15     }
.     return $data;
16     $q->free_result();
. }
17 public function create_task($data){
.     $insert = $this->db->insert('kom_hot', $data);

```



```
18         return $insert;
19     }
20 function updateDown($voteId){
21     $sql = "UPDATE hotel set DOWN_VOTE = DOWN_VOTE+1
22     WHERE ID = '$voteId' ";
23     $this->db->query($sql, array('id',$voteId));
24     return $this->db->affected_rows();
25 }
26 function updateUp($voteId){
27     $sql = "UPDATE hotel set UP_VOTE = UP_VOTE+1 WHERE
28     ID = '$voteId' ";
29     $this->db->query($sql, array('id',$voteId));
30     return $this->db->affected_rows();
31 }
32
33
34
35
36
```

37	
.	
38	
.	
39	
.	

Tabel 5.6 Implementasi Hotel\_Model

### 5.3 Implementasi Kode Pada Aplikasi

Aplikasi Sistem Informasi Geografis Berbasis Web pada penelitian ini memiliki beberapa method atau proses. Pada Bab implementasi program ini, akan dijelaskan proses dalam membangun peta. Bahasa pemograman yang dipakai adalah *JavaScript*.

#### 1. Implementasi Geojson Data Wisata Alam

```

1. var al = <?php echo json_encode($wisal);?>;
2.     var alLayer = L.geoJson(null);
3.     var alam = L.geoJson(al, {
4.         pointToLayer: function (feature, latlng) {
5.             return L.marker(latlng, {
6.                 riseOnHover: true
7.             });
8.         },
9.     }).addTo(map);
    
```

Tabel 5.7 Implementasi Geojson Data Wisata Alam

#### 2. Implementasi Geojson Data Budaya

```

1. var bud = <?php echo json_encode($wisbud);?>;
2.     var alLayer = L.geoJson(null);
3.     var budaya = L.geoJson(bud, {
4.         pointToLayer: function (feature, latlng) {
5.             return L.marker(latlng, {
6.                 riseOnHover: true
7.             });
8.         },
9.     }).addTo(map);
    
```

Tabel 5.8 Implementasi Geojson Data Budaya

#### 3. Implementasi Geojson Data Hotel

```

1. var hot = <?php echo json_encode($hot);?>;
2.     var allLayer = L.geoJson(null);
3.     var hotel = L.geoJson(hot, {
4.         pointToLayer: function (feature, latlng) {
5.             return L.marker(latlng, {
6.                 riseOnHover: true
7.             });
8.         },
9.     }).addTo(map);
10.

```

Tabel 5.9 Implementasi Geojson Data Hotel

#### 4. Implementasi Basemap

```

1. Var a = L.tileLayer("http://
2. {s}.tiles.mapbox.com/v3/spatialnetworks.map-6l9yntw9/{z}/
3. {x}/{y}.jpg70", {
4.     maxZoom: 19,
5.     subdomains: ["a", "b", "c", "d"],
6.     attribution: 'Basemap <a
7. href="https://www.mapbox.com/about/maps/"
8. target="_blank">© Mapbox © OpenStreetMap</a>'
9. });
10. var b = L.tileLayer("http://
11. {s}.tiles.mapbox.com/v3/spatialnetworks.map-xkumo5oi/{z}/
12. {x}/{y}.jpg70", {
13.     maxZoom: 19,
14.     subdomains: ["a", "b", "c", "d"],
15.     attribution: 'Basemap <a
16. href="https://www.mapbox.com/about/maps/"
17. target="_blank">© Mapbox © OpenStreetMap</a>'});

```

Tabel 5.10 Implementasi Basemap

#### 5. Implementasi Menampilkan Data Geojson Pada *Sidebar* dan Marker

```

1. alam.eachLayer(function(locale) {
2.     var prop = locale.feature.properties;
3.     var popup = '<h3>Wisata Alam</h3><div>' + prop.nama;
4.     var listing =
5.     listings.appendChild(document.createElement('div'));

```

```
6.     listing.className = 'item';
7.     var link =
8.     listing.appendChild(document.createElement('a'));
9.     link.href = '#';
10.    link.className = 'title';
11.    link.innerHTML = prop.nama;
12.    if (prop.lokasi) {
13.        link.innerHTML += '<br /><small class="quiet">' +
14.    prop.lokasi + '</small>';
15.        popup += '<br /><small class="quiet">' + prop.lokasi
16.    + '</small>';
17.        popup += '<br /><br /><i class= "fa fa-thumbs-o-up
18.    ">&nbsp;'; + prop.up_vote + '</li>&nbsp;&nbsp;&nbsp;|' + '|'
19.    &nbsp;&nbsp;&nbsp;<i class= "fa fa-thumbs-o-down ">&nbsp;'; +
20.    prop.down_vote + '</li>';
21.    }
22.    var details =
23.    listing.appendChild(document.createElement('div'));
24.    if (prop.link) {
25.        details.innerHTML += '<small class="quiet"> <a
26.    target="_blank" href="'+prop.link+'">Read
27.    More</a></small>';
28.    }
29.    link.onclick = function() {
30.        setActive(listing);
31.        map.setView(locale.getLatLng(), 16);
32.        locale.openPopup();
33.        return false;
34.    };
35.    locale.on('click', function(e) {
36.        map.panTo(locale.getLatLng());
37.        setActive(listing);
38.    });
39.    popup += '</div>';
40.    locale.bindPopup(popup);
41.    locale.setIcon(L.icon({
42.        iconUrl: '<?php echo base_url
43.    ('assets/images/panoramicview.png');?>',
```

```

44.         iconSize: [40, 40],
45.         iconAnchor: [28, 28],
46.         popupAnchor: [0, -34]
47.     }));
48. });

```

Tabel 5.11 Implementasi Menampilkan Data Geojson Pada Sidebar dan Marker

## 6. Implementasi Rute Plan Pada Peta

```

1.  var control = L.Routing.control({
2.      routewhileDragging: true,
3.      plan: new (L.Routing.Plan.extend({
4.          createGeocoders: function() {
5.              var container =
6.  L.Routing.Plan.prototype.createGeocoders.call(this),
7.              reverseButton = button('&#8593;&#8595;',
8.  container);
9.              L.DomEvent.on(reverseButton, 'click', function()
10.                 var waypoints = this.getWaypoints();
11.                 this.setWaypoints(waypoints.reverse());
12.             }, this);
13.             return container;
14.         }
15.     }))( [ ],
16.     {
17.         geocoder: L.Control.Geocoder.nominatim(),
18.         routewhileDragging: true
19.     })
20. }).addTo(map);
21. alam.on('click', function(e) {
22.     var container = L.DomUtil.create('div'),
23.     startBtn = button('Start from this location',
24.     container),
25.     destBtn = button('Go to this location',
26.     container);
27.
28.     L.DomEvent.on(startBtn, 'click', function() {
29.         control.spliceWaypoints(0, 1, e.latlng);
30.         map.closePopup();
31.     });
32.
33.     L.DomEvent.on(destBtn, 'click', function() {
34.
35.     control.spliceWaypoints(control.getWaypoints().length - 1,
36.     1, e.latlng);
37.     map.closePopup();
38.     });
39.

```

40.	L.popup()
41.	.setContent(container)
42.	.setLatLng(e.latlng)
43.	.openOn(map);
44.	});
45.	

Tabel 5.12 Implementasi Rute Plan Pada Peta

## 7. Implementasi Komentar

```

1. <form class="form-horizontal style-form" method="POST"
2. action="<?php echo site_url('alam/kawahsubmit');?>">
3. <div class="form-group">
4.   <label for="name" class="col-sm-2 control-
5. label">Nama</label>
6.   <div class="col-sm-6">
7.     <input id="id" name="name" placeholder="Nama"
8. type="text" class="form-control" value="<?=
9. set_value('name') ?>" />
10.   <span class="text-danger"><?php echo form_error('name');
11. ?></span>
12.   </div>
13. </div>
14. <div class="form-group">
15.   <label for="email" class="col-sm-2 control-
16. label">Email</label>
17.   <div class="col-sm-6">
18.     <input id="id" name="email" placeholder="Email"
19. type="text" class="form-control" value="<?=
20. set_value('email') ?>" />
21.     <span class="text-danger"><?php echo
22. form_error('email'); ?></span>
23.   </div>
24. </div>
25. <div class="form-group">
26.   <label for="message" class="col-sm-2 control-
27. label">Testimoni</label>
28.   <div class="col-sm-6">
29.     <textarea input type="text" rows= "4" name="message"
30. id="" class="form-control" placeholder="Testimoni"
31. value="<?= set_value('message') ?>"></textarea>

```

```
32.     <span class="text-danger"><?php echo
33.     form_error('message'); ?></span>
34. </div>
35. </div>
36. <div class="row">
37. <div class="col-md-10 col-sm-12 col-md-offset-2 col-sm-
38. offset-0"> <input id="btn_add" name="btn_add"
39. type="submit" class="btn1 btn1-warning" value="Submit" />
40.         </div>
41.     </div>
42. </form>
43. <div class="dem"></div>
44. <br>
45. <?php
46. if ($shouts) {
47.     echo '<ul>';
48.     foreach ($shouts as $shout) {
49.         $gravatar = 'http://www.gravatar.com/avatar.php?
50. gravatar_id=' . md5(strtolower($shout->email)) .
51. '&size=70';
52.         ?>
53. <li>
54.         <div class="meta">
55.             
56.             <p><?= $shout->name ?></p>
57.         </div>
58.         <div class="about shout">
59.             <p><?= $shout->message ?></p>
60.         </div>
61.     </li>
62. <?php
63. }
64. }
65. else {
66.     echo '<p class="error">No shouts found!</p>';
67. }
68. ?>
69.
```

Tabel 5.13 Implementasi Komentar

## 8. Implementasi Voting

```

1. <script>
2. $(document).ready(function(){
3.     $(".voteMe").click(function() {
4.         var voteId = this.id;
5.         var upOrDown = voteId.split('_');
6.         $.ajax({
7.             type: "post",
8.             url: "<?php echo site_url('alam/voteijen'); ?>",
9.             cache: false,
10.            data: 'voteId='+upOrDown[0] + '&upOrDown='
11.            +upOrDown[1],
12.            success: function(response){
13.                try{
14.                    if(response=='true'){
15.                        var newValue = parseInt($
16.                        ("#" + voteId + '_result').text()) + 1;
17.                        $("#" + voteId + '_result').html(newValue);
18.                    }else{
19.                        alert('Sorry Unable to update..!');
20.                    }
21.                }catch(e) {
22.                    alert('Exception while request..!');
23.                }
24.            },
25.            error: function(){
26.                alert('Error while request..!');
27.            }
28.        });
29.    });
30. });
31. </script>
32. <hr>
33. <?php foreach ($vote as $rate) : ?>
34. <div class="panel panel-default center"
35. style="width:300px;">

```



.	
4	
.	

Tabel 5.16 Implementasi Tabel Kecamatan

3. Tabel Wisata

1	CREATE TABLE IF NOT EXISTS `wisata` (
.	`jenis_id` int(11) NOT NULL,
2	`jenis_wisata` varchar(255) NOT NULL,
.	`jenis_id_kec` varchar(255) NOT NULL
3	) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
.	
4	
.	
5	
.	

Tabel 5.17 Implementasi Tabel Wisata

4. Tabel Jenis Wisata

1.	CREATE TABLE IF NOT EXISTS `jenis_wisata` (
2.	`id` int(11) NOT NULL,
3.	`wisata_jenis_id` int(11) NOT NULL,
4.	`wisata_id_kec` int(11) NOT NULL,
5.	`nama_wis` varchar(255) NOT NULL,
6.	`alamat_wis` varchar(255) NOT NULL,
7.	`x` double NOT NULL,
8.	`y` double NOT NULL,
9.	`img` varchar(255) NOT NULL,
8.	`up_vote` int(11) NOT NULL,
9.	`down_vote` int(11) NOT NULL,
10	`ket` text NOT NULL
.	) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=latin1;
11	
.	

Tabel 5. 18 Implementasi Tabel Jenis Wisata

5. Tabel Hotel

1.	CREATE TABLE IF NOT EXISTS `hotel` (
2.	`id` int(11) NOT NULL,
3.	`kec_id` int(11) DEFAULT NULL,



```

4. `nama` varchar(255) NOT NULL,
5. `alamat` varchar(255) DEFAULT NULL,
6. `x` double NOT NULL,
7. `y` double NOT NULL,
8. `img` varchar(255) NOT NULL,
9. `ket` text NOT NULL,
8. `up_vote` int(11) NOT NULL,
9. `down_vote` int(11) NOT NULL
10 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
.

```

Tabel 5.19 Implementasi Tabel Hotel

6. Tabel Komentar Wisata

```

1. CREATE TABLE IF NOT EXISTS `kom_wis` (
2. `id_ko` int(11) NOT NULL,
3. `wisata_id` int(11) NOT NULL,
4. `nama` varchar(255) NOT NULL,
5. `pesan` varchar(255) NOT NULL,
6. `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
7. ) ENGINE=InnoDB AUTO_INCREMENT=42 DEFAULT CHARSET=latin1;

```

Tabel 5.20 Implementasi Tabel Komentar Wisata Alam

7. Tabel Komentar Hotel

```

1. CREATE TABLE IF NOT EXISTS `kom_hot` (
2. `id` int(11) NOT NULL,
3. `hotel_id` int(11) NOT NULL,
4. `nama` int(11) NOT NULL,
5. `pesan` int(11) NOT NULL
6. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

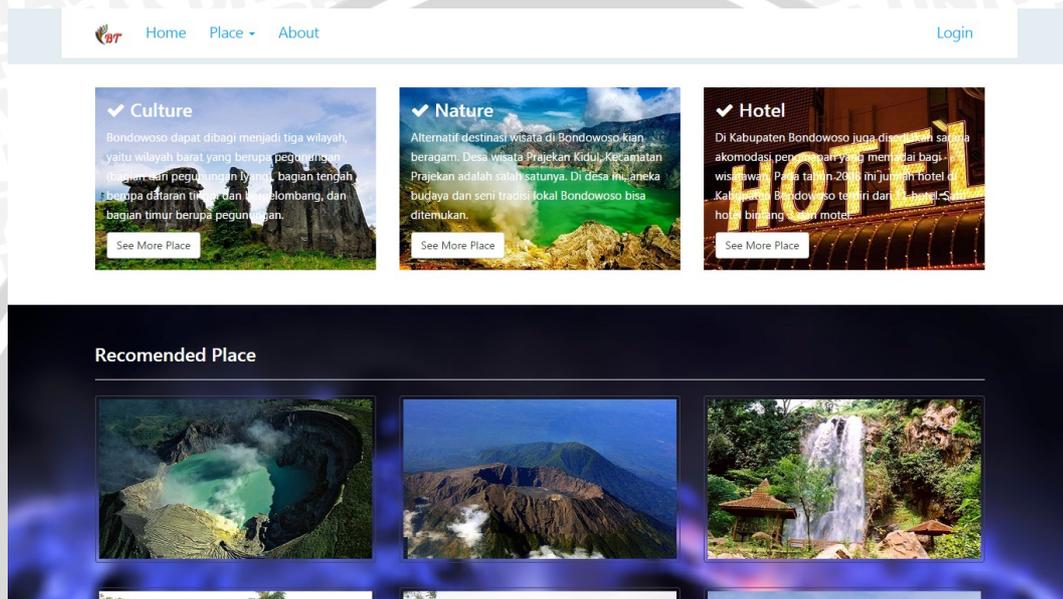
```

Tabel 5.21 Implementasi Tabel Komentar Hotel

## 5.5 Implementasi User Interface

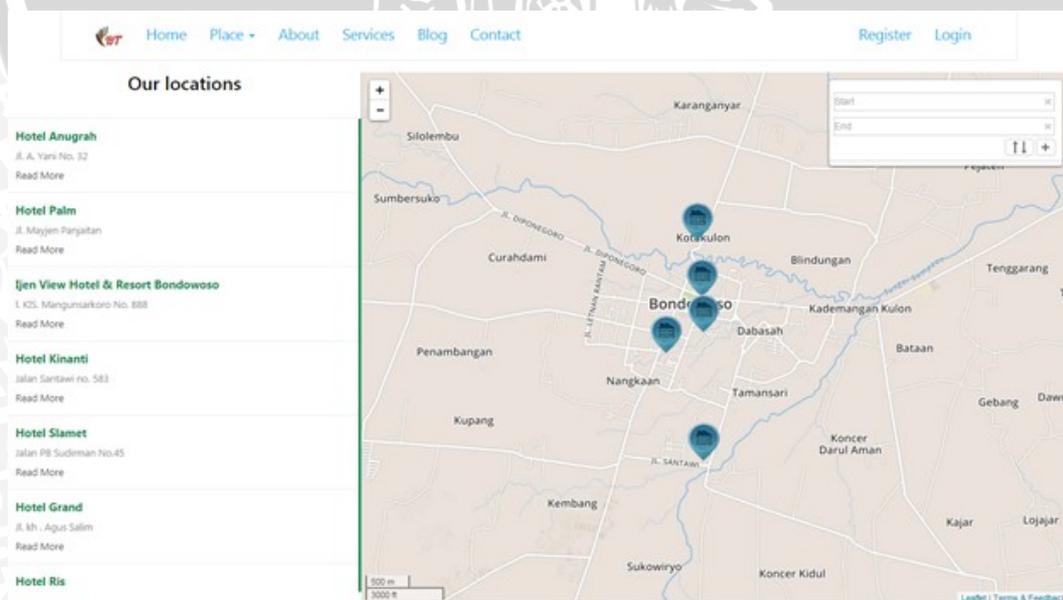
Implementasi *user interface* dibangun berdasarkan pada perancangan *user interface* pada bab perancang menggunakan bahasa pemrograman html dan *JavaScript*.

### 1. Halaman Home



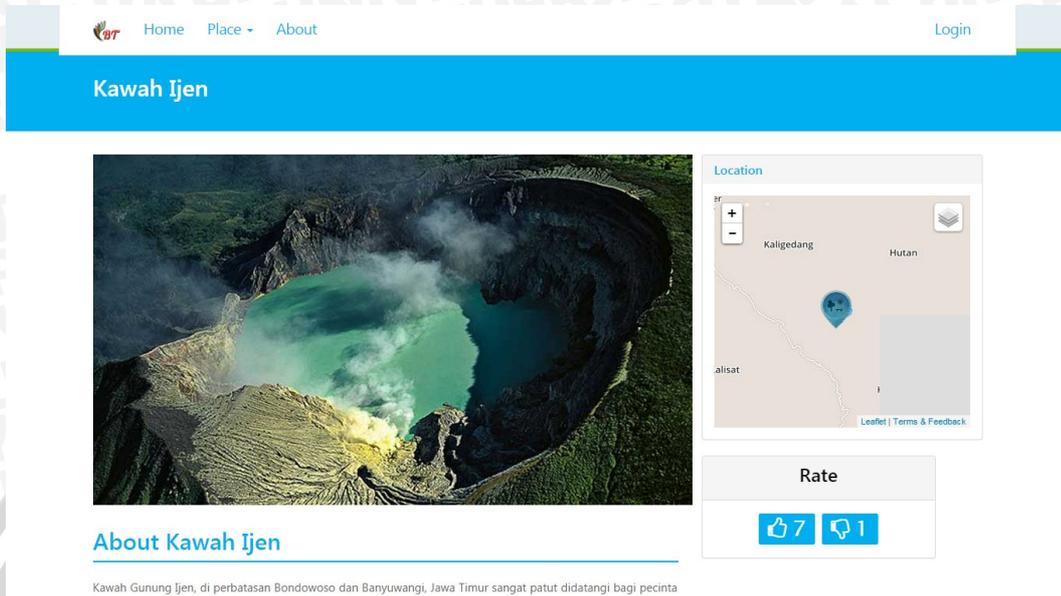
Gambar 5.2 Halaman Home

### 2. Halaman Map



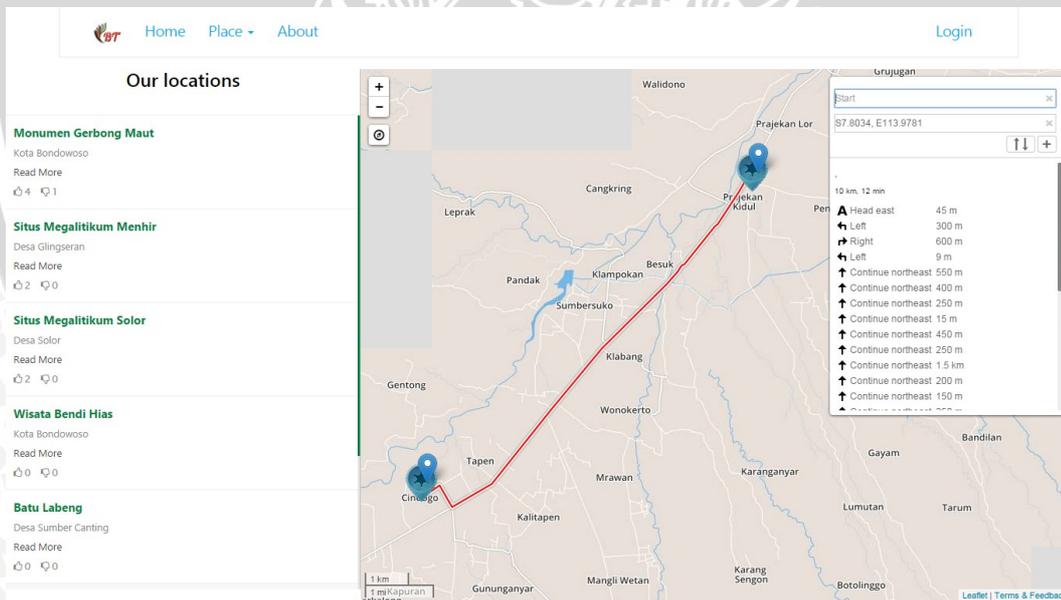
Gambar 5.3 Halaman Map

### 3. Halaman Informasi Wisata



Gambar 5.4 Halaman Informasi Wisata

### 4. Implementasi Rute Wisata

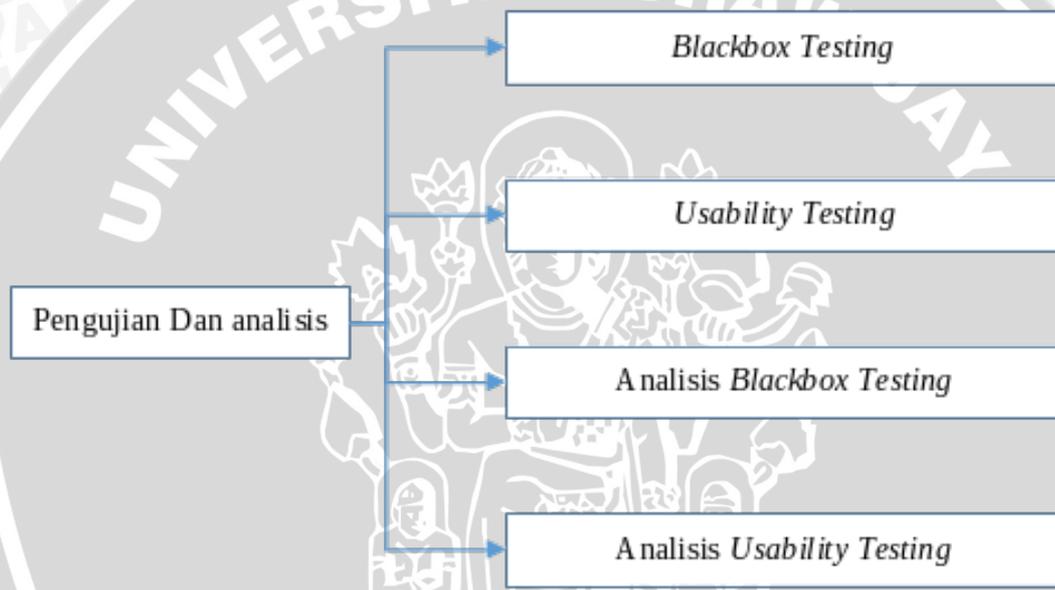


Gambar 5.5 Implementasi Rute Wisata

## BAB VI

### PENGUJIAN DAN ANALISIS

Pengujian dan analisis sistem ini bertujuan untuk menguji komponen sistem yang telah dirancang sebelumnya dan untuk memastikan bahwa setiap elemen dari system telah berfungsi sesuai dengan yang diharapkan. Dalam pengujian ini akan membahas mengenai Rencana Pengujian, Kasus dan Hasil Pengujian dan Kesimpulan Hasil Pengujian.



Gambar 6.1 Pohon Pengujian

### 6.1 BlackBox Testing (Uji Validasi)

*BlackBox Testing* adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada input dan output aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum). Tahap pengujian atau testing merupakan salah satu tahap yang harus ada dalam sebuah siklus pengembangan perangkat lunak. Berikut ini pengujian dari sistem :

#### 1. Pengujian Validasi Admin

No Kasus	VAL_01
Nama Kasus Uji	Login Admin
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas login kecamatan bagi admin
Prosuder Uji	<ol style="list-style-type: none"> <li>1. Admin masuk halaman login</li> <li>2. Admin mengisi username dan password</li> <li>3. Admin menekan tombol login</li> <li>4. Password atau username salah kembali ke halaman login</li> </ol>
Hasil Yang DiHarapkan	Sistem dapat masuk ke halaman admin
Hasil Yang DiDapat	Sistem dapat masuk ke halaman admin
Status Validasi	Valid

Tabel 6.1 Pengujian Login Admin

No Kasus	VAL_02
Nama Kasus Uji	Logout Admin
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas logout bagi admin
Prosuder Uji	<ol style="list-style-type: none"> <li>1. Admin masuk halaman admin</li> <li>2. Admin menekan tombol logout</li> </ol>
Hasil Yang DiHarapkan	Sistem dapat keluar dari halaman admin
Hasil Yang DiDapat	Sistem dapat keluar dari halaman admin
Status Validasi	Valid

Tabel 6.2 Pengujian Logout Admin

#### 2. Pengujian Olah data Kecamatan Oleh Admin

Kasus uji validasi olah data kecamatan oleh administrator sebagai berikut.

No Kasus Uji	VAL_03
Nama Kasus Uji	Tambah Data Kecamatan
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas tambah data kecamatan bagi admin

Prosuder Uji	<ol style="list-style-type: none"> <li>1. Admin masuk halaman kecamatan</li> <li>2. Admin menekan tombol tambah data</li> <li>3. Admin memasukkan data kecamatan</li> <li>4. Admin menekan tombol simpan</li> </ol>
Hasil Yang DiHarapkan	Sistem dapat menyimpan data kecamatan baru.
Hasil Yang DiDapat	Sistem menyimpan data kecamatan baru.
Status Validasi	Valid

Tabel 6.3 Kasus Uji Validasi Tambah Data Kecamatan

No Kasus Uji	VAL_04
Nama Kasus Uji	Ubah Data Kecamatan
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas ubah data kecamatan bagi admin
Prosuder Uji	<ol style="list-style-type: none"> <li>1. Admin masuk halaman kecamatan</li> <li>2. Admin menekan tombol ubah data</li> <li>3. Admin memasuki halaman ubah data kecamatan</li> <li>4. Admin memperbaiki sebagian atau seluruh data</li> <li>5. Admin menekan tombol simpan</li> </ol>
Hasil Yang DiHarapkan	Sistem dapat menyimpan data baru kecamatan. Masuk ke halaman lihat data kecamatan.
Hasil Yang DiDapat	Sistem dapat menyimpan data baru kecamatan. Masuk ke halaman lihat data kecamatan.
Status Validasi	Valid

Tabel 6.4 Kasus Uji Validasi Ubah Data Kecamatan

No Kasus Uji	VAL_05
Nama Kasus Uji	Hapus Data Kecamatan
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas hapus data kecamatan bagi admin
Prosuder Uji	<ol style="list-style-type: none"> <li>1. Admin masuk halaman kecamatan</li> <li>2. Admin menekan tombol hapus data</li> <li>3. Admin menekan ya pada notifikasi penghapusan</li> </ol>
Hasil Yang DiHarapkan	Sistem dapat menghapus data wisata alam yang dipilih..
Hasil Yang DiDapat	Sistem dapat menghapus data wisata alam yang

	dipilih.
Status Validasi	Valid

Tabel 6.5 Kasus Uji Validasi Hapus Data Kecamatan

3. Pengujian Olah data Wisata Oleh Admin

Kasus uji validasi olah data wisata alam oleh administrator sebagai berikut.

No Kasus Uji	VAL_06
Nama Kasus Uji	Tambah Data Wisata
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas tambah data wisata bagi admin
Prosuder Uji	<ol style="list-style-type: none"> <li>1. Admin masuk halaman wisata</li> <li>2. Admin menekan tombol tambah data</li> <li>3. Admin memasukkan data wisata</li> <li>4. Admin menekan tombol simpan</li> </ol>
Hasil Yang DiHarapkan	Sistem dapat menyimpan data wisata baru.
Hasil Yang DiDapat	Sistem menyimpan data wisata baru.
Status Validasi	Valid

Tabel 6.6 Kasus Uji Validasi Tambah Data Wisata

No Kasus Uji	VAL_07
Nama Kasus Uji	Ubah Data Wisata
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas ubah data wisata bagi admin
Prosuder Uji	<ol style="list-style-type: none"> <li>1. Admin masuk halaman wisata</li> <li>2. Admin menekan tombol ubah data</li> <li>3. Admin memasuki halaman ubah data wisata</li> <li>4. Admin memperbaiki sebagian atau seluruh data wisata</li> <li>5. Admin menekan tombol simpan</li> </ol>
Hasil Yang DiHarapkan	Sistem dapat menyimpan data baru wisata. Masuk ke halaman lihat data wisata.
Hasil Yang DiDapat	Sistem dapat menyimpan data baru wisata. Masuk ke

	halaman lihat data wisata
Status Validasi	Valid

Tabel 6.7 Kasus Uji Validasi Ubah Data Wisata

No Kasus Uji	VAL_08
Nama Kasus Uji	Hapus Data Wisata
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas hapus data wisata bagi admin
Prosuder Uji	1. Admin masuk halaman wisata 2. Admin menekan tombol hapus data 3. Admin menekan ya pada notifikasi penghapusan
Hasil Yang DiHarapkan	Sistem dapat menghapus data wisata yang dipilih..
Hasil Yang DiDapat	Sistem dapat menghapus data wisata yang dipilih.
Status Validasi	Valid

Tabel 6.8 Kasus Uji Validasi Hapus Data Wisata

4. Pengujian Olah data Hotel Oleh Admin

Kasus uji validasi olah data hotel oleh administrator sebagai berikut.

No Kasus Uji	VAL_09
Nama Kasus Uji	Tambah Data hotel
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas tambah data hotel bagi admin
Prosuder Uji	1. Admin masuk halaman hotel 2. Admin menekan tombol tambah data 3. Admin memasukkan data hotel 4. Admin menekan tombol simpan
Hasil Yang DiHarapkan	Sistem dapat menyimpan data hotel baru.
Hasil Yang DiDapat	Sistem menyimpan data hotel baru.
Status Validasi	Valid

Tabel 6.9 Kasus Uji Validasi Tambah Data Hotel

No Kasus Uji	VAL_010
Nama Kasus Uji	Ubah Data Hotel
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas ubah data hotel bagi admin
Prosuder Uji	1. Admin masuk halaman wisata budaya 2. Admin menekan tombol ubah data 3. Admin memasuki halaman ubah data hotel 6. Admin memperbarui data hotel 5. Admin menekan tombol simpan
Hasil Yang DiHarapkan	Sistem dapat menyimpan data baru hotel. Masuk ke halaman lihat data hotel.
Hasil Yang DiDapat	Sistem dapat menyimpan data baru hotel. Masuk ke halaman lihat data hotel
Status Validasi	Valid

Tabel 6.10 Kasus Uji Validasi Ubah Data Hotel

No Kasus Uji	VAL_011
Nama Kasus Uji	Hapus Data Hotel
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas hapus data hotel bagi admin
Prosuder Uji	1. Admin masuk halaman hotel 2. Admin menekan tombol hapus data 3. Admin menekan ya pada notifikasi penghapusan
Hasil Yang DiHarapkan	Sistem dapat menghapus data hotel yang dipilih..
Hasil Yang DiDapat	Sistem dapat menghapus data hotel yang dipilih.
Status Validasi	Valid

Tabel 6.11 Kasus Uji Validasi Hapus Data Hotel

##### 5. Pengujian Sistem Menampilkan Peta Digital

No Kasus Uji	VAL_12
Nama Kasus Uji	Menampilkan Peta Digital
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menampilkan peta digital
Prosuder Uji	1. pengguna masuk halaman map 2. Sistem menampilkan peta digital
Hasil Yang DiHarapkan	Sistem menampilkan peta digital
Hasil Yang DiDapat	Sistem menampilkan peta digital
Status Validasi	Valid

Tabel 6.12 Kasus Uji Validasi Menampilkan Peta Digital

6. Pengujian Menampilkan Marker Pada Peta

No Kasus Uji	VAL_13
Nama Kasus Uji	Menampilkan Marker Pada Peta
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menampilkan marker lokasi wisata pada peta
Prosuder Uji	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu wisata</li> <li>2. Menampilkan halaman wisata</li> <li>3. menampilkan marker lokasi wisata</li> </ol>
Hasil Yang DiHarapkan	Sistem menampilkan marker lokasi wisata
Hasil Yang DiDapat	Sistem menampilkan marker lokasi wisata
Status Validasi	Valid

Tabel 6.13 Kasus Uji Validasi Mencari Lokasi Pengguna

7. Pengujian Mencari Lokasi Pengguna

No Kasus Uji	VAL_14
Nama Kasus Uji	Mencari Lokasi Pengguna
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menampilkan lokasi pengguna
Prosuder Uji	<ol style="list-style-type: none"> <li>4. pengguna masuk halaman map</li> <li>5. pengguna memilih fitur geolocatin</li> <li>6. menampilkan lokasi pengguna pada map</li> </ol>
Hasil Yang DiHarapkan	Sistem menampilkan lokasi pengguna pada map
Hasil Yang DiDapat	Sistem menampilkan lokasi pengguna pada map
Status Validasi	Valid

Tabel 6.14 Kasus Uji Validasi Mencari Lokasi Pengguna



8. Pengujian Mencari Rute Menuju Objek Wisata Oleh Pengguna

No Kasus Uji	VAL_15
Nama Kasus Uji	Mencari Rute Menuju Objek Wisata
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menampilkan rute yang berupa line untuk menuju tempat wisata
Prosuder Uji	<ol style="list-style-type: none"> <li>1. pengguna masuk halaman map</li> <li>2. pengguna menekan marker lokasi saat ini</li> <li>3. pengguna menekan marker objek wisata</li> <li>4. menampilkan rute</li> </ol>
Hasil Yang DiHarapkan	Sistem menampilkan rute menuju lokasi wisata
Hasil Yang DiDapat	Sistem menampilkan rute menuju lokasi wisata
Status Validasi	Valid

Tabel 6.15 Kasus Uji Validasi Mencari Rute Menuju Objek Wisata

9. Pengujian Pemberian Rating Objek Wisata Oleh Pngunjung

No Kasus Uji	VAL_16
Nama Kasus Uji	Pemberian Rating Objek Wisata
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam memberikan rating objek wisata
Prosuder Uji	<ol style="list-style-type: none"> <li>1. pengguna masuk halaman map</li> <li>2. pengguna memilih wisata pada sidebar.</li> <li>3. pengguna memilih readmore</li> <li>6. menampilkan halaman wisata</li> <li>5. pengguna memilih like atau dislike wisata</li> </ol>
Hasil Yang DiHarapkan	Sistem dapat memberikan rating pada objek wisata
Hasil Yang DiDapat	Sistem dapat memberikan rating pada objek wisata
Status Validasi	Valid

Tabel 6.16 Kasus Uji Validasi Pemberian Rating Objek Wisata Dan Hotel

10. Pengujin Pemeberian Komentar Objek Wisata Oleh Pegunjung



No Kasus Uji	VAL_17
Nama Kasus Uji	Pemberian Komentar Objek Wisata
Tujuan Pengujian	Untuk menguji validitas kinerja dari system dalam memberikan proses pemberian komentar objek wisata
Prosuder Uji	1. pengguna masuk halaman map 2. pengguna memilih wisata pada sidebar. 3. pengguna memilih readmore 6. menampilkan halaman wisata 5. pengguna memberikan komentar
Hasil Yang DiHarapkan	Pengguna dapat memberikan komentar dan di tampilkan pada halaman wista
Hasil Yang DiDapat	Pengguna dapat memberikan komentar dan di tampilkan pada halaman wista
Status Validasi	Valid

Tabel 6.17 Kasus Uji Validasi Mencari Mencari Objek Wisata Dan Hotel

### 6.2 Usability Testing

Pengujian usability dilakukan untuk menguji seberapa berguna produk dan seberapa mudahnya user interface yang digunakan pada pengembangan perangkat lunak. Pengujian ini dilakukan dengan cara membagikan kuesioner kepada calon pengguna. Jenis kuesioner yang digunakan dalam pengujian *usability* ini adalah SUS (*system usability scala*).

Skala Likert adalah suatu [skala psikometrik](#) yang umum digunakan dalam [kuesioner](#), dan merupakan skala yang paling banyak digunakan dalam [riset](#) berupa [survei](#). Nama skala ini diambil dari nama [Rensis Likert](#), yang menerbitkan suatu laporan yang menjelaskan penggunaannya.

Dalam pengujian ini kriteria khusus pada responden seperti rentan usia, jenis kelamin dan latar belakang pendidikan diabaikan. Jumlah responden calon pengguna yang diuji sebanyak 10 orang. Komponen pertanyaan yang digunakan untuk pengujian ditujukan pada tabel Dengan respon sebagai berikut

Penilaian:    5 = Sangat Setuju    4 = Setuju    3 = Biasa  
                  2 = Tidak Setuju    1 = Sangat Tidak Setuju

Kriteria interpretasi skor berdasarkan interval:

- Angka 0% – 19,99% = Sangat (Tidak Setuju/Buruk/Kurang Sekali)
- Angka 20% – 39,99% = Tidak setuju (Kurang Baik)
- Angka 40% – 59,99% = Cukup / Netral
- Angka 60% – 79,99% = (Setuju/Baik/suka)
- Angka 80% – 100% = Sangat (Setuju/Baik/Suka)

No	Pertanyaan	Penilaian				
		1	2	3	4	5
<b>Aspek Learnability</b>						
1	Apakah tampilan aplikasi yang dibangun cukup menarik?					
2	Apakah sistem aplikasi mudah dioperasikan?					
<b>Aspek Efficiency</b>						
3	Apakah aplikasi ini mempermudah pengunjung dalam memperoleh informasi wisata yang ada di Bondowoso?					
4	Apakah rute yang disediakan mempermudah pengunjung untuk menuju lokasi wisata?					
5	Apakah peta yang disediakan dalam penyajian lokasi wisata mudah dipahami?					
<b>Aspek Satisfaction</b>						
6	Apakah fitur-fitur yang disediakan cukup membantu?					
7	Apakah sistem aplikasi berjalan sesuai dengan prosedur yang diinginkan?					
8	Bagaimana kualitas keseluruhan aplikasi?					

Tabel 6.18 Kuisiner Usability Testing

Pada Tabel 6.18 ditunjukkan pertanyaan-pertanyaan menjelaskan seberapa cepat pengguna mengoperasikan aplikasi dan tingkat kepuasan pengguna. Dari pernyataan diatas didapatkan total jawaban dari 10 responden. Rincian jumlah dari jawaban hasil kuesioner ditunjukkan pada Tabel 6.19



No	Responden	Penilaian atas Pertanyaan yang di Ajukan							
		1	2	3	4	5	6	7	8
1	Responden ke 1	5	5	5	4	5	4	4	5
2	Responden ke 2	4	5	5	5	4	5	4	4
3	Responden ke 3	5	5	5	4	5	5	5	5
4	Responden ke 4	5	5	4	4	4	5	5	5
5	Responden ke 5	5	5	4	4	5	5	5	5
6	Responden ke 6	4	4	5	5	4	4	4	4
7	Responden ke 7	4	5	5	5	5	4	5	5
8	Responden ke 8	4	4	4	5	4	4	5	5
9	Responden ke 9	4	5	4	4	4	4	4	4
10	Responden ke 10	4	4	4	4	5	4	5	4

Tabel 6.19 Rincian Jumlah Dari Jawaban Hasil Kuesioner

Untuk mendapatkan hasil interpretasi, harus diketahui dulu skor tertinggi (Y) untuk item penilaian dengan rumus sebagai berikut

$$Y = \text{skor tertinggi likert} \times \text{jumlah responden}$$

Nilai skor tertinggi untuk kategori “sangat setuju” adalah 5, yang berarti  $5 \times 8 = 40$ . Maka untuk interpretasi skor aplikasi secara keseluruhan didapat sebagai berikut:

Penyelesaian akhir:

Sangat Kurang =  $1 \times 0 = 0$

Kurang =  $2 \times 0 = 0$

Cukup Bagus =  $3 \times 0 = 0$

Bagus =  $4 \times 41 = 164$

Sangat Bagus =  $5 \times 40 = 200$

Total skor = 277

Index akhir % = Total skor / Y x 100

$$= 364 / 40 \times 100 = 91\% \text{ Sangat (Setuju/Baik/Suka)}$$



### 6.3 Analisis Pengujian Validasi

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian sistem informasi objek wisata kabupaten bondowoso yang telah dilakukan. Proses analisis mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan. Analisis dilakukan terhadap hasil pengujian di setiap tahap pengujian. Proses analisis yang dilakukan meliputi analisis hasil pengujian validasi. Validasi dilakukan dengan memantau antara kinerja sistem dengan daftar kebutuhan. Hasil pengujian fungsionalitas pada kasus uji Val\_01 hingga Val\_17 dengan metode *Black-Box Testing* secara keseluruhan adalah sangat baik. Dapat disimpulkan bahwa implementasi dan fungsionalitas aplikasi dapat berjalan sesuai dengan spesifikasi kebutuhan yang telah dijabarkan pada analisis kebutuhan.

### 6.4 Analisis Pengujian Usability

Dengan pengujian *usability* menggunakan kuesioner jenis Skala Likert didapatkan hasil interpretasi 91% secara keseluruhan, yang berarti responden memberi tanggapan sangat baik pada aplikasi. Dari tanggapan para responden yang meliputi tampilan aplikasi hingga cara penggunaan aplikasi menunjukkan bahwa responden sangat setuju dengan adanya sistem informasi geografis objek wisata Bondowoso.

## BAB VII

### KESIMPULAN DAN SARAN

## 7.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

- 1 Sesuai dengan rumusan masalah Sistem Informasi Geografis Mengenai Objek Wisata Bondowoso data yang dianalisis berupa data koordinat wisata yang berada di Kabupaten Bondowoso. Data yang dibutuhkan adalah data wisata alam, data wisata budaya dan hotel.
- 2 Perancangan aplikasi Sistem Informasi Geografis Mengenai Objek Wisata di Kabupaten Bondowoso dibuat sesuai dengan kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan dengan menggunakan pemodelan UML (*unified modeling language*) dan ERD (*entity relationship diagram*).
- 3 Sistem Informasi Geografis Mengenai Objek Wisata di Kabupaten Bondowoso telah dibuat sesuai perancangan dan diimplementasikan dari komponen-komponen yang telah ditentukan, antara lain: Codeigniter framework, LeafletJS dan Sistem Informasi Geografis.
- 4 Berdasarkan pengujian *usability* aplikasi Sistem Informasi Geografis Mengenai Objek Wisata di Kabupaten Bondowoso memiliki skor 91% yang berarti bahwa nilai kegunaan dari aplikasi sangat baik dan menunjukkan bahwa aplikasi Sistem Informasi Geografis Mengenai Objek Wisata di Kabupaten Bondowoso mudah digunakan sehingga dapat membantu pengguna dalam mendapatkan informasi terkait dengan objek wisata di Kabupaten Bondowoso

## 7.2 Saran

Aplikasi Sistem Informasi Geografis Mengenai Objek Wisata di Kabupaten Bondowoso dibuat sesuai dengan kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan yaitu gambaran umum perangkat lunak sistem, melakukan proses identifikasi aktor yang terlibat dalam sistem perangkat lunak, melakukan proses analisis data yang diperlukan, membuat daftar kebutuhan pengguna berdasarkan penjabaran gambaran umum perangkat lunak dan menggunakan pemodelan diagram *use case* untuk menggambarkan kebutuhan tersebut

Saran dalam aplikasi ini untuk pengembangan yang lebih lanjut adalah penambahan beberapa fitur. Menambahkan Fitur GeoTagging agar pengguna bisa memasukkan wisata yang menarik pada peta, Penambahan fitur nearest place agar pengguna bisa mengetahui objek wisata terdekat dan Dibuat aplikasi mobile untuk aplikasi ini.

## DAFTAR PUSTAKA

- [THC-11] Tzu-How Chu, Meng-Lung Lin, Chia-Hao Chang, Cheng-Wu Chen. 2011. *“Developing a Tour Guiding Information System for Tourism Service Using Mobile Gis and GPS Techniques”*. Taiwan
- [MTF-13] Maria Teresa Francoso, Diogenes Cortijo Costa, Mariana Mello Valin and Rogerio Rodrigues Amarente. 2013. *“Use of Open Source For The Development of Web GIS for Accessibility to Tourist Attractions”*. Sao Paolo
- [AM-10] Adnan, M., Singleton, A.D., and Longley, P.A. 2010, *“Developing Efficient Web-based GIS Applications”*. LONDON
- [MM-14] Maclean, Malcolm. 2014. *“Leaflet Tips and Tricks Interactive Maps Made Easy”*. CALIFORNIA
- [RS-07] Rosenberg Doug and Stephens Matt. 2007. *“Use Case Driven Object Modeling with UML”*. New York
- [DWT-07] Dennis Alan, Wixom Haley Barbara and Tegarden David. 2007 *“System Analysis Design UML Version 2.0”*. NEW YORK
- [CD-08] Charter, Deny. 2008. *“Konsep Dasar Gis”*. Palembang
- [MG-12] Myers, Glenford. 2012. *“The Art Of Software Testing 3rd Edition”*. Kanada
- [HR-06] Husein, Rahmad. 2006. *“Konsep Dasar Sistem Informasi Geografis”*. Yogyakarta
- [AP-07] Rahmad, Addul. Pilouk, M. 2007. *“Spatial Data Modeling For 3D Gis”*. Malaysia
- [DI-11] Daqiqil Ibnu. 2011. *“Framework Codeigniter Best Practice”*. Pekanbaru
- [GD-07] Gumelar Dhani. 2007. *“Data Spasial”*. Bandung
- [EA-11] Ekadinata Andree. 2011. *“Sistem Informasi Geografis”*. Malang
- [JZ-10] J.G Zheng. 2010. *“Entity Relationship Diagram Basic”*. Georgia
- [BWS-14] www.bondowosokab.go.id.2014. *“wisata bondowoso”*. (Diakses tanggal 4 Juli 2014).

LAMPIRAN

