

**SISTEM INFORMASI GEOGRAFIS PARIWISATA KOTA MEDAN
MENGUNAKAN METODE ALGORITMA *DIJKSTRA***

SKRIPSI

M.FERDIANSAH RKT

NIM. 0702171050



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

**SISTEM INFORMASI GEOGRAFIS PARIWISATA KOTA MEDAN
MENGUNAKAN METODE ALGORITMA *DIJKSTRA***

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

M.FERDIANSAH RKT

NIM. 0702171050



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi
Lamp : -

Kepada Yth.
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Medan Sumatera Utara Medan

Assalamu'alaikum Wr, Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara:

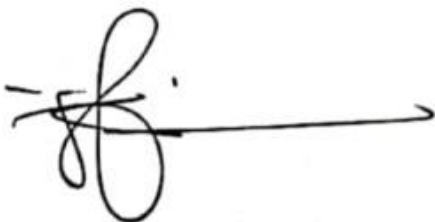
Nama	: M Ferdiansah Rkt
Nomor Induk Mahasiswa	: 0702171050
Program Studi	: Sistem Informasi
Judul	: Sistem Informasi Geografis Pariwisata Kota Medan Menggunakan Metode Algoritma Dijkstra

Dapat disetujui untuk segera *dimunaqasyahkan*. Atas perhatiannya kami ucapkan terimakasih.

Medan, 05 September 2021 M
28 Muharram 1443 H

Komisi Pembimbing

Pembimbing I



Ilka Zufria, M.Kom
NIP.198506042015031006

Pembimbing II



Aninda Muliani Harahap, M.Kom
NIP.198611292019032009

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : M.Ferdiansah Rkt

Nomor Induk Mahasiswa : 0702171050

Program Studi : Sistem Informasi

Judul : Sistem Informasi Geografis Pariwisata Kota Medan
Menggunakan Metode Algoritma *Dijkstra*

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila dikemudian hari ditemukan plagiat dalam skripsi ini saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dari sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 05 September 2021



M Ferdiansah Rkt
NIM.0702171050



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN FAKULTAS SAINS DAN TEKNOLOGI
Jl. Lap. Golf, Desa Durian Jangak, Kec. Pancur Batu 20353
Telp. (+6261) 4536090, Fax. (+6261) 6615683
Url: <http://saintek.uinsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor : B.180/ST/ST.V.2/PP.01.1/11/2021

Judul : Sistem Informasi Geografis Pariwisata Kota Medan
Menggunakan Metode Algoritma Dijkstra

Nama : M Ferdiansah Rkt

Nomor Induk Mahasiswa : 0702171050

Program Studi : Sistem Informasi

Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Sistem Informasi Fakultas Sains dan Teknologi UIN Sumatera Utara Medan Dan Dinyatakan LULUS.

Pada Hari/Tanggal : 14 September 2021

Tim Ujian Munaqasyah,

Ketua

Samsudin, ST. M.Kom
NIP. 19761 272011 11002

Dewan Penguji,

Penguji I

Samsudin, ST. M.Kom
NIP. 19761 272011 11002

Penguji III

Ilka Zulfria, M.Kom
NIP. 198506042015031006

Penguji II

Muhammad Dedi Irawan, M.Kom
NIP. 199001312019031019

Penguji IV

Aninda Muliani Harahap, M.Kom
NIP. 198611292019032009

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. Mhd. Syahnan, M. A
NIP. 196609051991031002

ABSTRAK

Dinas Pariwisata Kota Medan dalam memberikan informasi lokasi pariwisata saat ini masih menggunakan situs resmi yang hanya menampilkan penjelasan tentang wisata serta pemetaan titik lokasi wisata saja, sistem saat ini kurang efisien dan membuat para wisatawan atau masyarakat tidak banyak mengetahui tentang lokasi ataupun *route* yang akan dilalui dan pencarian tempat wisata yang ada di kota Medan. Peneliti mencoba menyelesaikan masalah yang terjadi dengan menerapkan sistem informasi geografis yang diimplementasikan dengan *OSM (Open Street Map)*. Pada penelitian ini, pengumpulan data menggunakan metode kualitatif dan aplikasi dikembangkan menggunakan metode *waterfall*. Hasil analisa akan digambarkan menggunakan *UML (Unified modelling language)*. Hasil penelitian ini menunjukkan bahwa sistem informasi geografis pariwisata untuk mengetahui titik lokasi dan *route* terdekat dapat memberikan manfaat kepada wisatawan dan masyarakat setempat untuk berwisata di kota medan.

kata kunci : sistem informasi geografis, *open street map*, *waterfall*, pariwisata

ABSTRACT

The Medan City tourism office in providing information on tourism locations currently still uses the official website which only displays explanations about tourism and mapping of tourist location points, the current system is less efficient and makes tourists or the public not much aware of the location or route to be traversed and search for tourist attractions in the city of Medan. Researchers try to solve the problems that occur by implementing a geographic information system that is implemented with OSM (Open Street Map). In this study, data collection used qualitative methods and applications were developed using the waterfall method. The results of the analysis will be described using the Unified modeling language. The results of this study indicate that a tourism geographic information system to find out the closest location points and routes can provide benefits to tourists and local communities to travel in the city of Medan.

keywords : geographic information system, open street map, waterfall, tourism

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah Subhanahu wata'ala, dengan segala karunia-Nya yang telah diberikan kepada saya, untuk menyusun skripsi yang berjudul **“Sistem Informasi Geografis Pariwisata Kota Medan Menggunakan Metode Algoritma Dijkstra”**.

Shalawat dan beriring salam, marilah kita panjatkan shalawat kepada Nabi kita yang luar biasa Muhammad Sallallahu 'Alaihi Wasallam yang telah membimbing kita dengan pelajaran-pelajaran Islam yang telah di didiknya.

Pembuatan skripsi ini merupakan ketentuan yang telah ditetapkan oleh Universitas Islam Negeri Sumatera Utara Medan, bahwa untuk mahasiswa wajib memenuhi syarat-syarat dalam memperoleh gelar Sarjana Komputer dari program S1 Sistem Informasi.

Dalam kesempatan ini penulis dengan tulus dan ikhlas tidak lupa mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberi dukungan dan membantu penulis dalam penyelesaian proposal skripsi ini, penulis ingin mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Syahrin, M.A. selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.
2. Bapak Dr. Mhd. Syahnan, M.A. selaku Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan.
3. Bapak Samsudin, ST., M.Kom. selaku Ketua Program Studi Sistem Informasi UIN Sumatera Utara Medan.
4. Bapak Suendri, M.Kom. selaku Sekretaris Program Studi Sistem Informasi UIN Sumatera Utara Medan.
5. Bapak Ilka Zufria, M.Kom. selaku Dosen Pembimbing I, yang telah membimbing saya selama pembuatan skripsi.
6. Ibu Aninda Muliani Harahap, M.Kom. selaku Dosen Pembimbing II, yang telah membimbing saya selama pembuatan skripsi.

7. Orang tua saya serta abang dan adik-adik kandung saya, terimakasih atas segala *support*, kasih sayang, do'a dan cinta serta pengorbanannya kepada saya.
8. Kepada diri sendiri yang telah bersungguh-sungguh untuk menyelesaikan skripsi ini.
9. Dina Ayu Wardani, Dinary Dwi Hatami, Dwi Nenda Putri dan seluruh teman-teman yang tergabung dalam Program Studi Sistem Informasi'5 angkatan 2017 yang selalu mendukung dan memberikan semangat.

Akhir kata penulis berharap semoga proposal skripsi ini bermanfaat bagi kita semua, khususnya bagi mahasiswa/I Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.

Medan, 14 September 2021

Penulis



M Ferdiansah Rkt

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Sistem	6
2.2 Informasi.....	7
2.3 Sistem Informasi.....	10
2.3.1 Komponen Sistem Informasi	11
2.4 Geografi	14
2.5 Sejarah Sistem Informasi Geografis	14
2.6 Sistem Informasi Geografis	15
2.6.1 Komponen Sistem Informasi Geografis	17
2.6.2 Karakteristik Sistem Informasi Geografis	19
2.6.3 Manfaat Sistem Informasi Geografis.....	19
2.6.4 Subsistem Sistem Informasi Geografis.....	20
2.6.5 Fungsi Analisis Sistem Informasi Geografis	22

2.6.6 Sumber Kesalahan Dalam SIG	23
2.7 <i>WebGIS (Website Geographic Information System)</i>	24
2.8 Pengertian Pariwisata	25
2.8.1 Objek Wisata	26
2.8.2 Akomodasi (Penginapan)	27
2.8.3 Oleh-oleh Khas Daerah	28
2.8.4 <i>Event</i>	28
2.9 Kota Medan	29
2.10 Pariwisata Kota Medan.....	31
2.11 Algoritma <i>Dijkstra</i>	31
2.12 Sistem Database.....	33
2.12.1 Pengertian Sistem Database.....	33
2.12.2 Komponen Database.....	33
2.12.3 Abstraksi Data	34
2.13 <i>OSM (Open Street Map)</i>	34
2.14 <i>LeafletJS</i>	35
2.15 <i>UML (Unified Modelling Language)</i>	35
2.15.1 Bangunan Dasar <i>UML</i>	38
2.15.2 <i>Use Case Diagram</i>	39
2.15.3 <i>Activity Diagram</i>	41
2.15.4 <i>Sequence Diagram</i>	43
2.15.5 <i>Class Diagram</i>	45
2.16 <i>PHP (Hypertext Preprocessor)</i>	47
2.17 <i>MySQL</i>	48
2.18 <i>HTML (Hypertext Markup Language)</i>	50
2.19 <i>CSS (Cascade Style Sheet)</i>	51
2.20 <i>Framework Laravel</i>	52
2.21 <i>Waterfall</i>	52
2.22 Studi Sejenis	53
BAB III METODE PENELITIAN	56

3.1 Tempat Penelitian	56
3.2 Waktu Penelitian.....	56
3.3 Kebutuhan Sistem.....	58
3.3.1 Perangkat Keras	58
3.3.2 Perangkat Lunak	58
3.4 Cara Kerja.....	58
3.4.1 Metode Pengumpulan Data	59
3.4.2 Jenis Data.....	59
3.4.3 Metode Pengembangan Sistem.....	60
3.5 Kerangka Berpikir	61
BAB IV HASIL DAN PEMBAHASAN	62
4.1 <i>Requirement Planning</i>	62
4.1.1 Profil Dinas Pariwisata Kota Medan	62
4.1.1.1 Struktur Organisasi Dinas Pariwisata Kota Medan.....	63
4.1.2 Analisis Sistem Yang Sedang Berjalan	63
4.1.2.1 Prosedur Sistem Yang Sedang Berjalan.....	63
4.1.2.2 Identifikasi Masalah Sistem Yang Sedang Berjalan	64
4.1.3 Analisis Sistem Usulan	64
4.1.4 Analisis Algoritma <i>Dijkstra</i> SIG Pariwisata Kota Medan.....	65
4.2 <i>Design</i>	70
4.2.1 Desain Proses.....	70
4.2.1.1 <i>Use Case</i> Diagram	70
4.2.1.2 <i>Activity</i> Diagram	71
4.2.1.3 <i>Sequence</i> Diagram.....	78
4.2.2 Desain <i>Database</i>	82
4.2.2.1 <i>Class</i> Diagram.....	82
4.2.2.2 Spesifikasi Tabel <i>Database</i>	83
4.2.3 Desain Rancangan Antarmuka (<i>Interface</i>)	88
4.2.3.1 Struktur Menu	89
4.2.3.2 Perancangan Tampilan	89

4.3 <i>Implementation</i>	108
4.4 <i>Verification</i>	108
4.5 <i>Maintenance</i>	116
BAB V KESIMPULAN DAN SARAN	118
5.1 Kesimpulan	118
5.2 Saran	118
DAFTAR PUSTAKA	119

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
2.1	Sejarah Singkat Sistem Informasi Geografis	15
2.2	Implementasi Prinsip SIG dengan <i>Web</i>	31
2.3	Simbol <i>Use Case</i> Diagram	39
2.4	Simbol <i>Activity</i> Diagram	42
2.5	Simbol <i>Sequence</i> Diagram	43
2.6	Simbol <i>Class</i> Diagram	46
2.7	Studi Sejenis	53
3.1	Jadwal Penelitian.....	56
4.1	Daftar Lokasi <i>Rute</i>	66
4.2	Hasil Iterasi Ke-1	66
4.3	Hasil Iterasi Ke-2	67
4.4	Hasil Iterasi Ke-3	67
4.5	Hasil Iterasi Ke-4	68
4.6	Hasil Iterasi Ke-5	68
4.7	Hasil Iterasi Ke-6	68
4.8	Hasil Iterasi Ke-7	69
4.9	Atribut Users	83
4.10	Atribut Religis	83
4.11	Atribut Sejarahs.....	84
4.12	Atribut Kuliners	85
4.13	Atribut Buatans	85
4.14	Atribut Syariahs	86
4.15	Atribut Umums.....	87
4.16	Atribut Oleh_olehs	87
4.17	Atribut Events	88
4.18	Pengujian Metode <i>Black Box</i> Level <i>User</i>	108
4.19	Pengujian Metode <i>Black Box</i> Level Admin	111

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
2.1	Siklus Informasi	9
2.2	Blok Sistem Informasi Yang Berinteraksi	12
2.3	Komponen Sistem Informasi Geografis	18
2.4	Arsitektur Sistem <i>WebGIS</i>	25
2.5	Contoh Menentukan Jalur Terdekat Algoritma <i>Dijkstra</i>	32
2.6	Jenis-jenis Diagram <i>UML</i>	37
2.7	Contoh Desain <i>Use Case</i> Diagram.....	41
2.8	Contoh Desain <i>Activity</i> Diagram.....	43
2.9	Contoh Desain <i>Sequence</i> Diagram.....	45
2.10	Contoh Desain <i>Class</i> Diagram	47
2.11	Metode <i>Waterfall</i>	52
3.1	Metode <i>Waterfall</i>	60
3.2	Kerangka Berpikir	61
4.1	Struktur Organisasi Dinas Pariwisata Kota Medan.....	63
4.2	<i>Graf</i> Contoh Kasus Algoritma <i>Dijkstra</i>	65
4.3	<i>Use Case</i> Diagram SIG Pariwisata Kota Medan.....	70
4.4	<i>Activity</i> Diagram <i>Login</i> Admin	71
4.5	<i>Activity</i> Diagram Admin Wisata	72
4.6	<i>Activity</i> Diagram Admin Penginapan.....	73
4.7	<i>Activity</i> Diagram Admin Oleh-oleh.....	74
4.8	<i>Activity</i> Diagram Admin <i>Event</i>	75
4.9	<i>Activity</i> Diagram Admin <i>Logout</i>	76
4.10	<i>Activity</i> Diagram <i>User</i>	76
4.11	<i>Activity</i> Diagram <i>User</i>	77
4.12	<i>Sequence</i> Diagram Admin <i>Register</i> dan <i>Login</i>	78
4.13	<i>Sequence</i> Diagram Admin Wisata.....	78
4.14	<i>Sequence</i> Diagram Admin Wisata Edit Dan Hapus	79
4.15	<i>Sequence</i> Diagram Admin Penginapan	79
4.16	<i>Sequence</i> Diagram Admin Penginapan Edit Dan Hapus	80

4.17	<i>Sequence</i> Diagram Admin Oleh-oleh.....	80
4.18	<i>Sequence</i> Diagram Admin Oleh-oleh Edit Dan Hapus	81
4.19	<i>Sequence</i> Diagram Admin <i>Event</i>	81
4.20	<i>Sequence</i> Diagram <i>User</i>	82
4.21	<i>Class</i> Diagram SIG Pariwisata Kota Medan	82
4.22	Struktur Menu SIG Pariwisata Kota Medan	89
4.23	Tampilan Beranda	90
4.24	Tampilan Halaman Wisata	90
4.25	Tampilan Halaman <i>List</i> Wisata Religi	91
4.26	Tampilan Halaman <i>List</i> Wisata Sejarah.....	91
4.27	Tampilan Halaman <i>List</i> Wisata Kuliner.....	92
4.28	Tampilan Halaman <i>List</i> Wisata Buatan.....	92
4.29	Tampilan Halaman Penginapan	93
4.30	Tampilan Halaman <i>List</i> Penginapan Umum	93
4.31	Tampilan Halaman <i>List</i> Penginapan Syariah	94
4.32	Tampilan Halaman Oleh-oleh	94
4.33	Tampilan Halaman <i>List</i> Oleh-oleh	95
4.34	Tampilan Halaman <i>Event</i>	95
4.35	Tampilan Halaman <i>List Event</i>	96
4.36	Tampilan Halaman <i>About</i>	96
4.37	Tampilan Halaman <i>Login</i> Admin.....	97
4.38	Tampilan Halaman <i>Register</i>	97
4.39	Tampilan Halaman Admin Wisata	98
4.40	Tampilan Halaman Tabel Verifikasi & Tampil Wisata Religi	98
4.41	Tampilan Halaman Tabel Verifikasi & Tampil Wisata Sejarah	99
4.42	Tampilan Halaman Tabel Verifikasi & Tampil Wisata Kuliner	99
4.43	Tampilan Halaman Tabel Verifikasi & Tampil Wisata Buatan	100
4.44	Tampilan Halaman <i>Update</i> Wisata Religi	100
4.45	Tampilan Halaman <i>Update</i> Wisata Sejarah	101
4.46	Tampilan Halaman <i>Update</i> Wisata Kuliner	101
4.47	Tampilan Halaman <i>Update</i> Wisata Buatan	102

4.48	Tampilan Halaman Admin Penginapan	102
4.49	Tampilan Halaman Tabel Verifikasi & Tampil P.Umum	103
4.50	Tampilan Halaman Tabel Verifikasi & Tampil P.Syariah	103
4.51	Tampilan Halaman <i>Update</i> Penginapan Umum Admin	104
4.52	Tampilan Halaman <i>Update</i> Penginapan Syariah Admin	104
4.53	Tampilan Halaman Admin Oleh-oleh	105
4.54	Tampilan Halaman Tabel Verifikasi & Tampil Oleh-Oleh	105
4.55	Tampilan Halaman <i>Update</i> Oleh-oleh Admin	106
4.56	Tampilan Halaman Admin <i>Event</i>	106
4.57	Tampilan Halaman Tabel Tampil <i>Event</i> Admin	107
4.58	Tampilan Halaman <i>Update Event</i> Admin	107

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran	Halaman
I	Surat Izin Riset Balitbang	123
II	Surat Izin Riset Dinas Pariwisata Kota Medan	124
III	Formulir Hasil Wawancara	125
IV	Formulir <i>Black Box Testing</i>	126
V	<i>Source Code</i>	134

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dengan pesatnya perkembangan teknologi saat ini, *GIS (Geographic Information System)* tidak hanya menjadi *trend* dalam teknologi *kartografi* itu sendiri, tetapi juga telah menjadi salah satu kebutuhan teknis untuk memecahkan masalah pemerintahan. Hal ini dikarenakan SIG dapat digunakan sebagai referensi untuk menampilkan data geografis secara akurat. Potensi wisata Medan beragam dan sudah mencapai unsur 3A (atraksi, amenitas dan aksesibilitas). Tempat wisata di kota Medan terbagi menjadi dua kategori yaitu daya tarik tempat wisata dan daya tarik tempat kegiatan (*event*) wisata. Menurut pakar Soekadijo, pariwisata merupakan fenomena yang kompleks di masyarakat, yang meliputi hotel, objek wisata, cinderamata, pemandu wisata, angkutan wisata, biro perjalanan, restoran, dll.

Dinas Pariwisata Kota Medan adalah suatu instansi yang bergerak dalam bidang pariwisata. Pada saat ini Dinas Pariwisata masih mempromosikan wisata-wisatanya yang masih belum cukup tersebar luas ke kalangan wisatawan maupun masyarakat yang disebabkan penyebaran informasi wisata atau *event* masih menggunakan cara lama seperti melalui brosur, media massa serta media sosial lainnya. Cara ini tidak cukup untuk mempromosikan pariwisata secara luas kepada wisatawan dan masyarakat kota Medan karena gambaran kawasan wisata tidak dapat diperoleh, seperti visualisasi lokasi, jarak kawasan wisata dengan jalan yang akan dilalui, dan sulit bagi wisatawan dan masyarakat yang belum mengetahui untuk menentukan rencana perjalanan. Dinas Pariwisata Kota Medan saat ini juga masih melakukan pencarian tempat wisata melalui situs resmi yang hanya berupa penjelasan tentang wisata serta pemetaan titik lokasi wisata saja, dan hanya terdapat beberapa wisata saja yang memiliki pemetaan titik lokasi wisata. Hal ini membuat para wisatawan atau masyarakat tidak banyak mengetahui tentang lokasi ataupun *route* yang akan dilalui dan pencarian tempat wisata yang ada di kota Medan.

Salah satu pendekatan untuk memutuskan jalur terpendek adalah dengan menguraikan panduan sebagai diagram (grafik). Ada beberapa teknik yang bisa digunakan untuk menentukan jarak terpendek. Salah satu strategi untuk melacak jalan yang paling terbatas adalah perhitungan *Dijkstra*. Perhitungan ini digunakan untuk diagram arah pada setiap titik yang dilintasi dan dikaitkan dengan tepi tertimbang. Dengan mempertimbangkan bobot masing-masing pihak, perhitungan tersebut dapat digunakan untuk menentukan jalan tersingkat dari puncak tujuan. Perhitungan *Dijkstra* lebih terkonsentrasi secara komputasi ketika mencari cara terbaik di jaringan internet, waktu eksekusi normal perhitungan *Dijkstra* lebih terbatas daripada perhitungan algoritma lainnya, sehingga dibandingkan dengan algoritma lain, algoritma *Dijkstra* banyak digunakan dikarenakan lebih besar mendapatkan informasi *rute* yang terpendek.

Berdasarkan dari permasalahan diatas, Dinas Pariwisata Kota Medan sangat memerlukan suatu sistem untuk mempromosikan wisata-wisata Kota Medan beserta *rute* terpendek. Untuk mengatasi hal tersebut perlu dibuat perencanaan yang matang dalam membuat Sistem Informasi Geografis (SIG) menggunakan *OSM* (*Open Street Map*) melalui *Library LeafletJs* dalam mencari tempat pariwisata kota Medan dan menggunakan Algoritma *Dijkstra* dalam pencarian *rute* terpendek. Peneliti merencanakan suatu kerangka kerja yang dapat memberikan data total tentang destinasi pariwisata di kota Medan, seperti tempat penginapan di kota Medan, oleh-oleh khas kota Medan, dan informasi *event* yang ada di kota Medan serta memberikan informasi *rute* terpendek yang akan dilalui oleh para wisatawan dan masyarakat yang ingin liburan ke lokasi pariwisata kota Medan. Penelitian ini akan dilakukan sebagai laporan skripsi dengan judul: **"Sistem Informasi Geografis Pariwisata Kota Medan Menggunakan Metode Algoritma *Dijkstra*"**.

1.2 Rumusan Masalah

Berdasarkan landasan tersebut, dapat disimpulkan rumusan masalah yang di dapat dalam penelitian, antara lain:

1. Bagaimana membuat sistem informasi geografis yang dapat memudahkan

wisatawan dalam mencari informasi wisata, tempat penginapan, oleh-oleh khas, dan mendapatkan informasi *event* yang diadakan dengan menggunakan *OSM (Open Street Map) Library LeafletJs*?

2. Bagaimana cara peneliti untuk membuat wisatawan dan masyarakat menentukan *rute* terpendek dari satu objek wisata ke objek wisata lainnya dengan menerapkan algoritma *Dijkstra*, dan kemudian menentukan *rute* terpendek dari posisi dimana wisatawan dan masyarakat berada?

1.3 Batasan Masalah

Untuk membahas dan menyusun laporan ini secara tepat sasaran dan memahami dengan jelas situasinya, maka perlu dirumuskan batasan-batasan masalah, yaitu:

1. Ruang lingkup pengumpulan data dan tempat penelitian dilakukan di Dinas Pariwisata Kota Medan.
2. Pada SIG ini akan memberikan informasi *rute* terdekat yang terkait pada objek wisata, tempat penginapan, oleh-oleh khas, dan informasi *event* yang diadakan.
3. SIG ini akan memberikan rekomendasi *event*, tempat penginapan, oleh-oleh khas yang ada disekitar objek wisata yang dipilih.
4. Pada sistem ini tidak hanya admin yang dapat melakukan *input* pariwisata kota Medan, tetapi *user* juga dapat akses untuk *input* pariwisata kota Medan yang belum terdata di sistem informasi geografis.
5. Perancangan sistem dilakukan dengan menggunakan bahasa pemrograman *HTML, PHP, Mysql, Framework laravel 8*, dan menggunakan Metode Pengembangan *Waterfall*, serta menggunakan *OSM(Open Street Map) Library LeafletJs*.
6. Pencarian jalur (*rute*) terpendek menggunakan metode algoritma *Dijkstra*.
7. Sistem berupa Aplikasi berbasis *Web GIS*.

1.4 Tujuan Penelitian

Adapun tujuan peneliti melakukan penelitian skripsi ini untuk:

1. Menghasilkan SIG (Sistem Informasi Geografis) lokasi wisata Medan beserta tempat penginapan, oleh-oleh khas Medan, dan *event* yang diadakan di kota Medan berbasis *Web GIS* menggunakan *OSM (Open Street Map)*.
2. Wisatawan dan masyarakat kota Medan dapat dengan mudah menemukan lokasi *route* terpendek ke tempat wisata, tempat penginapan, oleh-oleh khas serta *event* yang diadakan di kota Medan dengan menggunakan Algoritma *Dijkstra*.

1.5 Manfaat Penelitian

Manfaat penelitian yang diharapkan dalam pembuatan Sistem Informasi Geografis Pariwisata Kota Medan yaitu:

1. Manfaat Bagi Peneliti
 - a. Dapat mengaplikasikan ilmu-ilmu yang telah didapatkan semasa perkuliahan di Program Studi Sistem Informasi.
 - b. Dapat menganalisa sistem yang sedang berjalan pada Dinas Pariwisata Kota Medan.
 - c. Dan memberikan sebuah sistem usulan serta untuk memenuhi salah satu syarat kelulusan S1 Program Studi Sistem Informasi.
2. Manfaat Bagi Program Studi
 - a. Dapat mengetahui kemampuan mahasiswa dalam menguasai teori dan praktek yang diperoleh selama kuliah.
 - b. Serta memberikan bahan bagi pembaca atau peneliti selanjutnya.
3. Manfaat Bagi Universitas
 - a. Dapat mengetahui kemampuan dari mahasiswa dalam menerapkan ilmu yang telah didapatkan sebagai bahan evaluasi.
4. Manfaat Bagi Dinas Pariwisata Kota Medan
 - a. Mendapatkan sumbangan ide dalam usaha menarik masyarakat dan wisatawan luar sehingga dapat menambahkan *income* Pemerintah Daerah Kota Medan.
 - b. Sistem Informasi Geografis ini dapat membantu Dinas Pariwisata Kota Medan untuk mempromosikan pariwisata yang berada di kota Medan dan

beserta *route* terpendek.

5. Manfaat Bagi Wisatawan dan Masyarakat

- a. Memudahkan untuk memperoleh informasi letak wisata, tempat penginapan, oleh-oleh khas, dan *event* yang biasanya diadakan Dinas Pariwisata Kota Medan, dan mempermudah mereka untuk mencari lokasi *route* terdekat ke tempat wisata, tempat penginapan, oleh-oleh khas, dan *event*.

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Sistem mempunyai pengertian yang beragam. Banyak sumber dan pakar yang telah menyampaikan pengertian dari sistem. Menurut Jogiyanto dalam penelitiannya Fanny Andilia, Ada dua kumpulan pendekatan dalam mengkarakterisasi kerangka kerja sistem (Andalia & Setiawan, 2015). Ada yang menekankan pada prosedurnya, ada juga yang lebih menekankan pada prosedurnya, ada juga yang lebih menekankan pada komponennya (Setiawan, 2020). Kerangka kerja sistem pada dasarnya adalah berbagai komponen yang paling berdampak dalam menyelesaikan latihan bersama untuk mencapai tujuan tertentu, Kerangka kerja sistem juga memiliki atribut atau properti tertentu, sebagai berikut:

1. Komponen (*Components*)

Kerangka kerja sistem terdiri berasal berbagai bagian yang saling berhubungan, bekerja sama menghasilkan suatu solidaritas. Batas sistem (*Boudary*) ruang lingkup merupakan wilayah yang titik potong antara suatu sistem dengan sistem yg tidak sinkron atau menggunakan iklim eksternal. pembatasan sistem ini memungkinkan sistem buat ditinjau menjadi unit yang tidak bisa dibagi.

2. Lingkungan luar (*Environtment*)

Apa pun di luar batas sistem yang memengaruhi aktivitas sistem. Iklim di luar sistem ini dapat berharga dan juga dapat menjadi tidak nyaman bagi sistem.

3. Penghubung sistem (*Interface*)

Media mengasosiasikan satu subsistem dengan subsistem yang berbeda untuk membingkai unit soliter sehingga aliran aset dimulai dengan satu subsistem lalu ke subsistem berikutnya.

4. Masukan sistem (*Input*)

Energi yang masuk ke dalam sistem disebut masukan sistem, uang tunai dapat sebagai pendukung (*Maintenance Input*) dan sinyal (*Signal Input*).

5. Keluaran sistem (*Output*)

Efek samping dari energi yang ditangani dan diatur menjadi konsumsi yang

berharga, hasil ini merupakan kontribusi untuk subsistem yang berbeda.

6. Pengolahan Sistem (*Process*)

Merupakan bagian yang siklus kontribusi untuk berubah menjadi hasil yang ideal.

7. Sasaran Sistem (*Objective*)

Setiap sistem harus memiliki tujuan atau target yang mempengaruhi sumber data yang diperlukan dan hasil selanjutnya (Hendra, 2012).

Penilaian dari beberapa spesialis yang mencirikan mengenai sistem sebagai berikut:

1. Menurut Jogiyanto bahwa sistem adalah bermacam-macam komponen yang berinteraksi untuk mencapai tujuan tertentu (Hutahaeen, 2014).
2. Seperti yang ditunjukkan oleh Jerry Futz Gerald bahwa sistem yang sebenarnya menyiratkan berbagai komponen yang bekerja sama untuk menyelesaikan suatu tujuan (Hutahaeen, 2014).
3. Menurut Umar Fahmi, sistem adalah suatu rangkaian tindakan yang menggambarkan adanya suatu perkembangan dari bagian-bagian yang berbeda yang memiliki hubungan yang harmonis, tersusun dan tujuan bersama yang bekerja atau berjalan dalam jangka waktu tertentu dan tersusun (Adil, 2017).
4. Seperti yang dikemukakan Indrajit, sistem mengandung pengertian bermacam-macam bagian yang memiliki komponen-komponen yang saling berkaitan satu sama lain (Hutahaeen, 2014).

Melihat penerangan pada atas, cenderung beralasan bahwa sistem artinya rangkaian tindakan berasal beberapa segmen, item, serta komponen yang diminta dan mempunyai korelasi yang direncanakan buat mencapai suatu tujuan.

2.2 Informasi

informasi yaitu data akan menjadi isu yang telah disusun menjadi suatu struktur yang mempunyai arti krusial bagi penerima manfaat dan bisa menjadi fenomena, suatu nilai yang berguna. dengan cara ini, pada hubungan mengganti informasi menjadi data terjadi *input-process-output* (Hendra, 2012). Berdasarkan

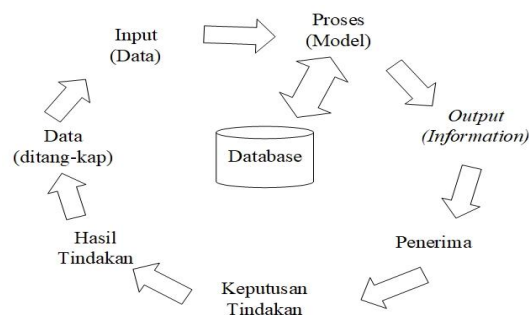
(Nugroho, 2017) Secara etimologis, istilah data asal dari istilah Prancis kuno *informacion* (tahun 1387) mengambil istilah dari bahasa Latin, khususnya *informationem* yang berarti “inspirasi, pemikiran atau diagram”. Data ini ialah sesuatu berasal *informare* yang mengandung pengertian tindakan. Secara lebih eksplisit, itu tidak lain adalah gerakan dalam "informasi yang disampaikan".

Seperti yang ditunjukkan oleh Notoadmodjo, bahwa semakin banyak data yang Anda miliki, semakin Anda dapat memengaruhi atau memperluas wawasan Anda tentang seseorang dan dengan informasi ini Anda dapat mengungkapkan masalah bahwa pada akhirnya seorang akan sinkron menggunakan informasi yang mereka miliki, cenderung bisa disimpulkan bahwa data merupakan “aneka macam realitas yang sudah disusun menjadi informasi, sebagai akibatnya cenderung lebih bernilai dan dapat dimanfaatkan oleh setiap individu yang membutuhkan informasi tersebut sebagai informasi atau dapat dimanfaatkan secara dinamis”. Data cenderung dianggap sebagai “bermacam-macam realitas yang telah ditangani ke dalam jenis informasi, sehingga cenderung lebih membantu dan dapat dimanfaatkan oleh setiap individu yang membutuhkan informasi sebagai informasi atau dapat dimanfaatkan secara dinamis” (Nugroho, 2017). Dari beberapa definisi yang dikemukakan sang para pakar, cenderung diasumsikan bahwa data merupakan informasi yang sudah disusun serta mengakibatkan informasi tadi lebih sempurna, signifikan serta berharga buat kelanjutannya secara dinamis. Menurut beberapa ahli bahwa informasi yaitu:

1. Sebagaimana dikemukakan oleh Anton M. Moeliono (1990), informasi adalah penerangan, keterangan, pemberitahuan, kabar atau berita.
2. Seperti yang ditunjukkan oleh Gordon B. Davis (1984), data akan menjadi informasi yang telah ditangani/disiapkan ke dalam struktur yang sangat signifikan bagi penerima manfaat dan benar-benar bernilai atau dirasakan dalam aktivitas atau pilihan saat ini atau di masa depan.
3. Menurut Kenneth C. Laudon (Gaol, 2008), data akan menjadi informasi yang telah dibentuk menjadi struktur yang berharga dan berguna bagi orang-orang.

4. Menurut Tsichritis dan Lochovsky (1970), informasi adalah pentingnya hubungan dan terjemahan informasi yang memungkinkan individu untuk memutuskan.
5. Menurut Synanski dan Pulschen (1995), informasi adalah persiapan informasi yang jelas dalam pengaturan untuk menyampaikan pentingnya kepada orang lain.

Berdasarkan penilaian sang para pakar di atas, sudah diduga bahwa data merupakan informasi yang telah disiapkan atau ditangani untuk membuat informasi tersebut menjadi informasi yang tepat, penting dan berharga untuk siklus dinamis. Sebuah data dianggap penting jika keuntungannya lebih menguntungkan daripada biaya untuk mendapatkannya. Sedangkan sifat data bergantung pada tiga hal, yaitu data harus tepat, tepat, dan aplikatif. *Precise* menyiratkan bahwa data harus secara jelas mencerminkan apa yang disiratkannya. Tepat waktu menyiratkan bahwa data yang muncul di penerima tidak boleh terlambat. Berlaku menyiratkan bahwa data memiliki manfaat bagi klien. Informasi ditangani melalui contoh tertentu sebagai data yang bisa dimanfaatkan sang penerima manfaat dalam menentukan pilihan serta mengambil langkah, yang berarti membentuk langkah lain yang akan membuat beberapa informasi balik . Data yang masih belum terolah akan disimpan ke pada sebuah bentuk database. informasi yang disimpan ini nantinya dapat dipulihkan buat diproses ulang menjadi data. informasi akan ditangkap sebagai informasi, diproses ulang melalui contoh tertentu, dll buat membingkai siklus. siklus ini dianggap oleh John Burch menggunakan siklus informasi (information cycle).



Gambar 2.1 Siklus Informasi (Riyanto, Ekaputra & Indelarko, 2019)

Pada siklus informasi di atas memiliki penjelasan tentang proses alurnya untuk berjalannya siklus informasi, sebagai berikut:

1. Sebuah data di input buat kebutuhan proses data.
2. Proses data akan membentuk data dasar yang kemudian dikeluarkan menjadi output (informasi).
3. Sesudah menjadi output data diteruskan ke penerima buat diambil keputusan tindakan dan ditindak lanjuti.
4. Hasil dari aktivitas selanjutnya akan balik ke informasi baru yang lalu bisa diproses ulang.

2.3 Sistem Informasi

Seperti yang dikutip Jogiyanto pada bukunya “Analisis dan Desain Sistem informasi” Robert A. Leitch serta K. Roscoe Davis mendefinisikan Sistem informasi menjadi berikut: “Sistem informasi ialah kerangka kerja di dalam suatu asosiasi yang menyatukan setiap hari pertukaran mempersiapkan kebutuhan, menjunjung tinggi kegiatan, bersifat administratif dan juga latihan penting asosiasi dan memberikan laporan fundamental kepada pihak luar tertentu”.

Untuk sementara itu sesuai kata (Rachman & Maulana, 2012) Sistem informasi adalah sekelompok bagian yang paling terkait yang bekerja sama buat mengumpulkan, memproses, menyimpan serta mendistribusikan informasi terkait buat mendukung proses pengambilan keputusan, koordinasi, serta pengendalian. Menurut (Kadir, 2002) menyimpulkan bahwa sebagian dari penilaian beberapa spesialis sehubungan dengan arti sistem informasi adalah (Kadir, 2014):

1. Seperti yang ditunjukkan oleh Alter (1992), Sistem Informasi merupakan deretan antara metodologi kerja, data, individu, serta penemuan data yang dikoordinasikan buat mencapai tujuan pada suatu asosiasi.
2. Seperti yang ditunjukkan oleh Gelinas, Oram, dan Wiggins (1990), Sistem informasi ialah suatu sistem kerja sintesis insan yang sebagian besar terdiri berasal sekelompok segmen berbasis PC serta manual yg didesain untuk

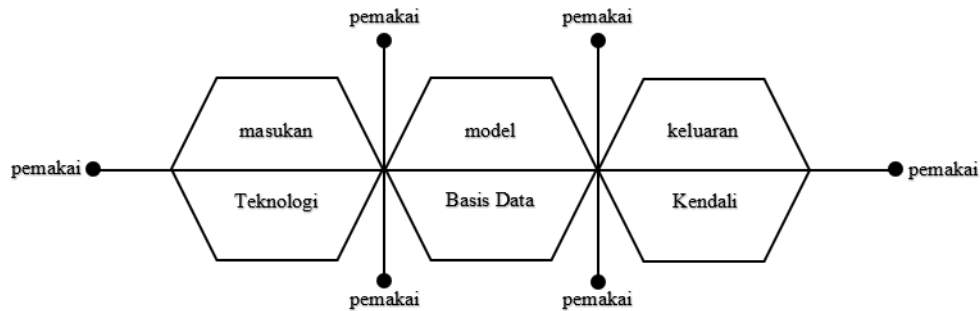
mengumpulkan, menyimpan, dan mengawasi informasi dan memberikan data, dan menyebarkannya ke klien.

3. Seperti yang ditunjukkan oleh Turban, McLean, dan Wetherbe (1999), Sistem Informasi adalah mengumpulkan, memproses, menyimpan, analisis, dan menyebarkan data untuk tujuan eksplisit.
4. Seperti yang ditunjukkan oleh Wilkinson (1992), Sistem Informasi adalah struktur yang mengatur aset (manusia komputer) untuk mengubah *input*.
5. Sesuai yang ditunjukkan oleh Alter (1992), Sistem Informasi adalah perpaduan sistem kerja, data, individu, dan inovasi data yang dikoordinasikan untuk mencapai tujuan di dalam suatu asosiasi.
6. Seperti yang ditunjukkan oleh Hall (2001), Sistem Informasi adalah sekelompok metode formal dimana informasi dikumpulkan, ditangani menjadi data, dan disebarkan ke klien.
7. Seperti yang ditunjukkan oleh Bodnar dan Hopwood (1993), Sistem Informasi adalah bermacam-macam peralatan dan pemrograman yang dimaksudkan untuk mengubah informasi menjadi data yang berharga.

Sistem Informasi adalah kerangka kerja di dalam asosiasi yang menyatukan pertukaran sehari-hari menyiapkan kebutuhan yang membantu elemen operasional administrasi asosiasi dengan kegiatan penting dari sebuah asosiasi untuk memiliki opsi untuk memberikan pertemuan eksternal yang pasti data yang diperlukan untuk dinamis (Anggraeni & Irviani, 2017).

2.3.1 Komponen Sistem Informasi

John Burch dan Gary Grudnitski percaya bahwa suatu sistem informasi terdiri dari apa yang disebutnya *building block*, yaitu blok *input*, blok model, blok keluaran dan blok teknis (*block*), blok basis data, dan blok kontrol. Masuki sistem di mana enam blok berinteraksi satu sama lain untuk menghasilkan unit mereka sendiri untuk mencapai tujuan mereka (Riyanto, Ekaputra & Indelarko, 2019).



Gambar 2.2 Blok Sistem Informasi Yang Berinteraksi

Selanjutnya adalah penggambaran bagian-bagian yang disinggung sebagai blok bangunan sistem informasi, sebagai berikut:

1. Blok Masukan yaitu *input* mewakili data informasi yang memasuki kepada data sistem informasi. kontribusi disini menggabungkan strategi serta media buat menangkap informasi yang akan dimasukkan, yang bisa menjadi laporan mendasar.
2. Blok Model yaitu blok ini terdiri dari campuran metode, alasan serta model numerik yang akan mengontrol informasi serta informasi yang disimpan pada perpaduan data menggunakan tujuan tertentu pada pikiran buat menciptakan hasil yang ideal.
3. Blok Keluaran yaitu hasil dari sistem informasi adalah hasil yang adalah data berkualitas serta dokumentasi berharga buat semua taraf dewan serta seluruh klien dan seluruh pengguna sistem.
4. *Block Technology*, yaitu teknologi yang merupakan *toolbox* dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output*, dan membantu kontrol sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama, yaitu *humanware & brainware*, *software* dan *hardware*. Seorang teknisi dapat menjadi orang yang memahami teknologi dan membuatnya bekerja, misalnya teknisi adalah operator komputer pribadi, programmer, operator pengolah kata, ahli telekomunikasi, analis sistem, penyimpan data, dll.

5. Blok Basis data adalah sejenis basis data, serangkaian sumber data yang saling terkait, disimpan pada perangkat keras komputer pribadi dan digunakan oleh perangkat lunak untuk mengoperasikannya. Data perlu disimpan dalam *database* untuk memberikan informasi lebih lanjut. Data dalam *database* perlu diatur dengan cara ini untuk mendapatkan informasi berkualitas tinggi. Organisasi *database* yang baik juga berguna untuk kapasitas penyimpanan yang efisien. Mengakses atau memanipulasi basis data menggunakan aplikasi paket yang dianggap sebagai *DBMS (Database Management System)*.
6. Blok Kendali yaitu Banyak hal yang dapat membahayakan sistem informasi, seperti peristiwa bencana, kebakaran, suhu, air, debu, kesalahan penyajian, kekecewaan sistem informasi, kesalahan, kegagalan, kerusakan, dll. Beberapa kontrol harus direncanakan dan dilaksanakan untuk menjamin bahwa hal-hal yang dapat membahayakan sistem informasi dapat dicegah atau jika kesalahan terjadi, mereka dapat segera ditangani. (Riyanto, Ekaputra & Indelarko, 2019).

Hanya saja, Sistem informasi menjadi semakin kompleks, termasuk banyak pelanggan, membutuhkan kantor jaringan untuk memungkinkan pelanggan yang didistribusikan di tempat yang berbeda untuk berbagi data. Dengan cara ini, hal-hal yang seharusnya dimungkinkan oleh sistem informasi diidentifikasi dengan kemampuan yang mereka miliki dan dapat mereka lakukan (Riyanto, Ekaputra & Indelarko, 2019). Sesuai menurut Turban, McLean dan Wetherbe, sebagai berikut:

1. Menyediakan komunikasi dalam organisasi atau antar organisasi yang murah, akurat, dan cepat.
2. Menyimpan informasi dalam jumlah yang sangat besar dalam ruang yang kecil tetapi mudah diakses.
3. Memungkinkan pengaksesan informasi yang sangat banyak di seluruh dunia dengan cepat dan murah.
4. Meningkatkan efektifitas dan efisiensi orang-orang yang bekerja dalam kelompok pada suatu tempat atau beberapa lokasi.
5. Menyajikan informasi dengan jelas yang menggugah pikiran manusia.

6. Mengotomasi proses-proses bisnis yang semi-otomatis dan tugas-tugas yang dikerjakan secara manual.
7. Mempercepat pengetikan dan penyuntingan.
8. Pembiayaan yang lebih murah daripada mengerjakan secara manual.

2.4 Geografi

Geografi berasal dari bahasa Yunani, khususnya *geo(s)* dan *graphein*. *Geo(s)* menyiratkan bumi sementara *graphein* bermaksud untuk menggambarkan, menggambarkan, atau gambar. Geografi adalah ilmu yang mengkaji atau mempelajari bumi dan segala isinya seperti tumbuhan, fauna, lingkungan, individu, dan setiap kerjasamanya. Juga, Geografi adalah ilmu yang meninjau kesamaan dan kontras dalam keajaiban geosfer dari perspektif ekologi dan lokal dalam pengaturan spasial. Secara singkat geografi itu ilmu pengetahuan yang menggambarkan tentang kebumihan (Setiawan, 2020). Oleh karena itu, geografi membahas tentang gejala dan karakteristik tanah dan penataan penghuninya menurut lokasinya, serta berupaya menjelaskan keterkaitan antara fenomena dan karakteristik tersebut.

2.5 Sejarah Sistem Informasi Geografis

Sistem informasi geografis pertama kali dikembangkan sang Tomlinson pada pertengahan 1960an untuk menyimpan, mengontrol, serta memecah informasi yang dikumpulkan untuk Cadangan Tanah Kanada. Pada tahun 1970, perusahaan kartografi terbesar telah memupuk interaksi perencanaan yang dimodernisasi. Sistem Informasi Geografis mulai dimanfaatkan di awal tahun 1980 ketika harga perangkat keras komputer semakin terjangkau. Munculnya *Graphical User Interface (GUI)* membawa wilayah yang lebih luas sistem informasi geografis dan membawa sistem informasi geografis menjadi penggunaan yang tendensi di awal 1980 (Adil, 2017).

Ini pertama kali diperkenalkan pada Indonesia di tahun 1972 menggunakan nama *Data Banks For Development*. Seperti di berbagai negara, di Indonesia kemajuan sistem informasi geografis dimulai di kalangan otoritas publik dan

militer. Kemajuan sistem informasi geografis telah berkembang pesat sejak ditopang oleh aset yang bergerak dalam iklim skolastik. Tabel 2.1 berikut merupakan sejarah singkat dari sistem informasi geografis.

Tabel 2.1 Sejarah Singkat Sistem Informasi Geografis (Setiawan, 2020)

<i>Mainframe</i> 1960s	<i>Mini Mainframe</i> 1970s	<i>Micro Mini Workstation Mainframe</i> 1980s	<i>Client/Server Workstation Mini Micro</i> 1990s	<i>Online (2008 - Sekarang)</i>
<i>Prototype GIS CGIS</i>	<i>Commercial GIS ODYSSEY</i>	<i>Custodial GIS ARC/INFO INTEGRAPH STRINGS</i>	<i>Desktop GIS PC ARC/INFO ARCVIEW MAPINFO IRIDISI</i>	<i>API Google Maps API & MapBox</i>

2.6 Sistem Informasi Geografis

Dijelaskan oleh (Oktarian & Usman, 2016) bahwa BAKOSURTANAL (Badan Koordinasi Survei dan Pemetaan Nasional) menjabarkan bahwa sistem informasi geografis merupakan berbagai macam peralatan *PC*, pemrograman, informasi geografis, dan staf yang terkoordinasi yang dimaksudkan untuk memperoleh, menyimpan, memperbaiki, mengontrol, menyelidiki, dan menampilkan semua jenis data geografis (Hamdi et al., 2018). Sistem informasi geografis adalah alat yang berharga untuk mengumpulkan, menyimpan, dan mengambil data yang diperlukan dan menampilkan data keuangan dari dunia nyata (Adil, 2017).

Secara singkat sistem informasi geografis ialah kerangka kerja data *PC* yang dipergunakan buat menangani informasi data yang berafiliasi dengan informasi geografis. Sistem informasi geografis menampilkan informasi pada bentuk grafis serta umumnya menggunakan peta menjadi tampilan antarmuka (Setiawan, 2020).

Sebagian besar *Geographic Information System (GIS)* memakai konsep “lapis” (*layer*). Setiap lapisan menangani satu komponen geografis pada daerah yang sama serta sesudah itu seluruh lapisan bisa ditumpuk pada atas satu sama lain untuk menerima data total. Setiap lapisan bisa diklaim menjadi plastik sederhana yang hanya berisi gambar-gambar tertentu. Pelanggan dapat memilih mana yang sederhana dan lalu ditumpuk satu sama lain sebagai akibatnya diperoleh gambar yang ialah campuran aneka macam plastik. *Geographic Information System (GIS)* sesungguhnya itu salah satu jenis dari *DSS (Decision Support Systems)* atau Sistem Pendukung Keputusan (SPK).

Pengguna dapat memilih transparansi yang diinginkan, dan kemudian saling menguntungkan dengan ditumpangkan bersama sehingga gambar digabungkan menjadi beberapa plastik sederhana (transparan). Sebenarnya Sistem Informasi Geografis (SIG) adalah sejenis *DSS*. Oleh karena itu, Menurut Martin (2002) *GIS* terkadang disebut sebagai sistem pendukung keputusan spasial (*SDSS*). *GIS* saat ini bahkan dapat mengkonsolidasikan tugas dinamis dalam pengambilan keputusan, misalnya:

1. Lacak jalur yang paling terbatas atau tercepat dari posisi A ke posisi B.
 2. Memutuskan apakah ada area berbeda yang memiliki contoh komparatif.
 3. Mengelompokkan daerah penjualan untuk meminimalkan jarak perjalanan.
- Dijelaskan (Kirom, 2014) ada beberapa informasi dengan kualitas geografis, antara lain:
- a. Data *Spasial*: informasi yang menyinggung perihal posisi, objek, serta hubungan pada antara mereka di ruang bumi. informasi *spasial* ialah salah satu segmen data yang didalamnya ada data perihal bumi, contohnya bagian atas dunia, bawah permukaan dunia, perairan, laut serta bawah iklim.
 - b. Data *Non-spasial*: informasi berkualitas yang menggambarkan panduan atau data pada panduan.

Menurut (Immasari, 2018) Bujur (*longitude*) adalah garis bujur yang menghubungkan sisi utara dan selatan (kutub) bumi, digunakan untuk mengukur susunan timur-barat suatu titik di ekuator bumi. Garis lintang (*latitude*) merupakan

perpotongan antara kutub utara dan selatan yang menghubungkan sisi timur dan barat bumi.

Sistem Informasi Geografis (SIG) seperti yang ditunjukkan oleh para ahli:

1. Menurut Chrisman (Adil, 2017) GIS merupakan sistem yang terdiri dari perangkat keras, perangkat lunak, data, *brainware*, asosiasi, dan yayasan yang digunakan untuk mengumpulkan, menyimpan, menyelidiki, dan menyebarkan data tentang wilayah di permukaan dunia.
2. Menurut Aronoff (Adil, 2017) GIS merupakan sistem berbasis PC yang dapat menangani informasi yang dirujuk secara geologis, khususnya bagian informasi, informasi papan (penyimpanan dan peninjauan), kontrol dan penyelidikan informasi dan hasil sebagai produk akhir. Hasil akhir dapat digunakan sebagai sumber perspektif dalam menciptakan pilihan pada isu-isu yang diidentifikasi dengan topografi.

2.6.1 Komponen Sistem Informasi Geografis

Komponen dari sistem informasi geografis menurut R. Astrini dan P. Oswald (Setiawan, 2020) yaitu:

1. Perangkat Keras

Merupakan perangkat fisik atau *hardware* yang dapat mendukung dalam pemetaan dan analisis geografi. Perangkat keras ini mempunyai kemampuan untuk menampilkan citra maupun gambar dengan resolusi yang tinggi sehingga informasi yang ditampilkan lebih detail, serta mempunyai kemampuan untuk mengolah data yang terdapat dalam sistem informasi geografis. Karena menyimpan informasi keruangan, maka perangkat keras dari sebuah sistem informasi geografis harus memiliki media penyimpanan dan pemrosesan yang besar.

2. Perangkat Lunak

Perangkat lunak atau *software* pada sistem informasi geografis merupakan aplikasi yang memiliki kemampuan untuk mendapatkan, menyimpan, memperbaiki, mengontrol, memeriksa, dan menampilkan informasi spasial atau

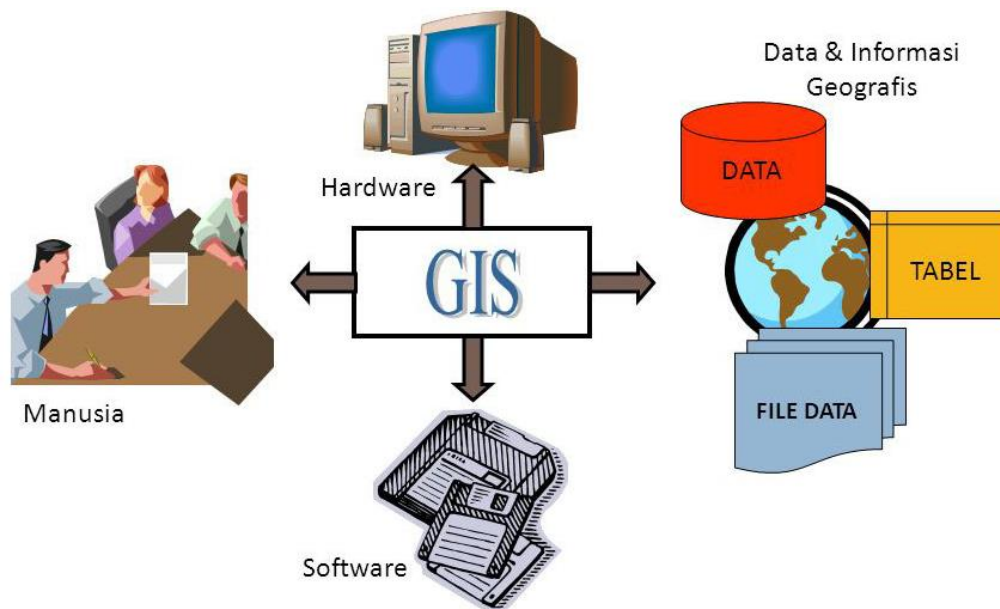
data non spasial.

3. Data

Data dapat dikatakan sebagai sebuah elemen yang tidak mempunyai arti apapun bagi penggunaannya. Untuk itu, sebuah data harus diproses terlebih dahulu agar menjadi sebuah informasi yang dapat dimengerti. Dalam konteks sistem informasi geografis, data dapat dibedakan menjadi data *spasial* dan data *non-spasial*. Data spasial adalah penggambaran keajaiban permukaan dunia atau keruangan yang akan diolah menjadi sistem informasi geografis dengan tujuan tertentu. Data spasial berkaitan dengan posisi pada koordinat tertentu. Berbeda dengan data *spasial*, data *non-spasial* memperkenalkan bagian dari penggambaran atau klarifikasi keajaiban di permukaan dunia sebagai kata-kata, angka, atau tabel. Data *non-spasial* berfungsi untuk menggambarkan keajaiban geografis karena memiliki sudut pandang yang menarik dan subjektif.

4. Brainware

Komponen terakhir dari sistem informasi geografis yaitu manusia yang merupakan seseorang yang menjalankan, membuat, atau orang yang memperoleh manfaat dari adanya sebuah sistem informasi geografis.



Gambar 2.3 Komponen Sistem Informasi Geografis

2.6.2 Karakteristik Sistem Informasi Geografis

Sistem informasi geografis memiliki beberapa karakteristik yang umum. Menurut Husen R (2006) dalam bukunya (Nirwansyah, 2017) beberapa karakteristik yang umum dimiliki oleh sistem informasi geografis di antaranya:

1. Merupakan sistem hasil pengembangan dari perangkat keras dan perangkat lunak untuk tujuan pemetaan.
2. Termasuk melibatkan ahli geografi, informatika, dan komputer, serta aplikasi terkait.
3. Isu yang dikembangkan yang menggabungkan inklusi, kualitas dan norma informasi, struktur, persepsi dan model informasi, koordinasi kelembagaan dan moral, pendidikan, *expert system*, *decision support system*, serta penerapannya.
4. Data terkait dengan wilayah geografis dan terdiri dari data informasi tekstual maupun grafik.
5. Bukan hanya sekedar merupakan pengubahan peta konvensional ke dalam bentuk peta digital untuk kemudian disajikan, dicetak atau diperbanyak kembali.
6. Cocok untuk mengumpulkan, menyimpan, mengubah, menunjukkan, mengendalikan, menggabungkan, dan memeriksa informasi spasial dari keajaiban topografi suatu daerah.
7. Siap untuk menyimpan informasi penting yang diharapkan untuk mengatasi suatu masalah. Contoh penanganan suatu isu adalah perubahan lingkungan yang membutuhkan data fundamental seperti curah hujan, suhu, angin, kondisi awan. Informasi standar umumnya dikumpulkan sesekali dan sepanjang rentang waktu yang luas.

2.6.3 Manfaat Sistem Informasi Geografis

Sistem informasi geografis mengkombinasikan informasi berupa data *spasial* dan data *non-spasial* dari berbagai sumber menjawab berbagai macam kebutuhan. Menurut Kraak dan Ormeling (2007) dalam bukunya (Nirwansyah, 2017) menyatakan bahwa terdapat beberapa pertanyaan yang mampu dijawab dengan dibangunnya sistem informasi geografis di antaranya:

1. “Ada apa disana?” pertanyaan ini bisa dijawab dengan cara menetapkan area pada panduan peta, nama, atau data lain pada panduan peta yang akan diperkenalkan.
2. “Dimana?” pertanyaan ini muncul dalam satu atau lebih lokasi yang mengikuti kriteria dari kondisi pertanyaan tersebut. Ini dapat berupa sekumpulan kordinat sebuah panduan peta yang menunjukkan area dari suatu objek.
3. “Sejak kapan berubah?” pertanyaan ini mengambil komponen *geospasial* yang berberkaitan dengan waktu. Contohnya pertanyaan yang berhubungan dengan perubahan perkembangan pembangunan suatu kota dari tahun ke tahun.
4. “Rute mana yang terbaik?” dalam mencari jalur optimal, sistem informasi geografis dapat membantu mencari jalur optimal yang bisa berupa jarak terdekat, kondisi lalu lintas serta biasa yang dikeluarkan.
5. “Hubungan apa yang terjadi?” pertanyaan ini lebih kompleks dan sering membutuhkan banyak data *geospasial*. Contohnya hubungan antara iklim mikro, wilayah jalur produksi dan konstruksi sosial yang meliputi iklim.
6. “Apa jika?” pertanyaan ini berhubungan dengan kegiatan perencanaan dan peramalan. Contohnya apa yang dibutuhkan untuk membuat jaringan transportasi publik di sebuah wilayah tertentu.

2.6.4 Subsistem Sistem Informasi Geografis

Adapun subsistem dalam Sistem Informasi Geografis yang perlu diketahui (Riyanto, Ekaputra & Indelarko, 2019), sebagai berikut:

1. *Input*, tahap ini *input* (pemasukan data) melakukan untuk mengumpulkan dan menyiapkan informasi spasial dan tambahan berkualitas dari berbagai sumber informasi. Informasi yang digunakan harus diubah menjadi konfigurasi lanjutan yang sesuai, siklus transformasi dikenal sebagai interaksi digitalisasi. Salah satu strategi untuk mengubah informasi sederhana menjadi informasi lanjutan adalah digitalisasi dengan menggunakan mesin digitizer, termasuk model digitalisasi di layar dari informasi pemotretan (baik foto yang ditinggikan maupun foto satelit) melalui penyaringan.

2. Manipulasi, atau pengendalian informasi adalah siklus mengubah informasi yang telah dimasukkan, hal ini dilakukan untuk mengubah jenis dan jenis informasi agar sesuai dengan kerangka yang akan dibuat, misalnya keseimbangan skala, mengubah kerangka proyeksi, spekulasi, dll
3. Manajemen Data, Tahap ini menggabungkan semua latihan yang diidentifikasi dengan persiapan informasi (menyimpan, mengatur, mengawasi, dan memecah informasi) ke dalam kerangka penimbunan yang langgeng, misalnya: sistem *file server* atau *database server* sesuai kebutuhan sistem. Jika menggunakan sistem *file server*, data disimpan dalam bentuk *file-file* seperti: *.txt, *.dat, dan lain-lain. Sedangkan jika menggunakan *database server*, biasanya itu memanfaatkan *Software Database Management System (DBMS)*, seperti: *Mysql*, *SQL Server*, *Oracle*, dan *DBMS* sejenis lainnya.
4. *Query*, strategi pencarian data untuk menjawab pertanyaan yang diajukan oleh klien *GIS*. pada *GIS* dengan sistem *file server*, *query* dapat dimanfaatkan dengan bantuan *compiler* atau *interpreter* yang digunakan dalam mengembangkan sistem, sedangkan untuk *GIS* dengan sistem *database server*, dapat memanfaatkan *SQL (Structured Query Language)* yang terdapat pada *DBMS* yang digunakan.
Melihat melalui penggunaan informasi di luar apa yang satu lapisan dapat memberikan data untuk penyelidikan informasi dan memperoleh informasi yang ideal, misalnya:
 - a. Jenis tanah apa yang dominan di hutan tertentu?
 - b. Berapa jumlah kelurahan di suatu wilayah tertentu?
 - c. Daerah mana yang umumnya cocok untuk pemukiman baru?
 - d. Kota mana yang memiliki curah hujan paling tinggi?
5. Analisis, ada dua macam kapasitas investigasi dalam SIG, yaitu kapasitas pemeriksaan spasial dan pemeriksaan sifat. Kapasitas pemeriksaan spasial adalah tugas yang dilakukan pada informasi spasial. Sementara itu, pekerjaan penyidikan properti merupakan unsur penyiapan informasi karakteristik, khususnya informasi yang tidak diidentikkan dengan ruang. Kapasitas untuk menyelidiki informasi spasial untuk memperoleh data baru. Pembuatan model

skenario “*What If*” salah satu fasilitas yang banyak digunakan yaitu analisis tumpang susun peta (*Overlay*).

6. *Visualisasi*, menampilkan hasil sebagai data baru atau basis informasi yang sudah ada, baik dalam bentuk *softcopy* maupun dalam versi cetak, misalnya seperti: peta (*map ascribes and information credits*), tabel, diagram dan lain-lain.

2.6.5 Fungsi Analisis Sistem Informasi Geografis

Fungsi SIG juga dapat dibedakan dari kapasitas pemeriksaan yang biasanya dapat dilakukan. Sesuai dengan ide informasi, ada dua macam kapasitas investigasi dalam GIS, khususnya kapasitas pemeriksaan spasial dan investigasi sifat (kumpulan data berkualitas). Pada dasarnya ada dua jenis analisis yakni analisis *proximity* dan analisis *overlay*. Secara umum, ada dua jenis fungsi analisis dalam GIS (Hutagaol & Sudarsono, 2015), yaitu:

1. Fungsi analisis atribut (*non-spasial*) antara lain, terdiri dari kegiatan dasar kerangka administrasi basis informasi dan augmentasinya.
2. Fungsi analisis *spasial* memiliki 6, yaitu:

- a. *Buffering*

Buffering adalah kapasitas yang terdapat dalam SIG yang menghasilkan informasi spasial baru sebagai poligon dengan pemisahan tertentu dari informasi spasial yang merupakan informasi. Pemeriksaan ini digunakan untuk menentukan zona penyangga suatu ruang, garis/lintasan.

- b. *Overlay*

Overlay adalah kapasitas dalam Sistem Informasi Geografis (SIG) yang memberikan lapisan lain informasi spasial yang merupakan efek lanjutan dari perpaduan dua lapisan yang menjadi informasi, diselesaikan dengan menggabungkan setidaknya dua panduan di wilayah yang sama.

- c. *Network*

Penyelidikan *spasial* diidentifikasi dengan kerangka organisasi (pemeriksaan jaringan), khususnya pengembangan atau pengembangan aset dimulai dengan satu area kemudian ke area berikutnya melalui komponen buatan yang

menyusun organisasi yang saling berhubungan satu sama lain (seperti sungai, jalan, dan sebagainya), garis, tautan, dan sebagainya).

d. *Find Distance*

Inspeksi ruang semacam ini melibatkan hubungan atau kedekatan komponen ruang dengan komponen ruang lainnya. Kemampuan inspeksi ini mengenali lapisan vektor yang berisi komponen spasial jenis titik, garis, atau poligon sebagai informasi untuk mengirimkan lapisan raster, dan pikselnya berisi nilai jarak dari semua komponen spasial yang terdapat dalam lapisan informasi.

e. *Clustering*

Clustering adalah *loop* karakterisasi yang digunakan untuk mengelompokkan piksel gambar hanya berdasarkan sudut pandang yang diukur. Pemeriksaan ini juga bertujuan untuk mengelompokkan benda menurut kualitasnya, sehingga protes yang paling dekat dengan benda yang berbeda akan berada dalam kelompok yang sama.

f. *Interpolasi*

Interpolasi adalah strategi untuk mengevaluasi nilai-nilai ambigu dengan menggunakan nilai-nilai yang diketahui di sekitarnya, nilai-nilai ini dapat direncanakan secara hati-hati secara berkala atau secara sporadis. Sifat hasil interpolasi tergantung pada akurasi yang diketahui dan sebaran fokus serta kemampuan numerik yang digunakan untuk mengevaluasi model untuk mencapai nilai yang wajar. Perhitungan numerik dalam sistem informasi geografis (SIG), panduan untuk mendapatkan hasil berdasarkan dimensi ideal dalam struktur spasial.

2.6.6 Sumber Kesalahan Dalam SIG

Untuk mendapatkan hasil Sistem Informasi Geografis hebat dan mencerahkan, penting untuk mencegah kesalahan dalam persiapan GIS, dalam hal apa pun potensi kesalahan sekecil apa pun dapat dibatasi. Sumber kesalahan dalam SIG (Riyanto, Ekaputra & Indelarko, 2019), yaitu:

1. Sumber kesalahan yang jelas adalah usia informasi, wilayah setengah jalan atau lengkap, skala, ketebalan persepsi, kepentingan, desain, keterbukaan, biaya.

2. Kesalahan yang berasal dari varietas biasa (Normal Varietas) dan sumber perkiraan pertama, yaitu presisi posisi spesifik, ketepatan substansi subjektif, varietas dalam (bagian informasi, predisposisi penonton, varietas reguler).
3. Kesalahan yang terjadi karena siklus tersebut adalah kesalahan numerik PC, kesalahan dalam pemeriksaan topologi, masalah karakterisasi, dan spekulasi.

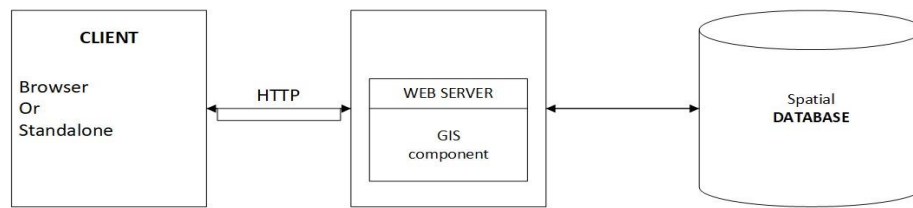
2.7 WebGIS (*Website Geographic Information System*)

WebGIS adalah sistem informasi geografi berbasis *web* terdiri dari beberapa bagian yang saling berhubungan. *Web Gis* adalah campuran dari rencana perencanaan yang realistis, peta terkomputerisasi dengan pemeriksaan geografis, pemrograman PC, dan kumpulan data yang saling berhubungan menjadi satu bagian dari arsitektur situs *web* dan perencanaan *web*. Secara umum Sistem Informasi Geografis dibuat tergantung pada aturan input/masukan data, manajemen, analisis dan representasi data. Dalam bidang web, prinsip-prinsip ini digambarkan dan dijalankan seperti yang ditampilkan dalam tabel terlampir:

Tabel 2.2 Implementasi Prinsip SIG dengan *Web*

Prinsip SIG	Pengembangan <i>Web</i>
<i>Data Input</i>	Klien
Manajemen Data	<i>DBMS</i> dengan komponen spasial
Analisis Data	<i>GIS Library</i> di <i>Server</i>
Representasi Data	Klien/ <i>server</i>

Agar dapat membuat koneksi dengan berbagai bagian lingkungan *web*, *server web* diperlukan. Karena standar prinsip data geografis menonjol dan benar, desain dan peningkatan arsitektur sistem rekayasa mengikuti arsitektur “Server Klien”.



Gambar 2.4 Arsitektur Sistem *WebGIS*

Gambar di atas memperlihatkan arsitektur minimal sebuah sistem *WebGIS*. Aplikasi berada disisi klien yang berkomunikasi dengan *server* sebagai penyedia data melalui *web* protokol seperti *HTTP* (*Hyper Text Transfer Protocol*) (Ramadhani & Awaluddin, 2016).

2.8 Pengertian Pariwisata

Masih banyak aset tetap, budaya, kuliner dan daya tarik normal pada daerah di Indonesia yang belum terkelola seperti yang diharapkan dan yang mengejutkan masih belum jelas bagi banyak individu dengan pengaturan pemerintah dibantu oleh penduduk di sekitar ruang yang bersangkutan bergabung untuk mengembangkan potensi ini. jadi membayar dari bidang industri perjalanan untuk meningkatkan ukuran perdagangan asing bangsa bekerja secara positif.

Sesuai Undang-Undang Nomor 10 Tahun 2009 tentang Perindustrian Perjalanan Wisata, Industri Perjalanan Wisata adalah bermacam-macam kegiatan industri perjalanan wisata dan diselenggarakan oleh kantor-kantor dan administrasi yang diberikan oleh daerah setempat, wisatawan perorangan, otoritas publik, pemerintah lingkungan dan para visioner bisnis. Bagian dari industri perjalanan itu sendiri merupakan bidang yang dapat menopang kemajuan suatu daerah, terutama dengan pedoman tentang kemandirian provinsi. Pengaturan ini ditegaskan berdasarkan jaringan terdekat yang memiliki modal yang dapat diandalkan untuk kemajuan daerahnya, salah satunya adalah kegiatan industri perjalanan wisata. (Rahma, 2020). Beberapa menurut beberapa ahli tentang pariwisata, antara lain:

1. Menurut UU RI No.9 Tahun 1990, pariwisata adalah semua yang diidentikkan dengan latihan gerak yang dilakukan dengan sengaja dan singkat untuk

mengapresiasi barang dan atraksi wisatawan, termasuk penyalahgunaan barang dan atraksi wisatawan, serta organisasi terkait di bidang itu.

2. Menurut Richard Sihite, pariwisata adalah latihan perjalanan yang diselesaikan untuk sementara waktu atau sebentar untuk tujuan selain menghasilkan uang.
3. Menurut Mathieson dan Wall (1982), pariwisata adalah kemajuan latihan sebagai pengembangan singkat individu ke tujuan di luar tempat rumah atau lingkungan kerja mereka, latihan yang mereka selesaikan selama kunjungan mereka di tujuan dan kantor yang diberikan untuk mengatasi masalah mereka baik selama jalan-jalan maupun di area tujuan.
4. Menurut Soekadijo (1996), pariwisata adalah pariwisata merupakan gejala yang bertautan di masyarakat, yang meliputi hotel, objek wisata, cinderamata, pemandu wisata, angkutan wisata, biro perjalanan, restoran, dll.
5. Seperti dikemukakan oleh Suyitno (2001) tentang pariwisata adalah:
 - a. Ini sifatnya sementara, bahwa dalam sekejap para penghibur industri perjalanan akan kembali ke tempat asal mereka.
 - b. Termasuk beberapa bagian industri perjalanan, misalnya kantor transportasi, kenyamanan, restoran, tujuan liburan, hadiah dan lain-lain.
 - c. Memiliki tujuan tertentu yang pada dasarnya adalah untuk mendapatkan kesenangan.
 - d. Bukan untuk menghasilkan uang di tempat tujuan, malah kenyataannya bisa menambah bayaran untuk daerah atau daerah yang dikunjungi, karena uang yang dikeluarkan dibawa dari awal.

2.8.1 Objek Wisata

Objek wisata yaitu semua yang ada di obyek wisata itu adalah daya tarik sehingga wisatawan harus tetap pada tempatnya dan semua tempat atau kondisi biasa yang memiliki aset industri perjalanan yang dirangkai dan dikembangkan sehingga memiliki daya tarik dan dikembangkan sebagai tempat yang dikunjungi wisatawan. Barang-barang industri perjalanan dapat berupa objek wisata biasa seperti gunung, sungai, danau, lautan, pantai atau sebagai bangunan seperti galeri, benteng, tujuan warisan yang dapat diverifikasi, dll, objek wisata kadang-kadang

disebut tempat liburan atau daerah wisata yang ditandai untuk menetapkan daerah yang berarti untuk perjalanan. (Nurwanto, 2020).

2.8.2 Akomodasi (Penginapan)

Sebagaimana tercantum dalam Pedoman Menteri Pariwisata dan Ekonomi Inventif Nomor PM.53/HM.001/MPEK/2013 tentang Norma Usaha Penginapan/Tempat Tinggal, yang dimaksud dengan penginapan/hotel adalah soal memberikan kenyamanan sebagai kamar di struktur yang dapat dilengkapi dengan administrasi makanan. Selain itu, minum, olahraga pengalihan, dan berbagai pekerjaan lainnya selalu bertekad untuk menghasilkan keuntungan. Sedangkan menurut referensi kata besar di Indonesia, hotel/motel berarti gedung bertingkat, menyewakan sebagai tempat tinggal adalah menyediakan tempat makan, minum, dan istirahat bagi yang telah menyelesaikan atau menyelesaikannya perjalanan mereka (Susepti et al., 2017).

Menurut Shi Te (2000:62), orang pergi ke lebih dari sekedar akomodasi atau istirahat. Tujuan akhirnya adalah pertemuan, kursus, pesta, pernikahan, perayaan ulang tahun, pertunjukan, dan kegiatan lain yang memerlukan penataan kantor yang lengkap. terlebih lagi, keuntungan yang dapat memuaskan pengunjungnya. Dengan demikian, penginapan/hotel sebagai fasilitas kenyamanan bisnis dapat berfungsi sebagai tempat peristirahatan serta jenis kantor bantuan umum (Susepti et al., 2017).

Menurut Arief (2005:76-78) dalam mendukung kemajuan suatu negara, bisnis penginapan dapat berperan dalam berbagai hal (Susepti et al., 2017), antara lain:

1. Memperluas lapangan kerja industri perorangan.
2. Buat posisi baru.
3. Membantu otoritas publik dan area privat dalam pengajaran dan persiapan usaha.
4. Kenaikan gaji provinsi/negara bagian melalui wilayah penilaian.
5. Meningkatkan perdagangan/pembayaran asing di negara tersebut.
6. Meningkatkan hubungan antar negara di planet ini.

2.8.3 Oleh-Oleh Khas Daerah

Oleh-oleh adalah sesuatu yang dibawa seorang musafir ke rumahnya untuk kenangan yang berhubungan dengan hal itu, contohnya seperti suvenir, makanan (buah, makanan khas, dll) dan sebagainya.

Khas Daerah merupakan sesuatu dimana masing-masing daerah memiliki keunikan dari berbagai macam hal mulai dari benda hingga makanan khas dari daerah tersebut.

Oleh-oleh Khas Daerah diartikan sebagai segala sesuatu yang dapat dijadikan buah tangan atau kenangan yang dihasilkan dari daerah tersebut, dimana para wisatawan dapat membawa pulang ke tanah kelahiran tempat mereka tinggal untuk menjadi sebuah kenangan bagi para wisatawan.

2.8.4 *Event*

Manajemen *event* adalah bagian dari ilmu papan yang membuat dan mengembangkan gerakan yang sepenuhnya ditujukan pada acara sosial individu di satu tempat, memainkan pengaturan standar latihan untuk memperoleh data atau menyaksikan suatu acara, beberapa kelompok memanggil acara para eksekutif sebagai fitur dari usaha naik. Namun, selain itu, dengan melihat latihan yang mempengaruhi banyak individu dan dari sudut pandang organisasi, kesempatan diingat untuk penelitian menampilkan korespondensi seperti publikasi dan promosi. Kadang-kadang dewan harus dipertimbangkan karena latihannya dipengaruhi oleh korespondensi, khususnya teknik visual dan pesan, korespondensi antara orang-orang, pilihan dan situasi media, eksposur dan kemajuan dan beberapa latihan yang berbeda. (Kusuma, 2016).

Menurut Goldblatt (Noor, 2013) bahwa *Event* Manajemen adalah latihan-latihan mahir mengumpulkan dan menyatukan perkumpulan individu dengan tujuan akhir pesta, sekolah, pertunjukan, dan reuni, dan bertanggung jawab untuk mengarahkan ujian, merencanakan latihan, mengatur dan melakukan koordinasi dan pengawasan untuk memahami kehadiran suatu gerakan. Dalam hal ini para eksekutif dapat dicirikan sebagai mendapatkan suatu gerakan yang diawasi secara

ahli, sengaja, cakap, dan memadai. Latihannya menggabungkan ide (keinginan) untuk eksekusi hingga pengawasan, jadi kadang-kadang semua dewan harus bekerja keras dengan visi untuk membuat latihan yang memenuhi asumsi.

Dalam arti sempit, *event* diartikan sebagai tampilan presentasi atau perayaan dengan syarat ada koordinator, anggota dan tamu. Bagaimanapun, dari perspektif yang luas, acara tersebut dicirikan sebagai waktu pergerakan yang diselesaikan oleh sebuah asosiasi dengan membawa individu ke suatu tempat dengan tujuan agar mereka mendapatkan data atau pengalaman penting seperti tujuan lain yang diharapkan oleh koordinator.

2.9 Kota Medan

Kota Medan adalah ibu kota Sumatera Utara yang terletak di pesisir timur Sumatera Utara. Pendahulu Kota Medan sudah ada sejak tahun 1590. Dalam pergantian peristiwanya, Kota Medan mengalami penyesuaian waktu pemerintahan, dari kekuasaan Kesultanan Melayu Toko, Provinsi Hindia Belanda, belum lama ini. ternyata penting bagi Republik Indonesia. Peninggalan dari masa pemerintahan ini umumnya tersebar di kota Medan. Pola kemajuan, perubahan lingkungan di seluruh dunia, globalisasi moneter, dan modernisasi dapat memengaruhi adegan sosial dan yang dapat diverifikasi (Rezky Nur Awalia, 2017). Kehadiran adegan yang dapat diverifikasi dan sosial dapat menjaga legitimasi dan karakter baru dari sebuah adegan. Pelestarian pemandangan kota yang dapat diverifikasi dan sosial harus dimungkinkan dengan merencanakan kepribadian pemandangan dan mengikuti kualitas keunikan dan realitas pemandangan otentik. (Syam et al., 2019).

Saat ini Kota Medan merupakan salah satu masyarakat perkotaan yang didorong untuk menjadi Kota Warisan di Indonesia oleh Dinas Pekerjaan Umum melalui Program Rencana Permainan dan Perlindungan Kota *Legacy* (P3KP). Program *Legacy City* adalah menjadikan ruang metropolitan yang terlindungi, nyaman, bermanfaat, dan ekonomis bergantung pada penataan ruang, yang digambarkan dengan nilai-nilai warisan, melalui perubahan upaya pelestarian

menuju peningkatan metropolitan (warisan) dengan bantuan dan administrasi besar seperti penataan kota. pondasi yang pas. . Hal ini bergantung pada Undang-Undang Perlindungan Sosial UU Penataan Ruang Nomor 11 Tahun 2010 dan Nomor 26 Tahun 2007. Saat ini banyak proyek yang dapat diverifikasi di titik pusat Kota Medan, namun masyarakat dan wisatawan relatif sedikit mengetahui proyek tersebut. Dari rencana tata ruang Kota Medan 2011-2021, Kota Medan memiliki 6 kawasan perlindungan sosial (KCB), empat di antaranya berada di ruang TPK Kota Medan (Syam et al., 2019).

KCB yang menjadi fokus pendampingan Kota Medan adalah wilayah Kesawan, wilayah Benteng Maimun, wilayah Polonia, dan wilayah kota Madras. Pada tahun 2014, terdapat sekitar 1.225 benda peninggalan otentik di Kota Medan yang sebagian besar merupakan benda tabah. Upaya pengamanan barang bukti di Kota Medan telah dilakukan oleh Pemerintah Daerah Kota Medan dengan mengeluarkan Peraturan Daerah Nomor 6 Tahun 1988 tentang Pengamanan Bangunan dan Iklim yang memiliki nilai sejarah, bangunan, arkeologi, dan penghijauan kembali di wilayah kota Medan. Konservasi artikel otentik sebagai bukti rekaman dan modal sosial memiliki potensi luar biasa dalam pola kontes industri perjalanan dunia *worldwide* (Syam et al., 2019).

Menurut (Lubis, 2018) kota yang dipisahkan oleh sumbu barat laut/tenggara ini terhubung dengan Jalan Perumahan Kerajaan menuju Jalan Kesawan. Wilayah perkotaan dikelilingi oleh beberapa kota Melayu, yang terbesar adalah kota selatan. Berada di kota terpercaya. Sedikit di utara Benteng Toko Raja adalah kota Minang Kabau. Bagian barat pusat kota, terutama di daerah Bologna antara Store dan Sungai Babura, merupakan pemukiman orang-orang Eropa yang bekerja di berbagai pemerintahan dan peternakan. Di pinggiran rumah-rumah Eropa, mereka menemukan kota Bolt, umumnya dikenal sebagai kota sekolah Islam, dimana imigran dari India utara dan selatan tinggal. Untuk sementara waktu, kota-kota Timur Tengah terletak di antara aliran toko dan pasar ikan tua, dan di sebelah timur adalah kota-kota Cina, yang ditempati oleh calo dan buruh dari Kanton, Fukui, dan Jiayingqiu.. Migran Tionghoa ini diasingkan ke dalam bahasa sehari-hari yang

berbeda, termasuk kongfu, hokkien, dan hakka dan setelah itu masing-masing dari mereka dapat bergabung melalui korespondensi dengan bahasa kuo-yu (mandarin) sebagaimana adanya.

2.10 Pariwisata Kota Medan

Pariwisata adalah potensi yang harus dimiliki oleh setiap daerah. Perbedaan kualitas setiap daerah hanyalah modal dalam kemajuan industri perjalanan itu sendiri.

Begitu pula dengan kota Medan. Beberapa keunikan lokal dapat dikembangkan sebagai daya tarik industri perjalanan, seperti bangunan terkenal, tujuan sosial, kuliner, hingga keunikan individu.

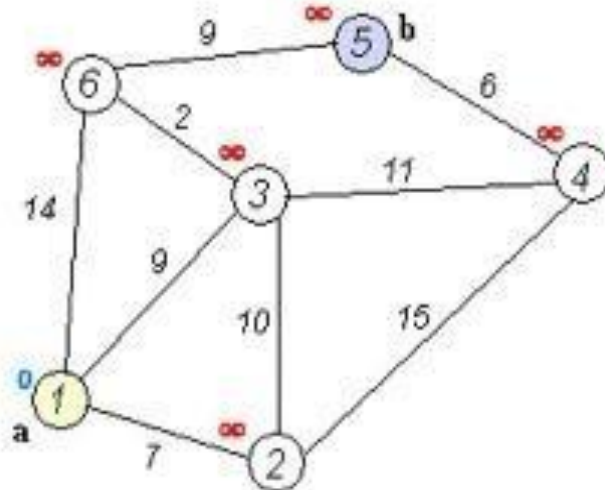
Sebagai upaya untuk memajukan industri perjalanan kota Medan, seperti beberapa komunitas perkotaan yang berbeda di Indonesia, telah menandai kotanya. Gerakan penandaan ini dimulai dari kemungkinan Dinas Kebudayaan dan Industri perjalanan Republik Indonesia (Rahmi, 2018).

2.11 Algoritma *Dijkstra*

Algoritma *Dijkstra* adalah perhitungan rakus yang digunakan untuk menangani masalah jarak paling terbatas untuk diagram terkoordinasi dengan beban tepi non-negatif. Inti dari perhitungan *Dijkstra* itu sendiri adalah untuk menemukan nilai pengeluaran yang paling dekat dengan target kapasitas pada diagram tertimbang, sehingga dapat membantu memberikan jalan keputusan. Dalam Perhitungan *Dijkstra*, hub digunakan karena Perhitungan *Dijkstra* menggunakan diagram terpandu untuk menentukan jalur jalan terpendek. Perhitungan ini berencana untuk melacak cara yang paling singkat tergantung pada beban terkecil dimulai dengan satu titik kemudian ke titik berikutnya. Asumsikan titik tersebut membahas struktur dan garis menggambarkan jalan, kemudian pada saat itu Perhitungan *Dijkstra* melakukan estimasi pada semua beban terkecil yang mungkin dari setiap titik. Pada Gambar 2.5 diagram model diberikan bobotnya dalam

menentukan cara pemanfaatan Algoritma *Dijkstra* (Ardana & Saputra, 2016). Dijelaskan oleh (Triase & Aprilia, 2020) bahwa algoritma *Dijkstra* adalah salah satu dari algoritma yang banyak dimanfaatkan dan sangat populer dalam menyelesaikan sebuah permasalahan pencarian jalur terpendek.

Tentukan terlebih dahulu titik mana yang akan menjadi *hub* awal, kemudian, pada titik tersebut bobot pemisahan dari *hub* utama ke *hub* terdekat secara individual, Perhitungan *Dijkstra* akan mendorong perburuan dimulai dengan satu titik kemudian ke titik berikutnya dan ke titik berikutnya sedikit demi sedikit.



Gambar 2.5 Contoh Menentukan Jalur Terdekat Dengan Algoritma *Dijkstra*

Logika dari sebuah perhitungan algoritma *Dijkstra*, yaitu:

1. Berikan bobot insentif (jarak) untuk satu sama lain sorot, kemudian, berikan nilai 0 ke *hub* awal dan nilai tak terbatas ke *hub* berikutnya (tidak terisi).
2. Atur semua *hub* dengan rapi dan atur *hub* awal sebagai *hub* keberangkatan.
3. Dari *hub* keberangkatan, pikirkan *hub* yang berdampingan dengan rapi dan bayangkan jaraknya dari titik keberangkatan. Misalnya, jika lepas landas langsung dari A ke B memiliki jarak 6 dan dari B ke hub C adalah 2, maka pada saat itu jarak ke C melalui B adalah $6+2=8$. Dalam hal jarak ini lebih kecil dari

jarak sebelumnya (yang telah direkam sebelumnya) hapus informasi lama, simpan kembali informasi jarak dengan jarak baru.

4. Ketika kita selesai memikirkan setiap jarak ke *hub* yang berdekatan, tandai *hub* yang dihubungi sebagai *hub* yang dihubungi. *Hub* yang dihubungi tidak akan pernah dicek lagi, jarak yang disimpan adalah jarak terakhir dan bobot terkecil.

Atur *hub* rapi dengan jarak terkecil (dari *hub* keberangkatan) sebagai *hub* keberangkatan berikut dan lanjutkan dengan kembali ke tahap 3.

2.12 Sistem Database

2.12.1 Pengertian Sistem Database

Menurut Fathansyah *system database* dicirikan sebagai kerangka kerja yang terdiri dari bermacam-macam Dokumen/formulir yang saling terkait (kumpulan data pada kerangka kerja *PC*) dan beberapa proyek (kerangka manajemen kumpulan data), memungkinkan beberapa klien atau proyek yang berpotensi berbeda untuk mendapatkan dan mengontrol catatan/formulir ini (Riyanto, Ekaputra & Indelarko, 2019). *Database* harus memiliki tiga fitur yang penting menurut Martin (1980), yaitu:

1. *Accessibility* adalah menyinggung kapasitas penerimaan untuk menyimpan atau memulihkan informasi dengan kepribadian tertentu.
2. *Generality* adalah menyinggung kapasitas untuk mendapatkan semua data untuk memulihkan atau menyesuaikan informasi.
3. *Flexibility* adalah menyinggung kapasitas dalam kenyamanan dan kemajuan *database*.

2.12.2 Komponen Database

Dalam suatu sistem basis data yang lengkap akan terdapat komponen utama yaitu perangkat keras, sistem operasi, basis data, sistem manajemen basis data (*DBMS*) dan pengguna (Riyanto, Ekaputra & Indelarko, 2019).

2.12.3 Abstraksi Data

DBMS memberikan pandangan teoretis yang memberikan untuk meninjau informasi data (*interface*) yang ramah *user* (*user friendly*). Abstraksi data ialah tingkatan bagaimana melihat data dalam sebuah sistem database. ada tiga tingkat abstraksi data, yaitu lapis fisik/internal, lapis konseptual, serta lapis pandangan pemakai (Riyanto, Ekaputra & Indelarko, 2019).

2.13 OSM (*Open Street Map*)

OSM (*Open Street Map*) merupakan sebuah proyek berbasis *web* untuk menciptakan panduan dunia yang bebas dan terbuka, seluruhnya terdiri dari sukarelawan dengan mengkoordinasikan garis besar menggunakan *GPS*, mendigitalkan citra satelit, dan mengumpulkan dan menyediakan data geografis yang tersedia tanpa syarat.

Dengan *Open Data Commons Open Database License 1.0*, *OSM* dapat menjamin, menyesuaikan, dan membagikan data peta secara luas. Ada berbagai jenis panduan canggih yang dapat diakses di *web*, Namun, kebanyakan dari mereka memiliki keterbatasan yang sah. Ini membuatnya tidak cocok untuk pemerintah langsung, para ahli dan akademisi, perintis, dan hubungan lain untuk menggunakan data yang disediakan dalam panduan ini. Tentu saja, peta dasar *OSM* dapat menyediakan data yang dapat diunduh secara gratis dan langsung, untuk digunakan dan didistribusikan kembali.

Ada banyak tempat di dunia, terutama daerah terpencil dan terpencil secara finansial di mana tidak ada motivasi bisnis bagi organisasi perencanaan untuk melakukan perencanaan di tempat-tempat ini. *OSM* bisa menjadi jawaban untuk tempat-tempat seperti ini, baik dari segi keuangan, tata kota, kemungkinan kegagalan, maupun untuk tujuan yang berbeda (Tanjaya et al., 2016).

2.14 LeafletJS

LeafletJS adalah repositori perpustakaan *JavaScript open source* yang dapat membantu membuat peta di halaman *web*. *Open source* berarti bahwa kode dapat dengan mudah memahami fungsinya, siapa pun dapat menggunakannya, dan siapa pun dapat berkontribusi pada proyek melalui peningkatan kode.

File Java Script juga ditumpuk dengan halaman situs web untuk menyediakan akses ke berbagai kapasitas yang memungkinkan penyediaan layanan peta. Mendukung browser modern di *desktop* dan *platform* seluler, sehingga pengguna dapat mengirimkan peta di mana saja.

Tujuannya adalah kemudahan penggunaan, berpusat pada eksekusi dan kemudahan penggunaan. Selebaran juga dapat memanfaatkan modul yang memperluas kegunaan. *Leaflet* memiliki *API* yang sangat baik, yang banyak dilaporkan, jadi bukan rahasia untuk menggunakannya dalam situasi apa pun kecuali dalam berbagai situasi. Perusahaan yang dikenal sebagai pengguna *Leaflet* termasuk *Flickr*, *foursquare*, *craigslist*, *Data.gov*, *IGN*, *Wikimedia*, *OSM*, *Meetup*, *WSJ*, *Mapbox*, *Cloudmade*, *CartoDB*, dan *GIS Cloud*.

Daripada itu, *Leaflet* menyediakan utilitas untuk menambahkan penanda, jendela *pop-up*, *overlay* garis dan bentuk, menggunakan banyak lapisan, *zoom*, dan cakram, tetapi ini hanya sorotan utama dari *Leaflet*. Salah satu keunggulan signifikan *Leaflet* adalah kemampuannya untuk memperluas ketersediaan konten melalui modul eksternal. Pada saat penulisan, ada lebih dari 80 modul independen yang dapat mengimplementasikan sorotan, seperti melapiskan peta panas, penanda animasi, memuat file *csv* data, menggambar bentuk kompleks, mengukur jarak, memanipulasi lapisan, dan menampilkan koordinat. (Tanjaya et al., 2016).

2.15 UML (Unified Modelling Language)

UML (Unified Modelling Language) merupakan bahasa berbasis grafis/gambar buat membayangkan, menentukan, merakit, dan melaporkan kerangka kemajuan pemrograman berbasis *OO (Object Oriented)*. *UML* sendiri

juga menyediakan spesifikasi untuk menyusun sistem cetak biru, yang menggabungkan konsep skala bisnis, kompilasi kelas dalam dialek pemrograman eksplisit, skema *database*, dan komponen yang diperlukan oleh perangkat lunak sistem (Zufria, 2013).

UML pertama kali diperkenalkan pada tahun 1990, ketika Gady Booch dan Ivar Jacobson dan James Rumbaugh mulai merangkul ide dan kemampuan tambahan dari teknologi masing-masing dan mencoba membuat strategi terikat, yang kemudian diberi nama *UML*. Pada bulan Oktober 1994, ketika Rumbaugh bergabung dengan Booch di *Rational Software Corporation*, upaya untuk memajukan *UML* dimulai. Proyek pertama mereka adalah menggabungkan metode Booch dengan *OMT (Object Modeling Technology)*. *UML* versi 0.8 dirilis pada Oktober 1995. Saat mengirimkan formulir *UML* 0.8, Jacobson juga menambahkan Reasonable untuk memperluas *UML* agar menyertakan *OOSE*. Hasilnya adalah adaptasi dari *UML* versi 0.9 yang dikirimkan pada bulan Juni 1996. Sepanjang tahun 1996, Rumbaugh, Booch, dan Jacobson juga menerima banyak kontribusi dari komunitas rekayasa perangkat lunak. Saat ini, banyak asosiasi perangkat lunak melihat *UML* sebagai sistem untuk bisnis mereka.

Kemudian, pada saat itu Rumbaugh, Booch, dan Jacobson memperkenalkan aliansi *UML* yang disebut *OMG (Object Management Group)*, di mana individu-individu tersebut adalah beberapa asosiasi yang sangat loyal yang tujuannya adalah untuk melengkapi atau memperluas ide-ide dasar *UML*. Upaya bersama ini telah menghasilkan *UML* 1.0 di hampir semua jenis pemrograman dan rencana kerangka data. *UML* sendiri juga menyediakan norma untuk membangun kerangka kerja rencana yang menggabungkan gagasan ukuran bisnis, membangun kelas dalam bahasa pemrograman eksplisit, cetak biru set data, dan segmen yang diperlukan dalam kerangka kerja pemrograman.

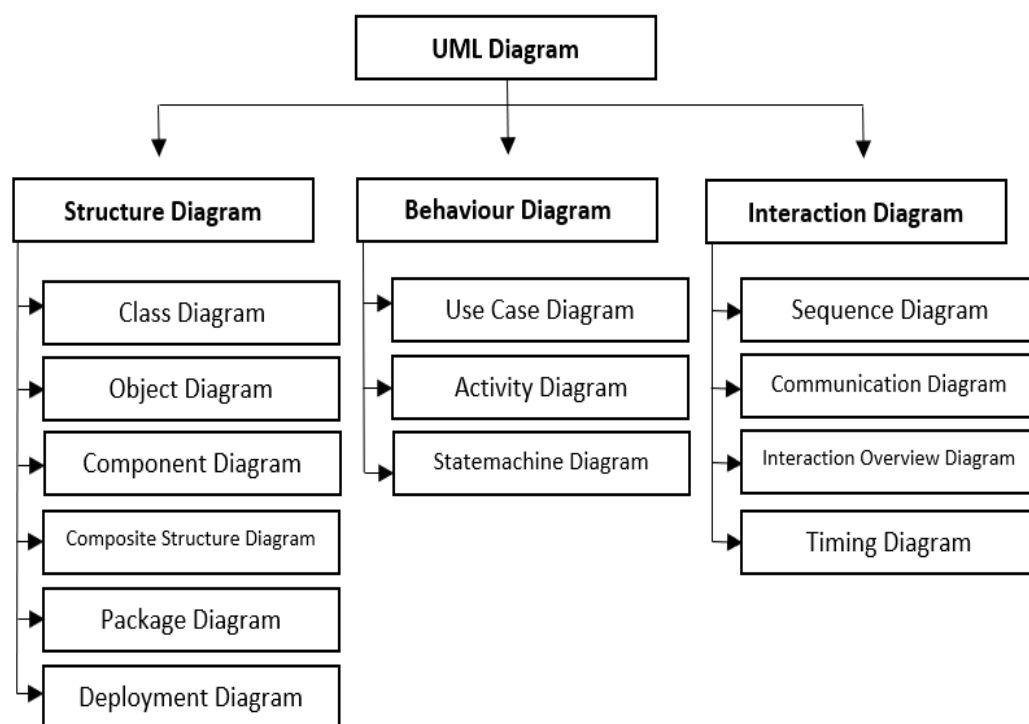
Tujuan dari *UML (Unified Modelling Language)* diantaranya adalah (Zufria, 2013):

1. Menyediakan bahasa tampilan visual siap pakai yang berterima kasih karena

telah membuat dan memperdagangkan model dengan cara yang sederhana dan secara umum dapat dibenarkan.

2. Menyediakan bahasa tampilan yang bebas dari berbagai dialek pemrograman dan siklus desain.
3. Sertakan prosedur yang ditentukan dalam demonstrasi.

Rosa dan Salahuddin menyatakan bahwa *UML (Unified Modeling Language)* adalah salah satu prinsip bahasa yang banyak digunakan di dunia modern untuk mendeskripsikan kebutuhan, menganalisis, mendesain, dan mendeskripsikan arsitektur dalam pemrograman berorientasi objek (PBO). Berikut ini adalah jenis-jenis diagram *UML (Unified Modelling Language)*:



Gambar 2.6 Jenis-jenis diagram *UML*

Mengingat penggambaran di atas, analis membuat aliran kerangka kerja yang ditampilkan sebagai *Usecase* diagram, *Activity* diagram, *Sequence* diagram, dan *Class* diagram dalam bentuk model *UML (Unified Modelling Language)*.

2.15.1 Bangunan Dasar *UML*

Untuk memahami *UML*, kita membutuhkan model konseptual dari "bahasa" ini. Metode *UML* menggunakan 3 struktur dasar untuk menggambarkan sistem/perangkat lunak yang sedang dikembangkan, yaitu:

1. Benda (*Things*)
2. Relasi (*Relationship*)
3. Diagram

Setiap konstruksi dasar dapat diterapkan ke seluruh fase pengembangan sistem. Ketiga jenis bangunan ini dapat saling melengkapi. Secara umum, *UML* digunakan dalam pengembangan sistem/perangkat lunak berorientasi objek, karena metode *UML* ini biasanya memiliki kelebihan sebagai berikut (Zufria, 2013):

1. *Uniformity*

Dengan metode *UML*, pengembang hanya perlu menggunakan satu metode, mulai dari tahap analisis hingga desain. Hal ini tidak dapat dilakukan dalam pendekatan pengembangan terstruktur. Dengan perkembangan aplikasi *GUI* (*graphical user interface*) saat ini, *UML* juga memungkinkan kita untuk merancang komponen antarmuka pengguna dan desain perangkat lunak dan desain *database* secara terintegrasi.

2. *Understandibility*

Dengan menggunakan metode ini, kode yang dihasilkan dapat diatur ke dalam kelas-kelas yang terkait dengan masalah yang sebenarnya, sehingga lebih mudah dipahami oleh siapa saja.

3. *Stability*

Kode program yang dihasilkan relatif stabil dari waktu ke waktu karena sangat dekat dengan masalah sebenarnya di situs.



4. *Reusability*


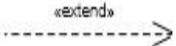

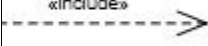
Menggunakan pendekatan berorientasi objek, kode dapat digunakan kembali, yang sangat mempercepat waktu pengembangan perangkat lunak.

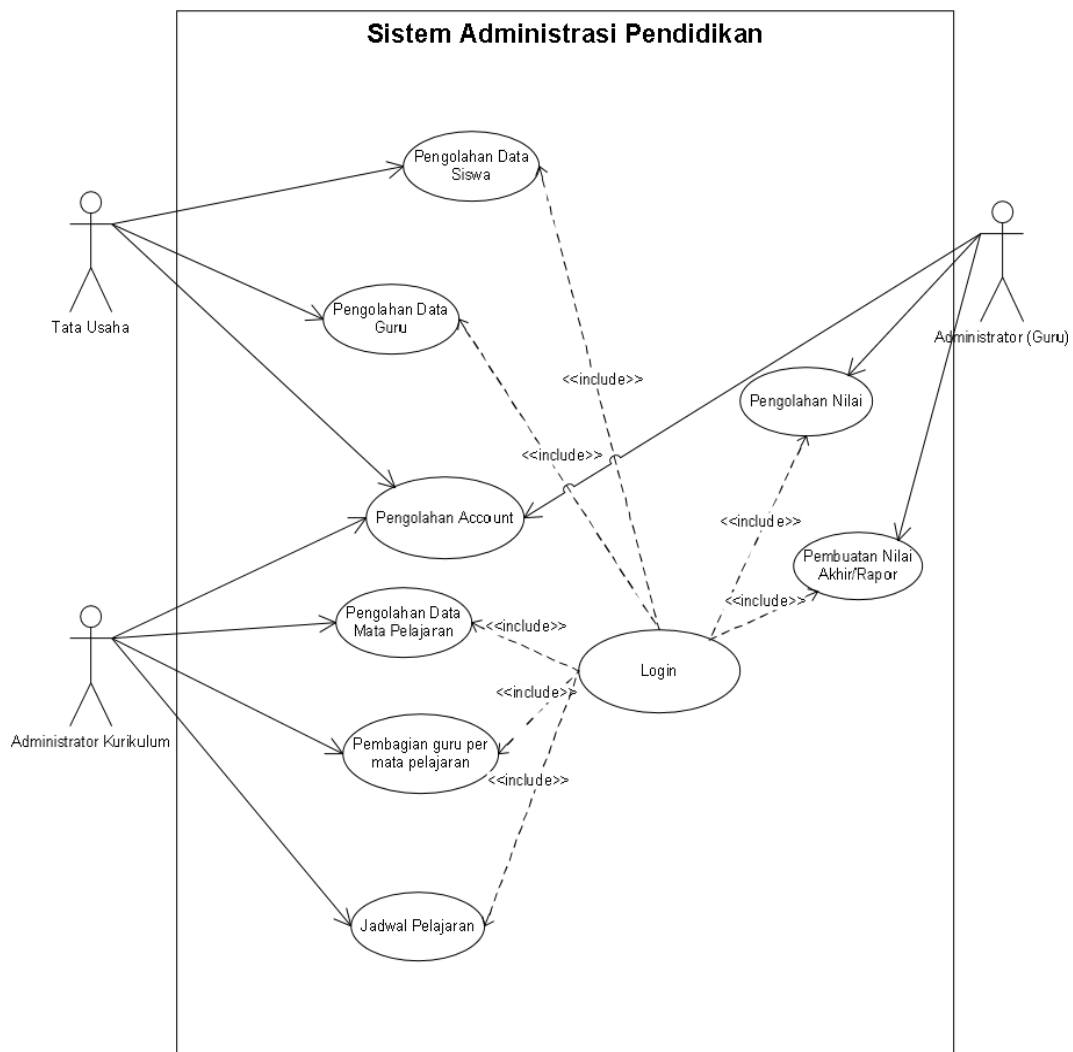
2.15.2 Use Case Diagram

Use case menggambarkan tampilan luar dari sistem yang akan kita modelkan. Model *use case* dapat dideskripsikan dalam *use case* diagram, tetapi perlu diingat bahwa diagram berbeda dari model karena model lebih luas dari diagram. *Use case* harus mampu menggambarkan urutan aktor yang menghasilkan nilai yang terukur (Suendri, 2019). Simbol yang digunakan dalam *Use case* diagram yaitu:

Tabel 2.3 Simbol *Use Case* Diagram

No	Simbol	Nama	Keterangan
1		<i>ACTOR</i>	Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
2		<i>USE CASE</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.

3		<i>ASSOCIATION</i>	Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.
4		<i>EXTEND</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan memiliki nama depan yang sama dengan use case yang di tambahkan.
5		<i>GENERALIZATION</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6		<i>INCLUDE</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsional atau sebagai syarat dijalankan use case ini.




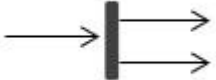

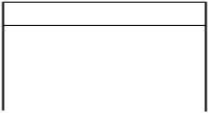


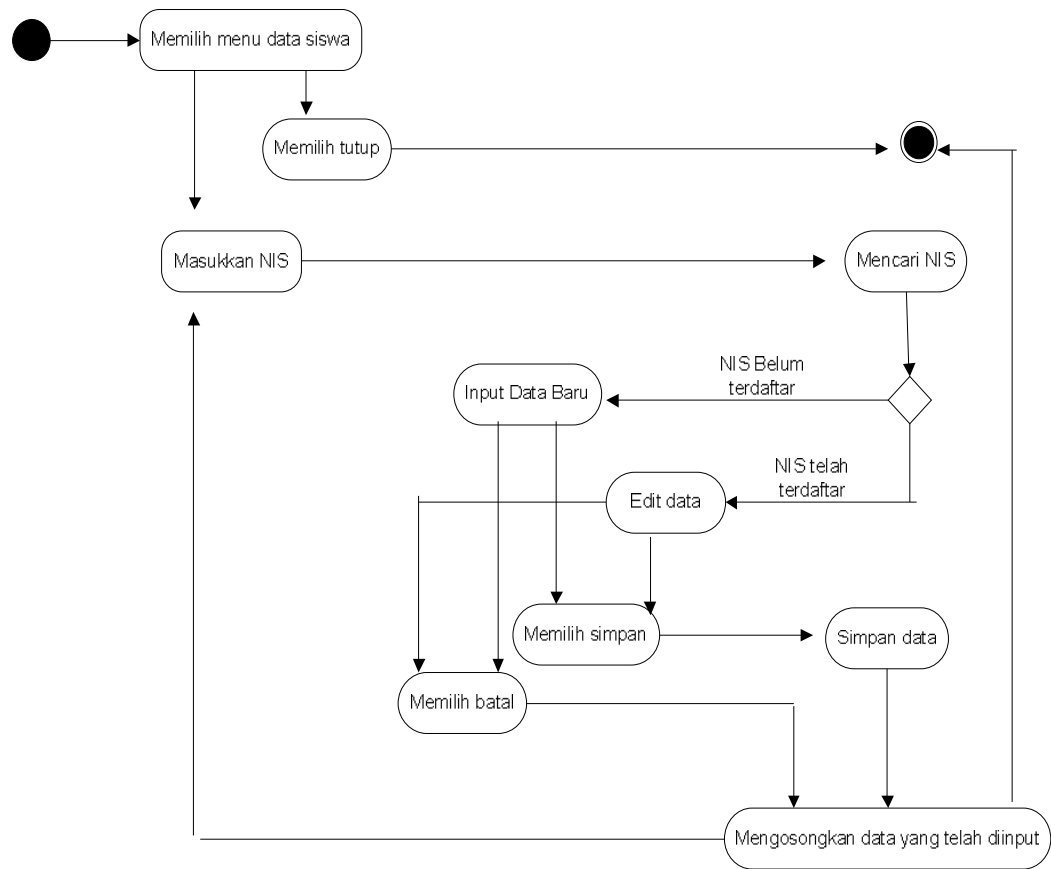
Gambar 2.7 Contoh Desain *Use Case* Diagram

2.15.3 Activity Diagram

Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk serangkaian tindakan, bagaimana setiap tindakan dimulai, dan keputusan yang mungkin terjadi sebelum tindakan berakhir. Diagram aktivitas juga dapat menggambarkan proses beberapa tindakan pada saat yang bersamaan. Diagram aktivitas adalah aktivitas, objek, status, transisi status, dan peristiwa. Dengan kata lain, diagram alur kerja aktivitas menggambarkan perilaku sistem dari aktivitas (Suendri, 2019). Berikut simbol *activity* diagram, yaitu:

Tabel 2.4 Simbol *Activity* Diagram

No	Simbol	Nama	Keterangan
1		<i>INITIAL</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2		<i>ACTIVITY</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>DECISION</i>	Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
4		<i>JOIN NODE</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
5		<i>FINAL</i>	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.
6		<i>SWIMLINE</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.



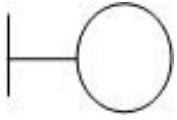
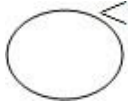

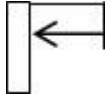


Gambar 2.8 Contoh Desain *Activity Diagram*

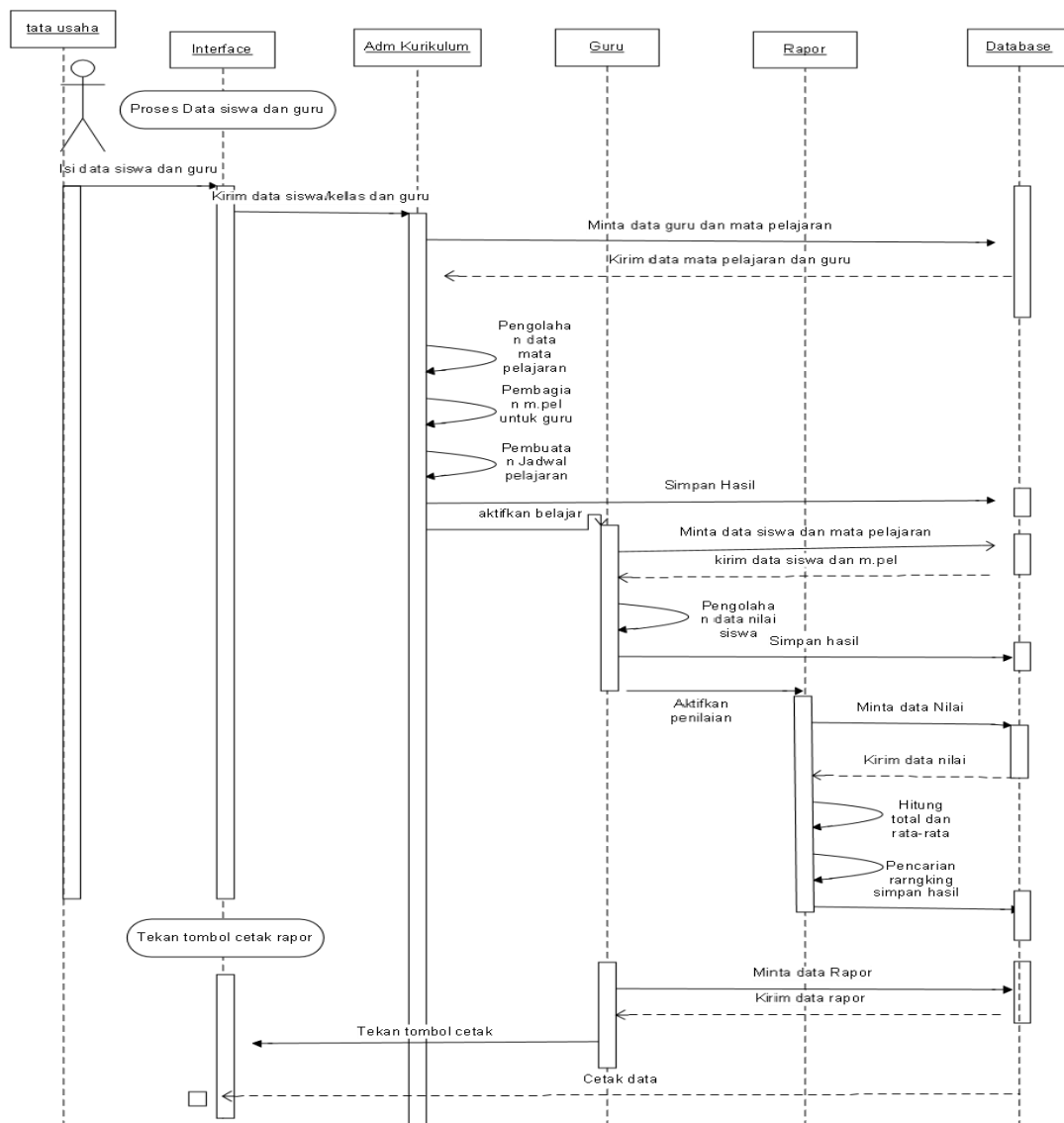
2.15.4 *Sequence Diagram*

Sederhananya, *sequence diagram* adalah deskripsi langkah demi langkah, termasuk urutan waktu (*sequence*) perubahan logis yang harus dilakukan untuk menghasilkan sesuatu berdasarkan *use case diagram* (Suendri, 2019). Berikut simbol *sequence diagram*, yaitu:

Tabel 2.5 Simbol *Sequence Diagram*

No.	GAMBAR	NAMA	DESKRIPSI
1		<i>Entity Class</i>	<i>Entity Class</i> , Ini adalah bagian dari sistem dan berisi serangkaian.

2		<i>Boundary Class</i>	<i>Boundary Class</i> , Berisi sekumpulan kelas, kelas-kelas ini menjadi antarmuka atau interaksi antara satu atau lebih peserta dengan sistem, seperti format tampilan dan bentuk cetak.
3		<i>Control Class</i>	<i>Control class</i> , Objek yang berisi logika aplikasi yang tidak bertanggung jawab atas entitas, seperti perhitungan dan aturan bisnis yang melibatkan banyak objek.
4		<i>Message</i>	<i>Message</i> , Simbol untuk mengirim pesan antar kelas.
5		<i>Recursive</i>	<i>Recursive</i> , Menjelaskan pengiriman pesan yang dikirim ke dirinya sendiri.
6		<i>Activation</i>	<i>Activation</i> , Mewakili eksekusi operasi objek, panjang kotak ini sebanding dengan durasi aktivitas operasi.
7		<i>Lifeline</i>	<i>Lifeline</i> , garis putus-putus terhubung ke objek, dan ada aktivasi di sepanjang garis hidup.



Gambar 2.9 Contoh Desain *Sequence Diagram*

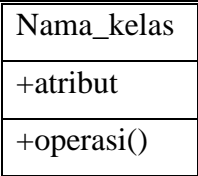
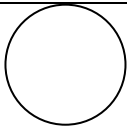


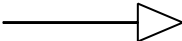
2.15.5 Class Diagram


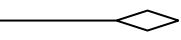
Class Diagram menjelaskan jenis objek dalam sistem dan berbagai hubungan statis yang ada di antara mereka. Diagram kelas juga menunjukkan atribut dan operasi kelas dan batasan yang terkandung dalam hubungan objek-objek ini (Samsudin et al., 2019). *Class diagram* Ini mencakup tiga bidang utama, yaitu: (Suendri, 2019):

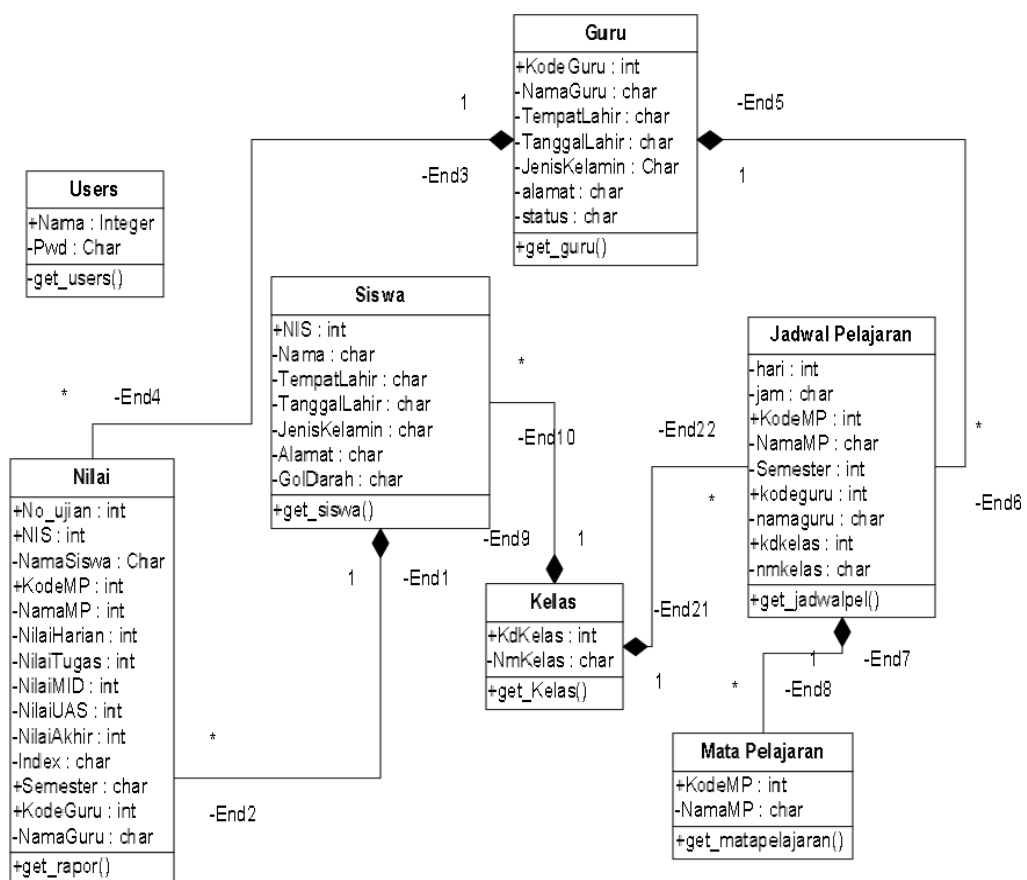
1. Nama, kelas harus mempunyai sebuah nama.

2. Atribut, adalah perangkat yang terpasang ke kelas. Nilai suatu kelas hanya dapat diproses dalam rentang atributnya.
3. Operasi, adalah sebuah proses yang dapat dieksekusi oleh kelas, dapat dieksekusi di kelas itu sendiri, juga dapat dieksekusi di kelas lain.

Tabel 2.6 Simbol *Class* Diagram

No.	GAMBAR	NAMA	DESKRIPSI
1		Kelas	Kelas pada struktur sistem.
2		Antar-muka/ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3		Asosiasi/ <i>association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4		Asosiasi berarah/ <i>Directed association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).

6		Kebergantungan/ <i>dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.
7		Agregasi/ <i>aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).



Gambar 2.10 Contoh Desain *Class Diagram*

2.16 PHP (*Hypertext Preprocessor*)

PHP berasal dari istilah "*hypertext preprocessor*", Ini adalah bahasa pemrograman tujuan umum untuk menangani pembuatan dan pengembangan situs *web*, dan dapat digunakan bersama dengan *HTML*. Saat ini *PHP* banyak digunakan

untuk membuat program *website* dinamis. Contoh aplikasi *PHP* adalah forum (*phpBB*) dan media *wiki* (perangkat lunak di belakang *Wikipedia*). *Mambo*, *Joomla*, *Postnuke*, *Xaraya*, dan aplikasi lainnya adalah contoh aplikasi yang lebih kompleks yang berbentuk *CMS* dan dibangun menggunakan *PHP*. Selain itu, *PHP* juga menjadi pilihan lain untuk *ASP.NET/C#/VB.NET* *Microsoft*, *ColdFusion* *Macromedia*, *JSP/Java* *Sun Microsystems* dan *CGI/Perl* untuk membuat atau menjalankan program.

Fungsi utama *PHP* sebagai kumpulan script atau bahasa pemrograman adalah untuk dapat mengumpulkan dan evaluasi hasil survei atau segala bentuk server basis data, dan menghasilkan efek tindak lanjut pada tahap selanjutnya. Efek *stripe PHP* ini merupakan bentuk Ini akan berkomunikasi dengan *database*, mengumpulkan dan mengklasifikasikan informasi, dan kemudian menampilkan operasi skrip lainnya ketika pengunjung situs membutuhkannya (menampilkan informasi berdasarkan permintaan pengguna) (Hidayatullah, 2020).

2.17 MySQL

MySQL adalah sistem manajemen *database SQL open source*, saat ini yang paling banyak digunakan. Sistem *database MySQL* dapat mendukung beberapa fungsi seperti *multi-threading*, *multi-user*, dan sistem manajemen *database SQL (DBMS)*. Jika kita membutuhkan sistem *database* yang cepat, handal dan mudah digunakan, kita akan segera menggunakan *MySQL*. Selain itu, *MySQL* juga dapat dikatakan sebagai implementasi dari sistem manajemen basis data relasional terdistribusi bebas (*RDBMS*) (Hidayatullah, 2020). Adapun kelebihan *MySQL*, antara lain:

1. Portabilitas

Dengan keunggulan tersebut, *MySQL* dapat berjalan dengan stabil di berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Server Mac Os X*, *Solaris*, dan *Amiga*.

2. Open Source

Keunggulan ini sering disebut sebagai perangkat lunak *open source*, dan

menyebabkan *MySQL* didistribusikan dan digunakan secara gratis.

3. *Multi-User*

Keunggulan ini memungkinkan *MySQL* untuk digunakan oleh banyak pengguna secara bersamaan tanpa masalah atau konflik.

4. *Performance Tunning*

Hal ini terkait dengan kecepatan akses, dimana *MySQL* memiliki kecepatan yang luar biasa dalam memproses *query* sederhana dan dapat memproses lebih banyak *SQL* per satuan waktu.

5. Ragam Tipe Data

MySQL memiliki beragam tipe data, seperti integer yang ditandatangani/tidak ditandatangani, angka *floating-point*, angka presisi ganda, karakter, teks, tanggal, cap waktu, dll.

6. Perintah dan fungsi

MySQL memiliki seperangkat operator dan fungsi yang lengkap, dan mendukung perintah pilih dan di mana dalam kueri.

7. Keamanan

Lapisan keamanan *MySQL* menggunakan kata sandi terenkripsi.

8. Skalabilitas dan Pembatasan

Keuntungan ini memungkinkan *MySQL* untuk menangani database besar dengan lebih dari 50 juta catatan dan 60.000 tabel dan 5 miliar baris, dan setiap tabel dapat menampung hingga batas indeks 32 indeks.

9. Koneksi

Protokol *TCP/IP*, soket *Unix (UNIX)* atau pipa bernama (*NT*) adalah koneksi yang dapat dibuat *MySQL*.

10. Lokalisasi

Keunggulan ini memungkinkan *MySQL* untuk mendeteksi pesan kesalahan pada klien dalam lebih dari 20 bahasa.

11. Antarmuka

Maksud dari keunggulan ini tidak berarti bahwa setiap programmer bertatap muka saat menjalankan *MySQL*, tetapi keunggulan ini dapat menunjukkan bahwa *MySQL* dapat terhubung ke berbagai aplikasi dan bahasa pemrograman

melalui penggunaan fungsi *API (application programming interface)*.

12. Klien dan Peralatan

MySQL dilengkapi dengan berbagai alat yang dapat digunakan untuk manajemen basis data, dan setiap alat yang tersedia mencakup instruksi *online*.

13. Struktur Tabel

Struktur tabel *MySQL* lebih fleksibel ketika berhadapan dengan *ALTER TABLE* dibandingkan *database* lain seperti *PostgreSQL* atau *Oracle*.

2.18 HTML (*HyperText Markup Language*)

HTML adalah singkatan dari *Hypertext Markup Language*. *HTML* adalah bahasa *markup* yang tujuannya untuk menandai isi suatu dokumen agar tidak dipahami sebagai teks biasa, dapat dihubungkan satu sama lain, dan dipahami oleh perangkat lunak pembaca yang dapat dipahami, yang disebut *web browser*. *HTML* dimodelkan menggunakan *SGML (Standard Generalized Markup Language)*, yang merupakan standar deklaratif untuk mendeskripsikan struktur dan atribut dokumen. Faktanya, *HTML* adalah kumpulan kode yang dapat digunakan untuk menentukan teks dalam dokumen sebagai elemen dokumen dan dapat ditautkan satu sama lain.

Adapun perbedaan dokumen *web* dengan dokumen *html*. Dokumen *web* adalah dokumen yang berisi informasi yang ditampilkan di setiap situs *web*, baik di internet ataupun intranet, pada umumnya dengan menggunakan browser *web*. Dokumen *web* dikenal juga dengan dokumen *HTML* adalah *file* teks biasa, yang elemen teks di dalamnya memiliki atau diberi kode elemen, yang disebut dengan *tag*. Dengan adanya kode elemen atau pada setiap atau kelompok teks, maka browser bisa menampilkan dokumen sesuai dengan definisi dari kodenya (Sidik, 2019).

Sejauh ini, kami telah mengembangkan *HTML 5*, yang merupakan versi terbaru, yang tidak hanya mendukung gambar dan teks, tetapi juga mendukung menu interaktif, audio, video, dll. Beberapa tugas utama *HTML* saat membangun sebuah *website* antara lain:

1. Tentukan tata letak situs *web*.
2. Pemformatan teks dasar, seperti pengaturan paragraf dan pemformatan *font*.
3. Buat daftar.
4. Buat tabel.
5. Sisipkan gambar, video, dan audio.
6. Membuat link.
7. Membuat formulir.

2.19 CSS (*Cascade Style Sheet*)

CSS (Cascade Style Sheet) adalah suatu bahasa yang digunakan untuk menjelaskan *style* dari suatu dokumen dan elemen *HTML*. *CSS* digunakan untuk menjelaskan bagaimana elemen-elemen dalam suatu dokumen *HTML* harus ditampilkan dengan menggunakan definisi yang dibuat secara terpisah dari kode *HTML* nya. Dengan adanya *CSS*, maka pengembang *web* tidak lagi menggunakan atribut dari suatu elemen, karena digantikan dengan menggunakan *CSS*. Penggunaan atribut, walaupun dengan atribut *style* diharapkan bisa dihindari.

Dengan *CSS* kita bisa memisahkan komponen atau elemen yang digunakan untuk bagaimana menampilkan informasi dengan informasinya itu sendiri. Sebelum ada *CSS*, maka isi dari dokumen *HTML* tercampur antara elemen dokumen dengan komponen untuk bagaimana menampilkannya, terlalu banyak atribut yang harus dituliskan dalam *tag* elemen.

Style atau gaya disini berhubungan dengan atribut elemen dokumen *HTML*. Memberikan *style* atau memberikan gaya kepada elemen-elemen dokumen agar dokumen ditampilkan secara menarik di dalam browser *web*. Cara memberikan *style* ke dalam elemen dokumen:

1. Inline
2. Internal
3. Eksternal

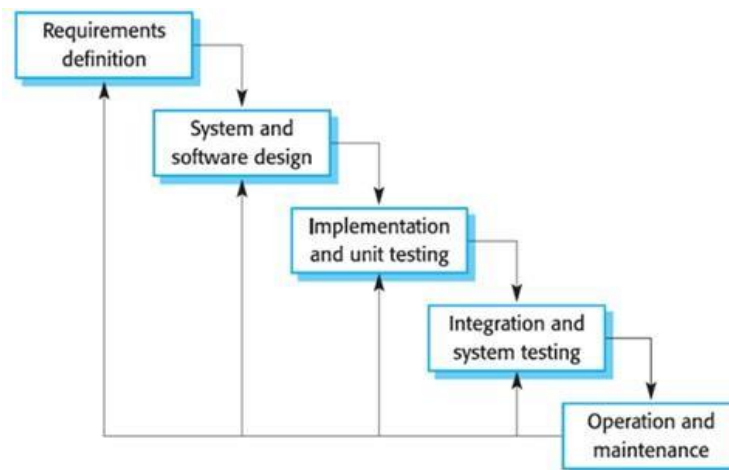
Ketiga cara pemberian *style* di atas dapat dilakukan secara kombinasi, tidak harus memilih salah satu cara dalam satu dokumen *web* (Sidik, 2019).

2.20 Framework Laravel

Kerangka kerja adalah struktur konseptual dasar yang digunakan untuk memecahkan atau menangani masalah yang kompleks. *Frame* merupakan bingkai atau *frame* dari *website* yang akan dibangun. *Laravel* adalah *framework open source* berbasis *PHP*, yang menggunakan konsep *model-view-controller* (Yudhanto & Prasetyo, 2019).

2.21 Waterfall

Metode penelitian yang digunakan dalam penelitian ini adalah pengembangan dari metode *waterfall*. Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematis dan berurutan. Metode *waterfall* memiliki tahapan sebagai berikut (Sasmito, 2017) :



Gambar 2.11 Metode *Waterfall*

1. *Requirements Definition*

Layanan sistem, kendala, dan tujuan ditentukan oleh hasil negosiasi dengan pengguna, yang kemudian didefinisikan secara rinci dan digunakan sebagai spesifikasi sistem.

2. *System and Software Design*

Tahap perancangan sistem mengalokasikan perangkat keras dan perangkat lunak yang dibutuhkan oleh sistem dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. *Implementation and Unit Testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. *Integration and System Testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*.

5. *Operation and Maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

2.22 Studi Sejenis

Dalam Penelitian ini sedikit banyak pencerahan dan referensi diperoleh dari penelitian-penelitian terdahulu yang bersangkutan dengan latar belakang permasalahan penelitian ini. Berikut penelitian terdahulu antara lain:

Tabel 2.7 Studi Sejenis

Peneliti	Judul	Persamaan	Perbedaan	Masalah Penelitian
Aulia	Sistem	Memberika	Tidak memberikan	Di kota Medan

Pratama Tambunan (2019)	Informasi Geografis Pemetaan Lokasi Kuliner Halal Di Medan	n kemudahan untuk mempromosikan wisata kuliner di Medan dengan menggunakan SIG	sistem informasi geografis berupa tempat penginapan, wisata Medan, dan informasi <i>event</i> Medan serta tidak menggunakan metode algoritma <i>Dijkstra</i> untuk mencari <i>route</i> terpendek	ini tidak semua wisata kuliner menyediakan daftar kuliner halal, maka dari itu dibangunlah SIG yang dapat mengelola data kuliner halal untuk para pengunjung
Helmi Kurniawan, Muhammad Rusdi Tanjung (Vol.7, No.1, Januari 2017)	Sistem Informasi Geografis Objek Wisata Alam di Provinsi Sumatera Utara Berbasis Mobile Android	Memudahkan wisatawan dalam mencari lokasi tempat wisata dan informasi menarik	SIG yang dibangun berbasis android versi 2.3 ke atas menggunakan <i>Eclipse</i> , dan tidak menggunakan fitur yang lain seperti halnya objek wisata lain selain alam, tempat penginapan, wisata kuliner/oleh-oleh khas Medan serta <i>event</i> yang di adakan di kota Medan maupun sesumatera utara	Di Sumatera utara para wisatawan belum dapat mengetahui informasi yang akurat serta belum menemukan lokasi wisata alam yang meliputi gambar dan data yang nyata dan benar
Antonio Gusmao,	Sistem Informasi	Menggunakan metode	Tidak memberikan sistem informasi	Pada objek wisata di timor

Sholeh Hadi Pramono, Sunaryo (Vol.7, No.2, Desember 2013)	Geografis Pariwisata Berbasis <i>Web</i> dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra	algoritma <i>Dijkstra</i> dan memberikan kemudahan untuk mencari objek wisata dengan rute terdekat	geografis berupa tempat penginapan, wisata kuliner, dan informasi <i>event</i> pada sistemnya	leste para pengunjung belum dapat informasi untuk mengetahui <i>rute</i> terdekat dengan titik awal berada ke titik tujuan
---	--	--	---	--

BAB III

METODE PENELITIAN

3.1 Tempat Penelitian

Dalam penulisan skripsi ini peneliti melakukan tempat penelitian pada Dinas Pariwisata Kota Medan yang beralamat di Jln. Prof. HM. Yamin Sh No.40, Gg.Buntu, Kec. Medan Timur, Kota Medan, Sumatera Utara 20232.

3.2 Waktu Penelitian

Peneliti membutuhkan waktu penelitian yang dilaksanakan dari bulan Maret s/d Juni 2021. Untuk penjelasan lebih detail pada waktu penelitian bisa dilihat pada Tabel 3.1, sebagai berikut:

Tabel 3.1 Jadwal Penelitian

Jadwal Penelitian	Maret				April				Mei				Juni				Juli				Agustus			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																								
Pengajuan dan Pengerjaan Proposal Skripsi																								
Seminar Proposal Skripsi																								
Pengumpulan Data dan Analisis Data Perencanaan Sistem																								
Perancangan Sistem																								
Pengcodingan Sistem																								
Pengujian Sistem																								

Tahapan penelitian yang akan dilakukan adalah sebagai berikut:

1. Identifikasi Masalah

Peneliti melakukan riset permasalahan yang ada pada Dinas Pariwisata Kota Medan untuk mencari solusi dari permasalahan tersebut.

2. Pengajuan Judul Proposal Skripsi

Peneliti melakukan pengajuan judul dari masalah yang diangkat, sebagai syarat

untuk mengajukan judul proposal skripsi dan mempersiapkan proposal skripsi untuk seminar proposal.

3. Penjadwalan

Saat melakukan penelitian, peneliti perlu mengatur jadwal untuk membantu menentukan batas waktu dalam pembuatan sistem.

4. Seminar Proposal

Seminar proposal diadakan untuk memahami penerapan penelitian yang diusulkan dengan mempresentasikan judul yang terkait. Seminar proposal ini akan menguji kelayakan penyiapan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan materi pendukung lainnya yang sudah disiapkan oleh peneliti serta sudah melakukan proses bimbingan dari dosen pembimbing I dan II.

5. Pengumpulan Data

Setelah selesai seminar proposal, dilakukan pengumpulan data di lokasi penelitian sebagai objek penelitian, wawancara, dan studi pustaka terkait penelitian.

6. Analisis Sistem

Setelah mengumpulkan data, peneliti menganalisa data yang diperoleh agar dapat membuat rekomendasi sistem usulan yang lebih baik dari sistem yang dijalankan sebelumnya.

7. Perancangan Sistem

Pada tahap ini peneliti mulai merancang sistem melalui diagram model *UML*, perancangan basis data dan perancangan antarmuka (*interface*).

8. Pengcodingan Sistem

Tahapan ini merupakan tahapan dimana peneliti melakukan pembuatan kode program sistem untuk dijadikan sebuah *website*.

9. Pengujian Sistem

Setelah pengkodean sistem selesai, peneliti akan menguji sistem agar dapat melihat apakah sistem tersebut beroperasi sesuai dengan yang diharapkan.

3.3 Kebutuhan Sistem

Adapun kebutuhan spesifikasi sistem yang diperlukan untuk menyelesaikan penelitian skripsi, kebutuhan sistem dibagi menjadi perangkat keras dan perangkat lunak.

3.3.1 Perangkat Keras

Adapun kebutuhan perangkat keras untuk penyelesaian pembuatan sistem tersebut. Spesifikasi perangkat keras yang digunakan dalam perancangan sistem yaitu:

1. Leptop atau *PC*
2. *Processor Intel(R)* atau *AMD*
3. *RAM 4GB*
4. *HDD 500GB* atau *SSD 128GB*

3.3.2 Perangkat Lunak

Adapun kebutuhan perangkat lunak untuk penyelesaian pembuatan sistem tersebut. Spesifikasi perangkat lunak yang digunakan dalam perancangan sistem yaitu:

1. *Operating System Windows 7*
2. *Xampp 8.0.7 64 bit*
3. *Visual Studio Code 1.58.0*
4. *Star Uml 3.0.2*
5. *Mendeley Desktop 1.19.8*
6. *Browser (Google Chrome)*

3.4 Cara Kerja

Dilihat dari jenis datanya, metode penelitian dalam metode penelitian ini adalah metode kualitatif, yaitu memahami fenomena yang dialami oleh objek penelitian secara keseluruhan dan mendeskripsikannya melalui teks dan bahasa khusus. Latar belakang alam dan metode pengumpulan data berupa observasi,

wawancara dan penelitian kepustakaan, serta pengembangan sistematis metode metode *waterfall* yang digunakan peneliti.

3.4.1 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan dalam penelitian ini yaitu melalui observasi, wawancara, dan studi pustaka (seperti jurnal dan buku-buku yang berkaitan dengan penelitian).

1. Observasi

Peneliti melakukan observasi pada Dinas Pariwisata Kota Medan. Pengamatan atau observasi dilakukan secara sistematis dengan cara mengamati langsung lokasi atau objek penelitian.

2. Wawancara

Wawancara merupakan tahapan pengumpulan data melalui tanya jawab antara peneliti dengan narasumber yang mengetahui masalah tersebut. Peneliti melakukan wawancara terhadap pegawai Dinas Pariwisata Kota Medan, dan Kasubbag Keuangan dan Penyusunan Program.

3. Studi Pustaka

Studi pustaka dilakukan dengan mempelajari banyak penelitian sebelumnya, baik dalam format jurnal, skripsi, serta kajian terhadap buku-buku yang berkaitan dengan permasalahan penelitian seperti buku Sistem Informasi Geografis Berbasis *Web* Menggunakan *Google Maps* dan *Mapbox API* oleh Eko Budi Setiawan (2020) dan buku Pengembangan Aplikasi Sistem Informasi Geografis Berbasis *Dekstop* dan *Web* oleh Riyanto dkk (2019).

3.4.2 Jenis Data

Data yang diperoleh dan dikumpulkan oleh peneliti dibagi menjadi dua jenis data yaitu:

1. Data Primer

Data primer merupakan data yang diperoleh dari instansi ataupun data yang dikumpulkan melalui perorangan di lokasi penelitian dengan cara melakukan wawancara, observasi ataupun pengamatan secara langsung. Dalam hal ini Dinas

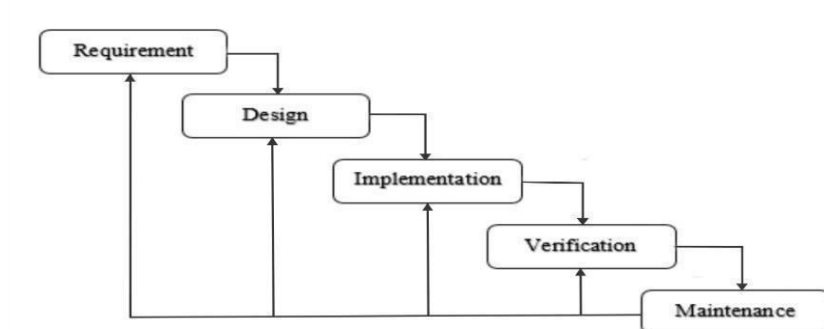
Pariwisata Kota Medan merupakan tempat yang menggunakan algoritma *Dijkstra* untuk penelitian dan pembuatan Sistem Informasi Geografis Pariwisata Kota Medan melalui observasi dan pendataan.

2. Data Sekunder

Data sekunder adalah data yang diperoleh dengan mengumpulkan data-data yang ada, mempelajari teori-teori yang berkaitan dengan pengembangan Sistem Informasi Geografis Pariwisata Medan, dan menggunakan algoritma *Dijkstra* sebagai dasar pembahasan permasalahan yang ada.

3.4.3 Metode Pengembangan Sistem

Dalam sistem ini, metode pengembangan sistem dengan menggunakan *waterfall* yang merupakan metode sistematis atau metode tindak lanjut dari pembentukan sistem.



Gambar 3.1 Metode *Waterfall*

1. *Requirement*

Pada tahap ini, pengumpulan data dapat diperoleh dengan observasi, wawancara, studi pustaka. Hal ini untuk mendapatkan data-data yang diperlukan.

2. *System Design*

Pada tahap ini, dilakukan hubungan antar tabel, desain sistem serta persyaratan dan spesifikasi perangkat keras dan perangkat lunak yang digunakan.

3. *Implementation*

Pada tahap ini, sistem akan dikembangkan sesuai desain yang telah ditentukan dan menggunakan kode program *PHP*, *HTML*, *MySQL* sebagai *database*.

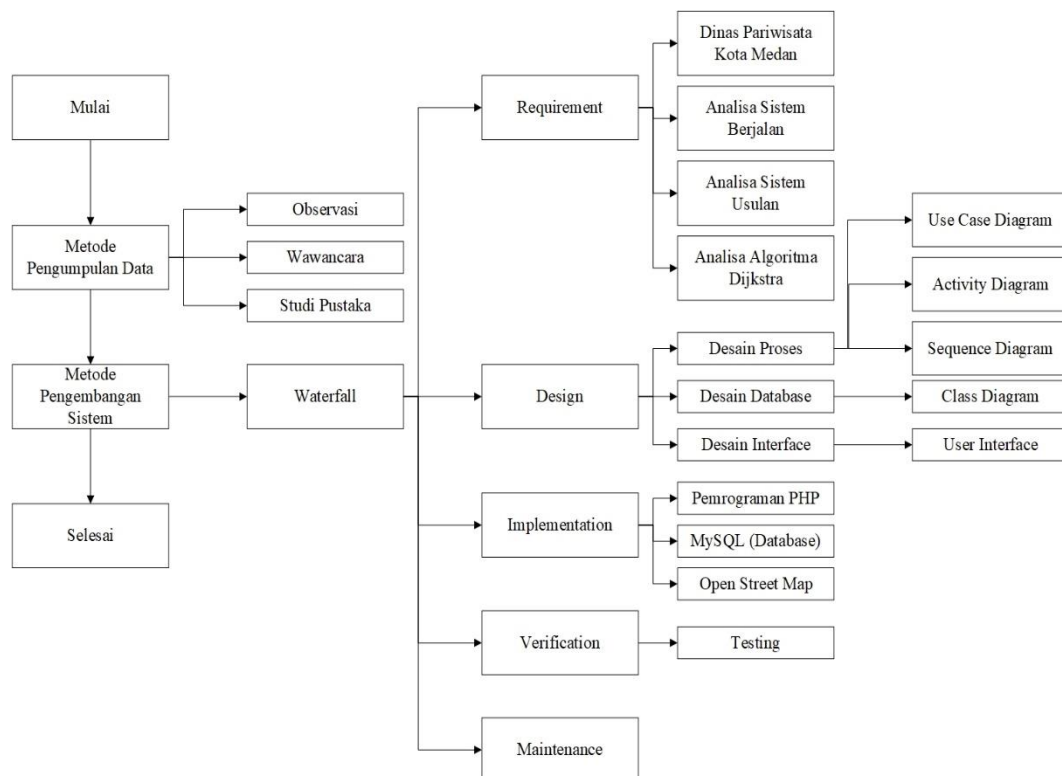
4. *Verification*

Pada tahap ini, dilakukan revisi jika pada sistem adanya malfungsi atau kesalahan pada sistem.

5. *Maintenance*

Pada tahap terakhir adalah *maintenance*. Tahap ini sistem sudah sesuai dengan kebutuhan yang diperlukan dan dapat menyelesaikan masalah. Perangkat lunak dapat dijalankan dan dipelihara. Meningkatkan implementasi sistem dan meningkatkan layanan adalah kebutuhan baru.

3.5 Kerangka Berpikir



Gambar 3.2 Kerangka Berpikir

BAB IV

HASIL DAN PEMBAHASAN

4.1 *Requirement Planning*

Requirement Planning adalah tahap rencana yang memiliki beberapa tahap yang salah satunya terkait tentang Dinas Pariwisata Kota Medan, Analisis sistem saat ini, analisis desain sistem baru/usulan dan analisis perhitungan algoritma *Dijkstra*.

4.1.1 Profil Dinas Pariwisata Kota Medan

Sejak berdirinya Dinas Pariwisata Kota Medan pada tahun 1991, dan penambahan urusan kebudayaan pada tahun 2001, hal ini diatur dalam Peraturan Daerah Kota Medan Nomor 20 Tahun 2002 tentang Maksud dan Fungsi Dinas dan Kebudayaan Kota Medan. Revisi terakhir adalah Perda Kota Medan Nomor 3 Tahun 2009, yang merupakan implementasi dari Peraturan Pemerintah RI Nomor 38 Tahun 2007 tentang Organisasi dan Tata Kerja Pembentukan Instansi Daerah Perkotaan.

Peraturan Pemerintah Republik Indonesia Nomor 41 Tahun 2007 tentang Organisasi Perangkat Daerah yang merupakan unsur penyelenggara pemerintahan daerah, dipimpin oleh penanggung jawab pelayanan di bawah dan bertanggung jawab kepada Walikota Medan, melalui perangkat daerah atau sekretaris.

Kemudian pada tahun 2017 melalui peraturan Walikota Medan (Perwal) Nomor 1 Tahun 2017 tentang Kedudukan, Susunan Organisasi, Tanggung Jawab dan Fungsi serta Tata Kerja Perangkat Daerah, Dinas Kebudayaan dan Pariwisata Kota Medan berganti nama menjadi Biro Pariwisata Kota Medan. Urusan kebudayaan dibagi menjadi Biro Kebudayaan Kota Medan. Visi dan misi Dinas Pariwisata Kota Medan:

1. Visi

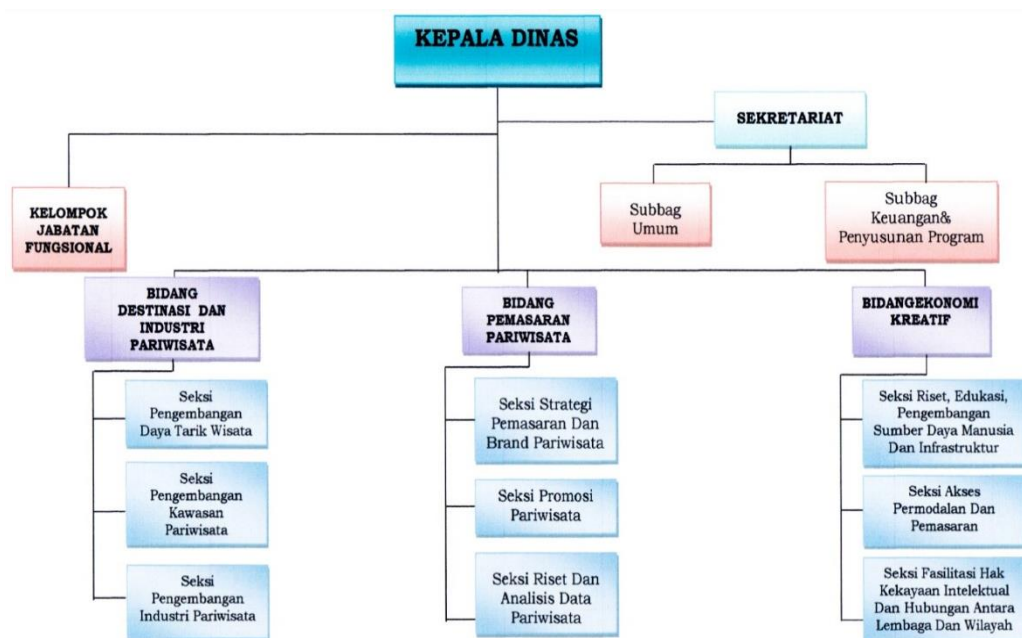
“Menjadi Kota Masa Depan yang Mutikultural, Berdaya Saing, Humanis, Sejahtera, dan Religius”.

2. Misi

“Meningkatkan efisiensi melalui deregulasi dan debirokratisasi sekaligus penciptaan iklim investasi yang semakin kondusif termasuk pengembangan kreatifitas dan inovasi daerah guna meningkatkan kemampuan kompetitif serta komparatif daerah.

4.1.1.1 Struktur Organisasi Dinas Pariwisata Kota Medan

Struktur organisasi merupakan susunan tentang pembagian tugas, wewenang, tanggung jawab dan yang saling berinteraksi. Adapun struktur organisasi pada Dinas Pariwisata Kota Medan adalah:



Gambar 4.1 Struktur Organisasi Dinas Pariwisata Kota Medan

4.1.2 Analisis Sistem Yang Sedang Berjalan

4.1.2.1 Prosedur Sistem Yang Sedang Berjalan

Berikut ini adalah uraian yang menjelaskan tentang sistem yang sedang berjalan pada sistem pariwisata kota medan yang ada di Dinas Pariwisata Kota Medan.

1. Pada sistem Dinas Pariwisata Kota Medan saat ini memberikan sebuah berita

dan informasi wisata-wisata yang berada di kota medan.

2. Pada saat melihat sebuah objek wisata wisatawan ataupun masyarakat hanya bisa melihat informasi wisata dan titik lokasi wisata, akan tetapi tidak semua objek wisata disediakan fitur yang memiliki titik lokasi objek wisata.

4.1.2.2 Identifikasi Masalah Sistem Yang Sedang Berjalan

Dilihat dari prosedur sistem yang sedang berjalan, penulis menilai bahwa alur tersebut terdapat permasalahan yaitu mulai dari penampilan titik lokasi pariwisata sampai *route* yang akan dilalui oleh wisatawan dan masyarakat masih belum efektif dan efisien serta tidak memberitahu rekomendasi yang lain yang diberikan oleh sistem yang sedang berjalan saat ini pada para wisatawan dan masyarakat. Adapun bentuk dari permasalahan dari sistem yang berjalan saat ini adalah:

1. Dalam pencarian pariwisata kota medan masih belum menampilkan atau tidak menyediakan semua peta titik lokasi pariwisata.
2. Sistem yang sedang berjalan ini tidak memberitahu pada *user* *route* mana yang akan dilaluinya dengan jarak terdekat.
3. Masih belum memiliki fitur rekomendasi pariwisata yang di dekat dengan *user* berada.
4. *User* tidak diberi akses untuk menambahkan lokasi pariwisata kota medan.

4.1.3 Analisis Sistem Usulan

Adapun saran dan masukan dari peneliti untuk mengusulkan membangun sebuah sistem yang baru dalam memperluas promosi pariwisata. Sistem yang dibangun ini diharapkan dapat membantu Dinas Pariwisata Kota Medan dalam mempromosikan pariwisata serta dapat mempermudah para wisatawan dan masyarakat ketika berkunjung ke suatu objek pariwisata. Adapun gambaran sistem yang di usulkan dan dibangun, yaitu:

1. Sistem yang diusulkan oleh peneliti adalah sistem berbasis jaringan. Sehingga wisatawan dan penduduk lokal dapat dengan mudah mengunjunginya.
2. Membantu Dinas Pariwisata Kota Medan untuk mempromosikan berbagai destinasi wisata di Kota Medan.

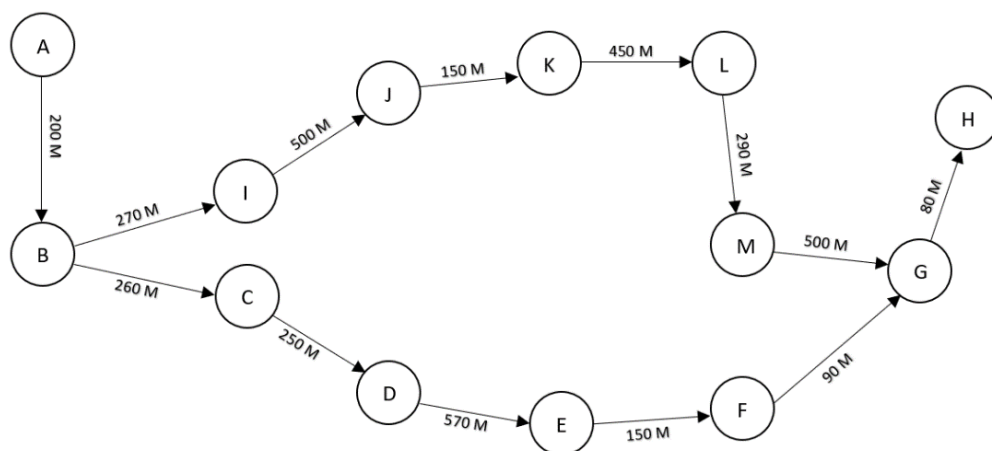
3. Pada sistem ini wisatawan akan dapat dengan mudah menemukan *route* terdekat yang akan dilaluinya dan dengan tampilan pemberitahuan arah jalan yang akan dilalui.
4. Sistem ini juga akan menampilkan sebuah rekomendasi pariwisata terdekat dari titik user berada.
5. Dan sistem ini juga menyediakan *form* tambah lokasi pariwisata yang dapat diisi oleh para wisatawan dan masyarakat dengan verifikasi yang dilakukan oleh admin ketika sudah di tambah.

4.1.4 Analisis Algoritma *Dijkstra* SIG Pariwisata Kota Medan

Untuk menghasilkan *route* Sistem Informasi Geografis Pariwisata Kota Medan dengan algoritma *Dijkstra* dilakukan dengan tahap sebagai berikut:

1. Membuat *graf* pada kasus yang sudah ditentukan.
2. Membuat tabel jarak
 - a. Dibuat tabel antar titik yang saling berhubungan
 - b. Dengan algoritma *Dijkstra* mencari titik dengan *route* terkecil dari titik awal menuju titik akhir.

Adapun contoh kasus yang sudah ditentukan dari *graf* yang ditunjukkan pada Gambar 4.2 dengan titik awal = A, dan titik tujuan = H dengan algoritma *Dijkstra* untuk menentukan *route* terpendek.



Gambar 4.2 *Graf* Contoh Kasus Algoritma *Dijkstra*

$A \rightarrow 0$, $B \rightarrow 200$ m. titik awal dimulai dari A dengan jarak 0 dan titik yang dituju B dengan jarak tempuh 200 m.

2. Jarak Jln.Bukit Barisan dengan Jln.St.KA.

Tabel 4.3 Hasil Iterasi Ke-2

From A to >>	B	C	D	E	F	G	H	I	J	K	L	M
B	200	460	∞	∞	∞	∞	∞	470	∞	∞	∞	∞
	A	B						B				

$B \rightarrow 200$ m, $C \rightarrow 260$ m, $I \rightarrow 270$ m. Jarak awal yaitu $A = 0$, maka titik B dengan jarak 200 m tetap. Kemudian jarak dari titik B ke titik C akan dijumlahkan dengan ditambah dari titik B dan $C = 200 + 260 = 460$, tetapi titik B dapat melewati titik I dikarenakan terdapat jalur yang dilewati dari titik B dan $I = 200 + 270 = 470$ m, maka dapatlah jalur yang terkecil yaitu titik C.

3. Jarak Jln.St.Ka dengan Jln.Pulau Pinang.

Tabel 4.4 Hasil Iterasi Ke-3

From A to >>	B	C	D	E	F	G	H	I	J	K	L	M
C	200	460	710	∞	∞	∞	∞	470	∞	∞	∞	∞
	A	B	C					B				

Titik C ke titik $D \rightarrow 250$ m. dari titik A,B, dan C sebelumnya yang berjumlah 460 m akan ditambahkan dengan tujuan yang dilewati yaitu titik D dengan jarak 250 m. Maka jumlah dari titik C dan D adalah $460 + 250 = 710$ m.

4. Jarak Jln.Pulau Pinang dengan Jln.Mesjid.

Tabel 4.5 Hasil Iterasi Ke-4

From A to >>	B	C	D	E	F	G	H	I	J	K	L	M
D	200	460	710	1.28	∞	∞	∞	470	∞	∞	∞	∞
	A	B	C	D				B				

Titik D ke titik E \rightarrow 570 m. Dari titik A,B,C dan D sebelumnya yang berjumlah 710 m akan ditambahkan dengan tujuan yang dilewati yaitu titik E dengan jarak 570 m. Maka jumlah dari titik D dan E adalah $710 + 570 = 1.280$ km.

5. Jarak Jln.Mesjid dengan Jln.Palang Merah.

Tabel 4.6 Hasil Iterasi Ke-5

From A to >>	B	C	D	E	F	G	H	I	J	K	L	M
E	200	460	710	1.28	1.43	∞	∞	470	∞	∞	∞	∞
	A	B	C	D	E			B				

Titik E ke titik F \rightarrow 150 m. Dari titik A,B,C,D dan E berjumlah 1.280 km akan ditambahkan dengan titik yang dilewati yaitu F, dari titik E ke titik F adalah $1.280 + 150 = 1.430$ km.

6. Jarak Jln.Palang Merah dengan Jln.Ahmad Yani.

Tabel 4.7 Hasil Iterasi Ke-6

From A to >>	B	C	D	E	F	G	H	I	J	K	L	M
F	200	460	710	1.28	1.43	1.52	∞	470	∞	∞	∞	∞

	A	B	C	D	E	F		B				
--	---	---	---	---	---	---	--	---	--	--	--	--

Titik F ke titik G \rightarrow 90 m. Dari titik A,B,C,D,E dan F sebelumnya berjumlah 1.430 km akan ditambahkan dengan titik yang dilewati yaitu G, dari titik F ke titik G adalah $1.430 + 90 = 1.520$ km.

7. Jarak Jln.Ahmad Yani dengan Tjong A Fie Mansion.

Tabel 4.8 Hasil Iterasi Ke-7

From A to >>	B	C	D	E	F	G	H	I	J	K	L	M
G	200	460	710	1.28	1.43	1.52	1.6	470	∞	∞	∞	∞
	A	B	C	D	E	F	G	B				

Titik G ke titik H \rightarrow 80 m. Dari titik A,B,C,D,E,F dan G sebelumnya berjumlah 1.520 km akan ditambahkan dengan titik yang dilewati yaitu H, dari titik G ke titik H adalah $1.520 + 80 = 1.6$ km.

Maka hasil untuk menentukan jarak terpendek dari titik awal = A ke titik tujuan = H sudah didapatkan dengan kesimpulan dari setiap titik, sebagai berikut:

1. Jalur terpendek dari A ke B adalah A-B dengan *route* 200 m.
2. Jalur terpendek dari A ke C adalah A-B-C dengan *route* 460 m.
3. Jalur terpendek dari A ke D adalah A-B-C-D dengan *route* 710 m.
4. Jalur terpendek dari A ke E adalah A-B-C-D-E dengan *route* 1.28 km.
5. Jalur terpendek dari A ke F adalah A-B-C-D-E-F dengan *route* 1.43 km.
6. Jalur terpendek dari A ke G adalah A-B-C-D-E-F-G dengan *route* 1.52 km.
7. Jalur terpendek dari A ke G adalah A-B-C-D-E-F-G-H dengan *route* 1.6 km.

Dari tabel iterasi yang sudah diperhitungkan dan dipilih, maka akan menghasilkan jalur terpendek dari titik A ke setiap titik yang ada untuk menuju ke titik H. Untuk melihat jalur mana yang terpilih dapat ditelusur hasil titik A ke titik H yang di dapatkan adalah A-B-C-D-E-F-G-H-G.

4.2 Design

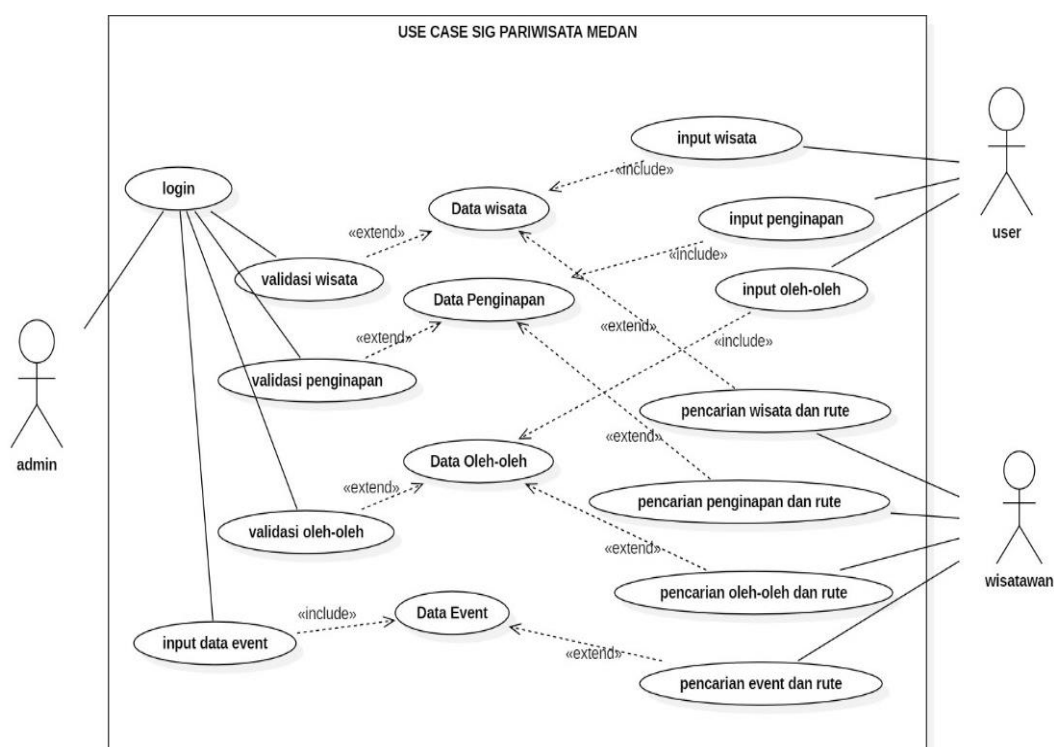
Setelah menganalisis sistem, tahap selanjutnya adalah perancangan sistem. Tujuan dari perancangan sistem adalah untuk memahami gambaran dari sistem. Pada tahap ini dilakukan perancangan proses, perancangan basis data, dan perancangan antarmuka sistem.

4.2.1 Desain Proses

Dalam membangun sebuah sistem, diperlukan alat bantu rancangan yang memudahkan peneliti dalam membangun sebuah sistem seperti *use case* diagram, *activity* diagram, *sequence* diagram.

4.2.1.1 Use Case Diagram

Use case diagram digunakan untuk melakukan tugas-tugas tertentu yang menggambarkan proses bisnis sistem. Berikut ini adalah perancangan *use case* diagram pada studi kasus Sistem Informasi Geografis Pariwisata Kota Medan yang akan dibangun.

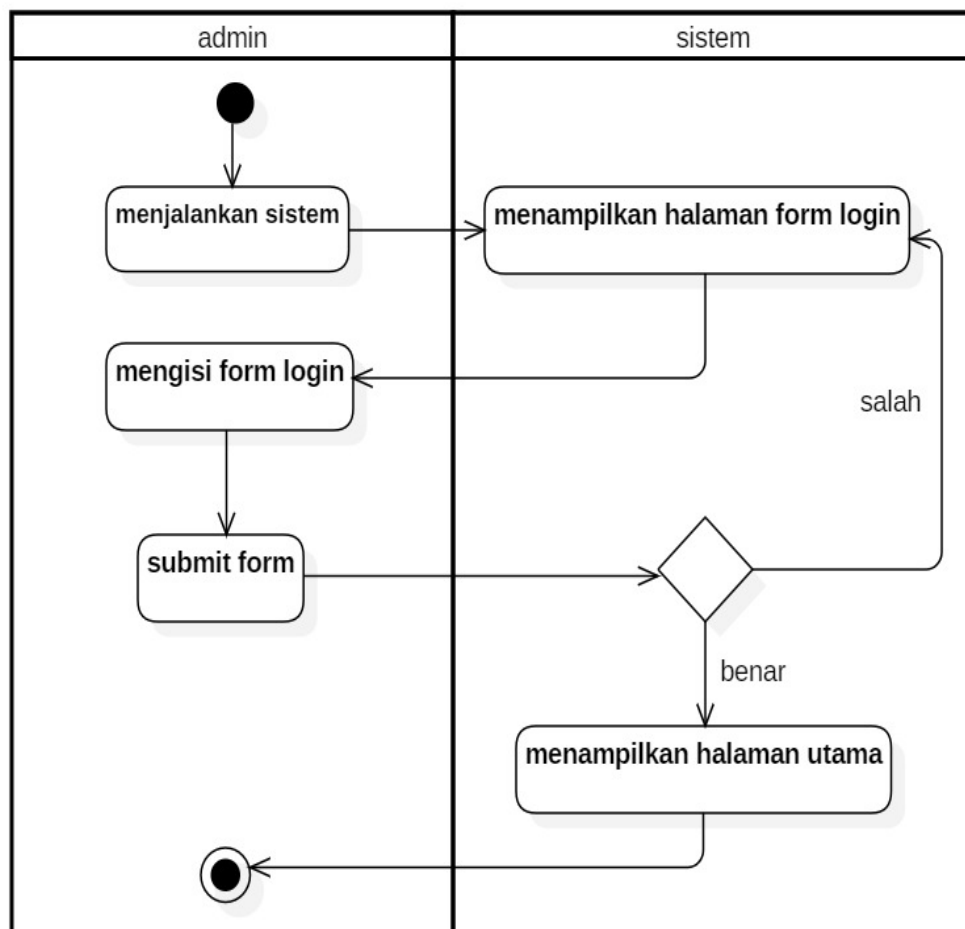


Gambar 4.3 Use Case Diagram SIG Pariwisata Kota Medan

4.2.1.2 Activity Diagram

1. Activity Diagram Login Admin

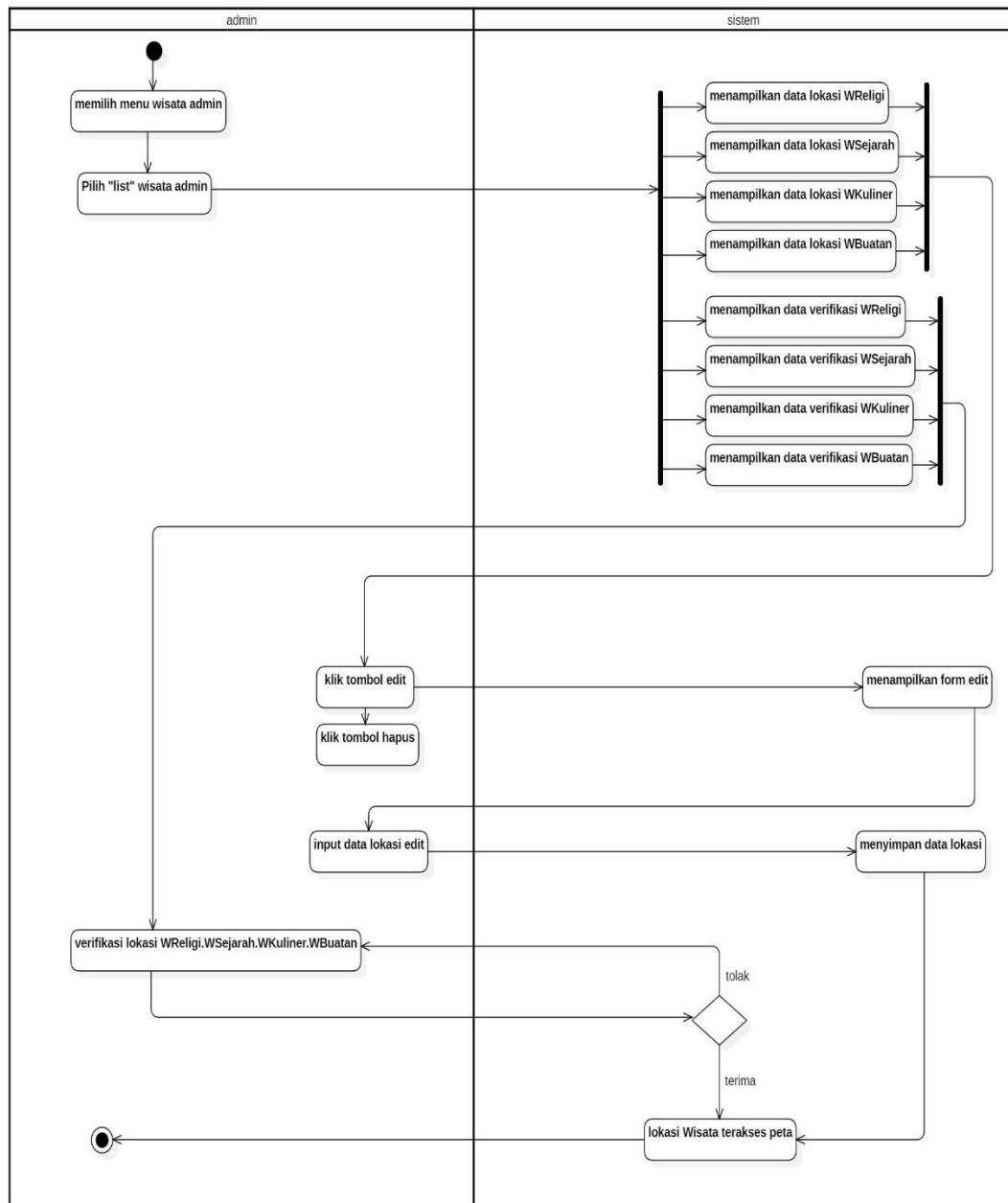
Aktivitas *login* admin dijelaskan langkah demi langkah, dimulai dari aktivitas admin yang menjalankan sistem, kemudian mengisi *form login* dan tekan *submit* untuk masuk ke halaman manajemen. Diagram aktivitas dirancang sebagai berikut:



Gambar 4.4 Activity Diagram Login Admin

2. Activity Diagram Admin Wisata

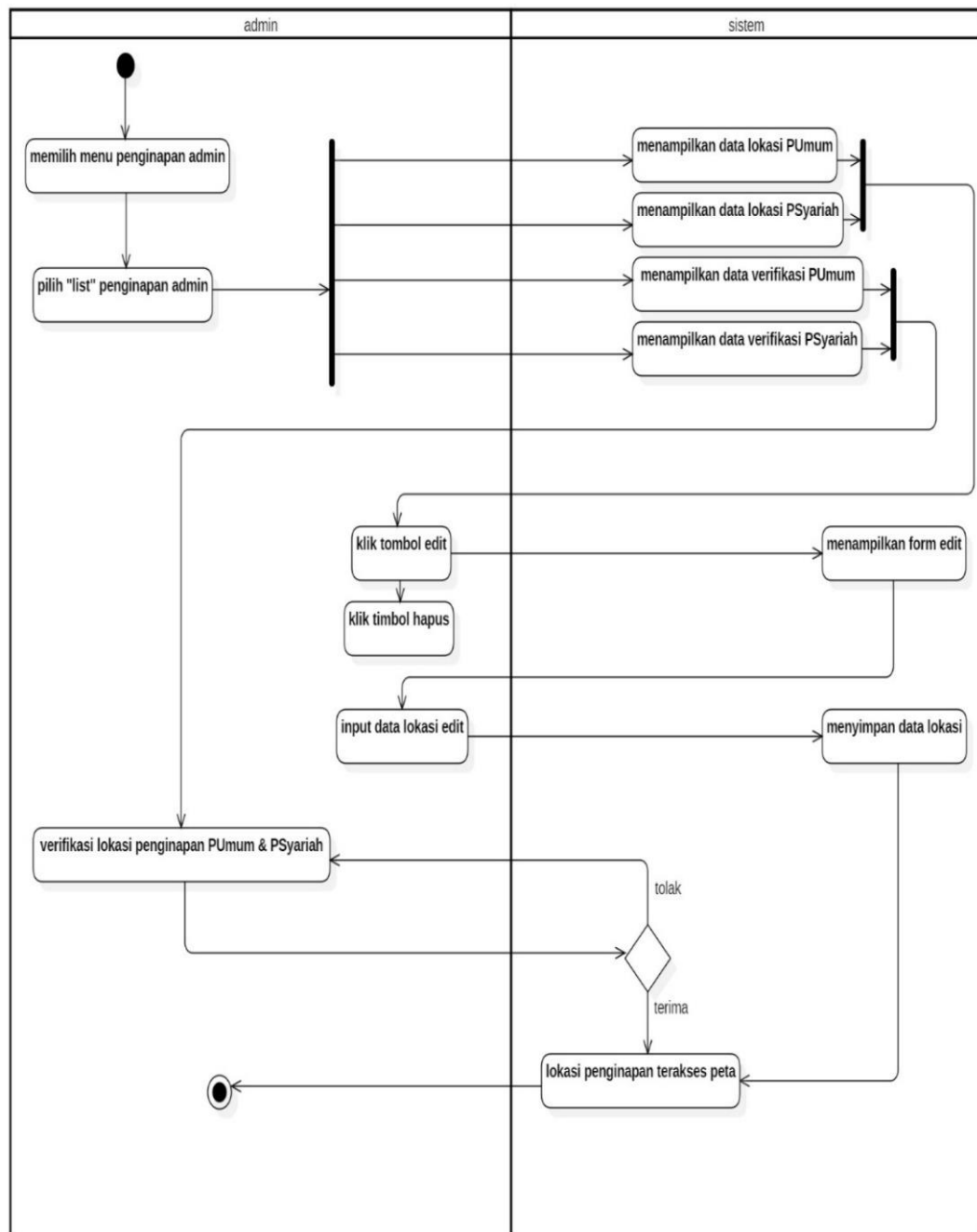
Aktivitas admin dalam memverifikasi, edit dan hapus wisata yang berada di kota medan diterangkan pada langkah-langkah state pada gambar di bawah ini. Berikut rancangan Activity diagram admin wisata:



Gambar 4.5 Activity Diagram Admin Wisata

3. Activity Diagram Admin Penginapan

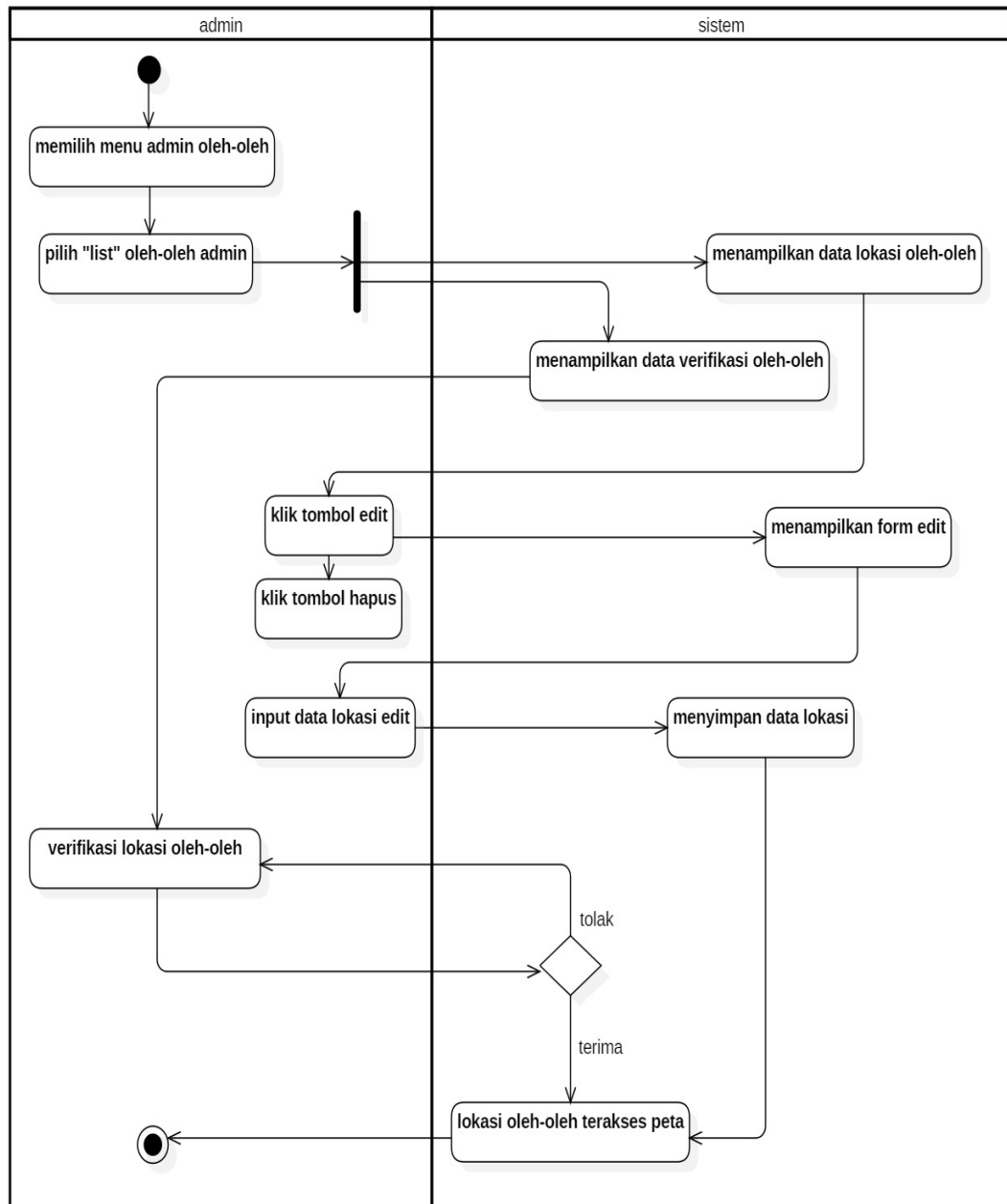
Aktivitas admin dalam memverifikasi, edit dan hapus penginapan yang berada di kota medan diterangkan pada langkah-langkah state pada gambar di bawah ini. Berikut rancangan *Activity* diagram admin penginapan:



Gambar 4.6 Activity Diagram Admin Penginapan

4. Activity Diagram Admin Oleh-oleh

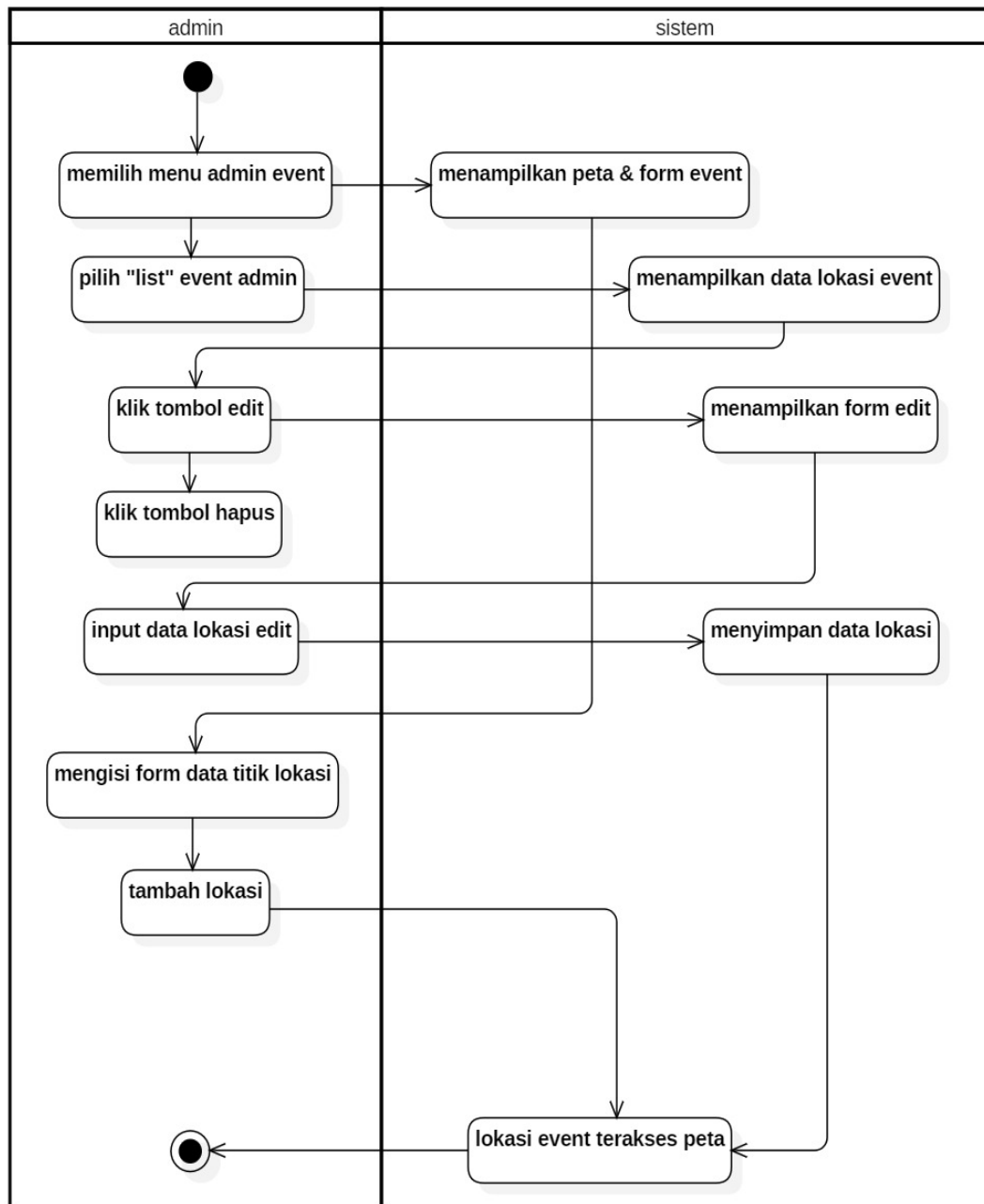
Aktivitas admin dalam memverifikasi, edit dan hapus oleh-oleh yang berada di kota medan diterangkan pada langkah-langkah state pada gambar di bawah ini. Berikut rancangan *Activity* diagram admin oleh-oleh:



Gambar 4.7 Activity Diagram Admin Oleh-oleh

5. Activity Diagram Admin Event

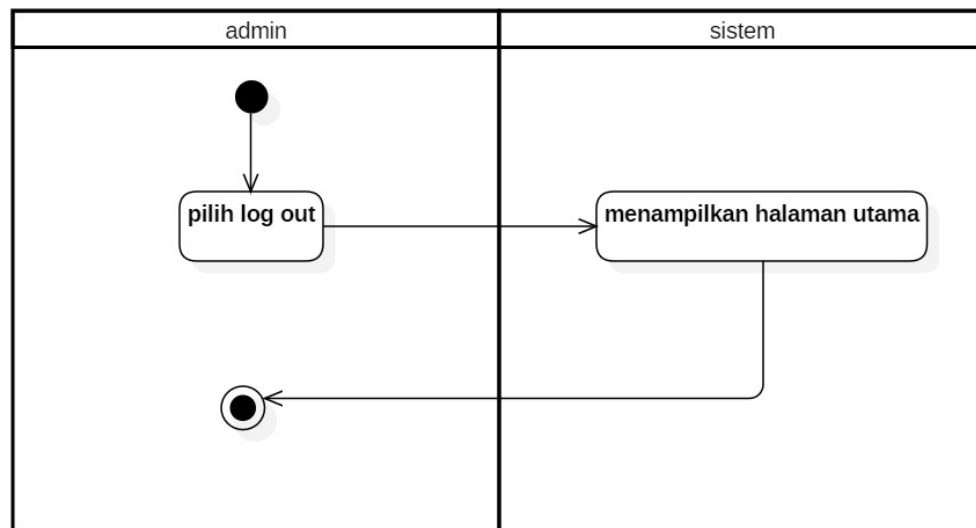
Aktivitas admin dalam menambah *event* yang diselenggarakan pihak Dinas Pariwisata Kota Medan, serta edit dan hapus *event* yang diterangkan pada langkah-langkah state pada gambar di bawah ini. Berikut rancangan Activity diagram admin *event*:



Gambar 4.8 Activity Diagram Admin Event

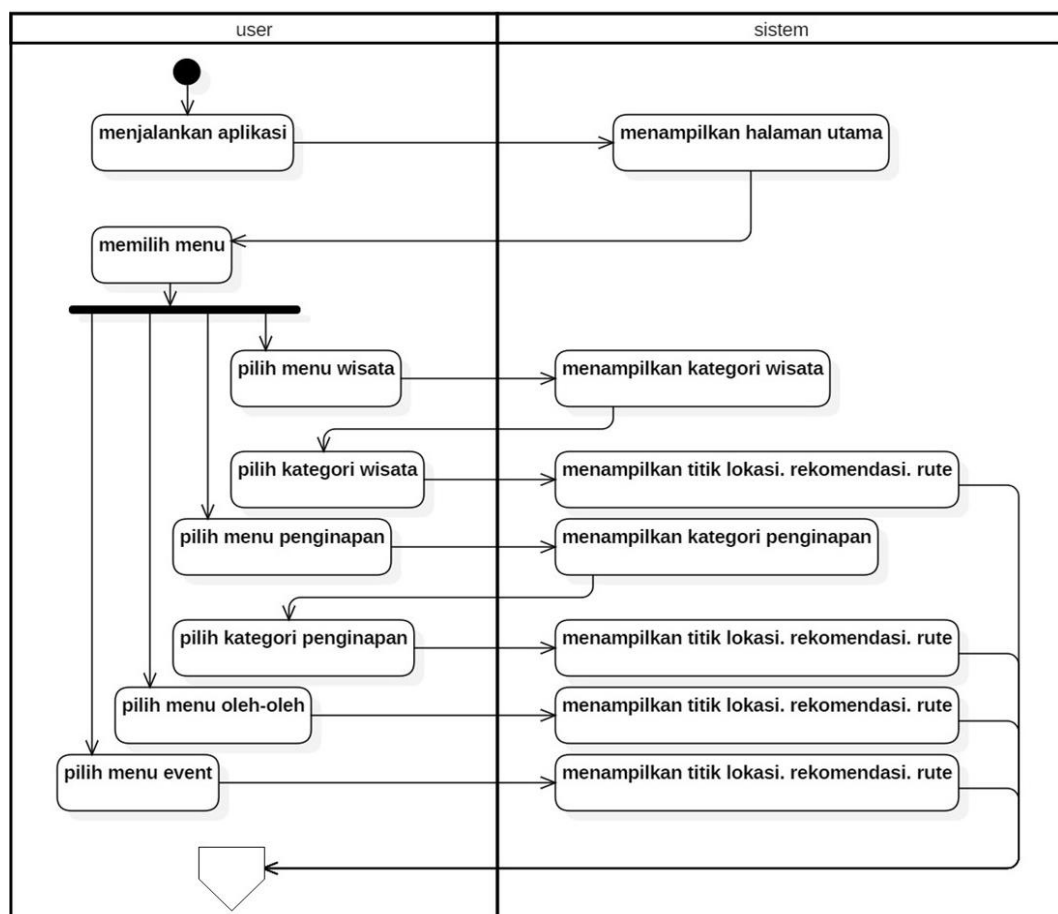
6. Activity Diagram Admin Logout

Aktivitas admin dalam memverifikasi, edit dan hapus penginapan yang berada di kota medan diterangkan pada langkah-langkah state pada gambar di bawah ini. Berikut rancangan Activity diagram admin penginapan:

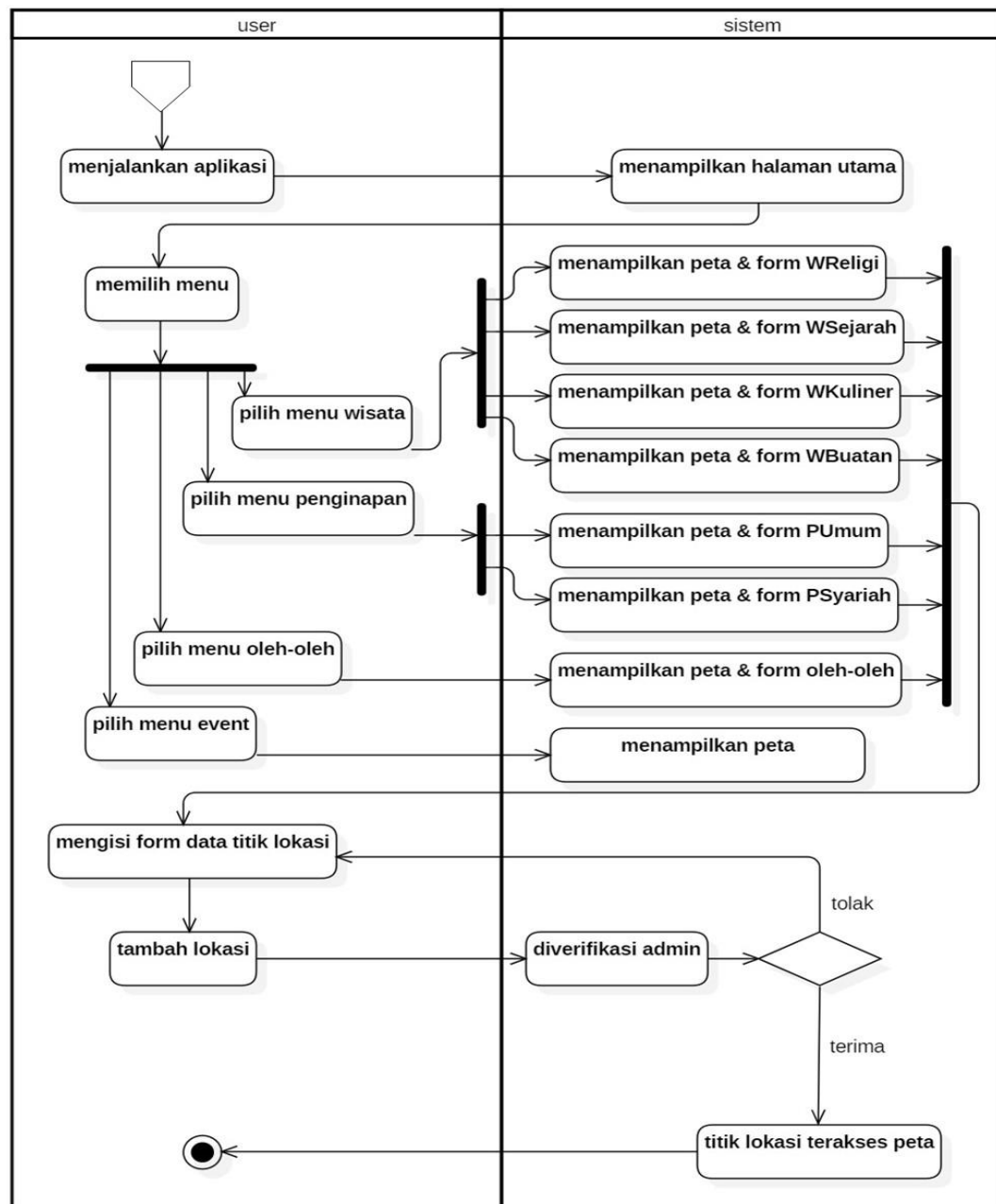


Gambar 4.9 Activity Diagram Admin Logout

7. Activity Diagram User



Gambar 4.10 Activity Diagram User



Gambar 4.11 Activity Diagram User

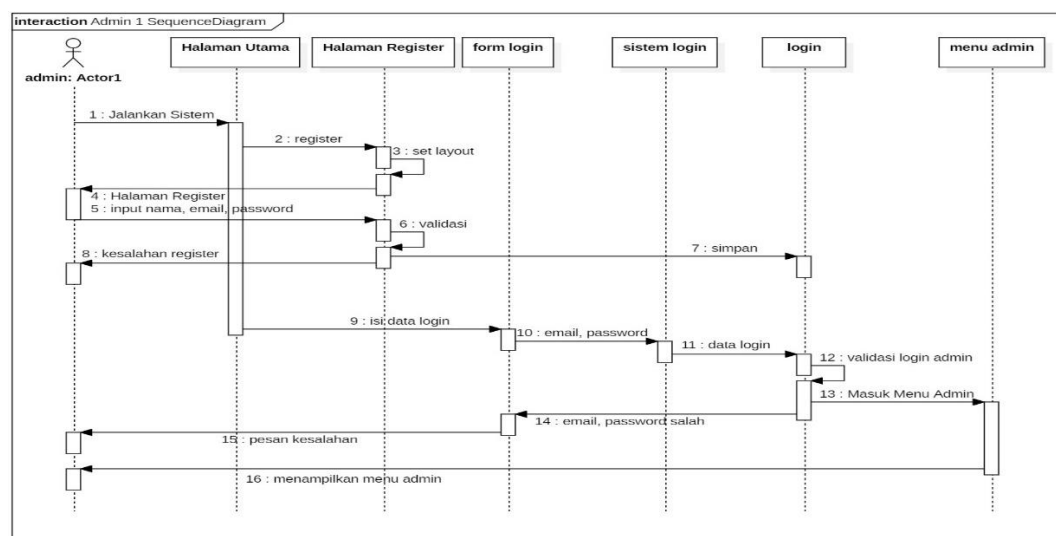
Aktivitas *user* diterangkan dalam langkah-langkah state, dimulai dari *user* bisa melihat titik lokasi objek pariwisata, rekomendasi pariwisata dari titik *user* berada serta memberitahukan *route* yang akan dilaluinya. Kemudian *user* juga bisa menambah objek pariwisata yang ada di kota medan dengan diverifikasi oleh admin terlebih dahulu. Berikut rancangan dari Activity diagram *user* yang ada pada gambar di atas.

4.2.1.3 Sequence Diagram

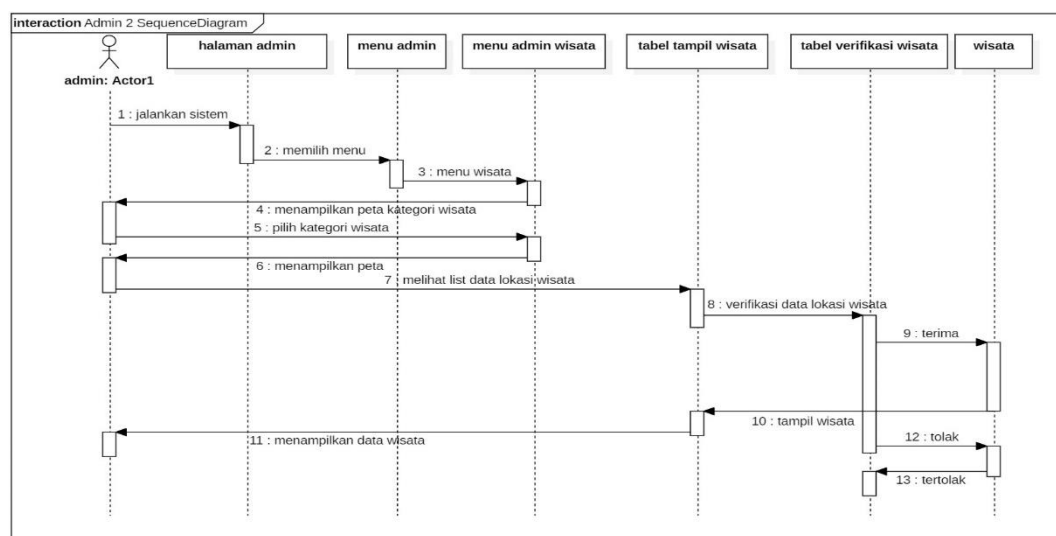
Adapun *sequence* diagram Sistem Informasi Geografis Pariwisata Kota Medan dapat digambarkan sebagai berikut:

1. Sequence Diagram Admin

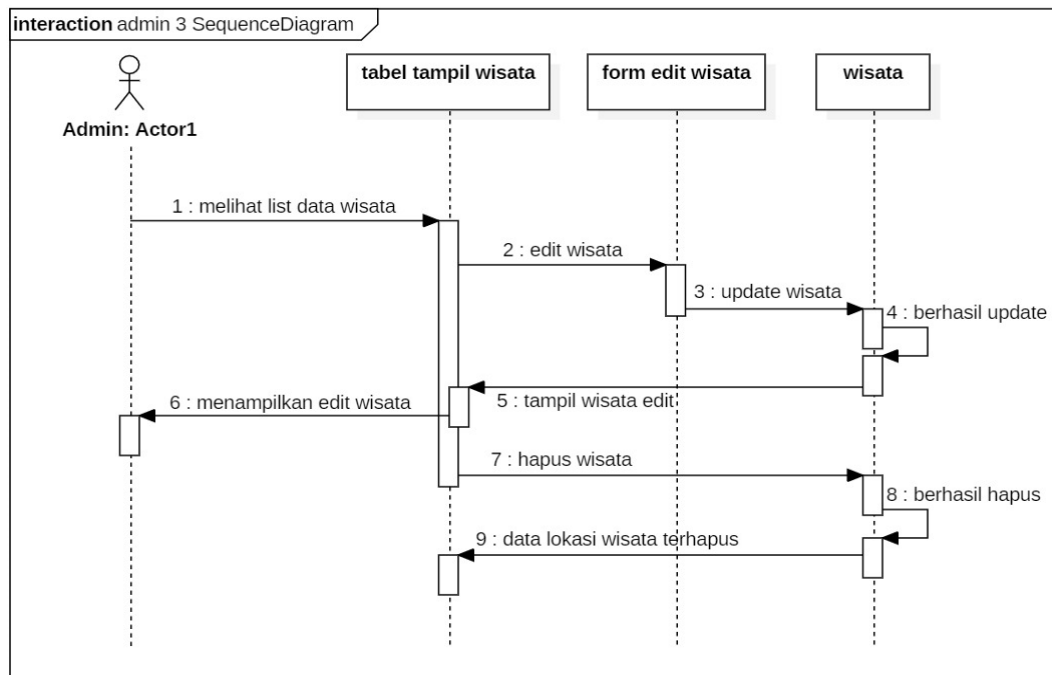
Sequence diagram admin menjelaskan bahwa terdapat kotak aktivasi, dan garis-garis di dalamnya memberitahu aktor atau objek aktivitas yang terjadi saat berinteraksi dengan objek lain:



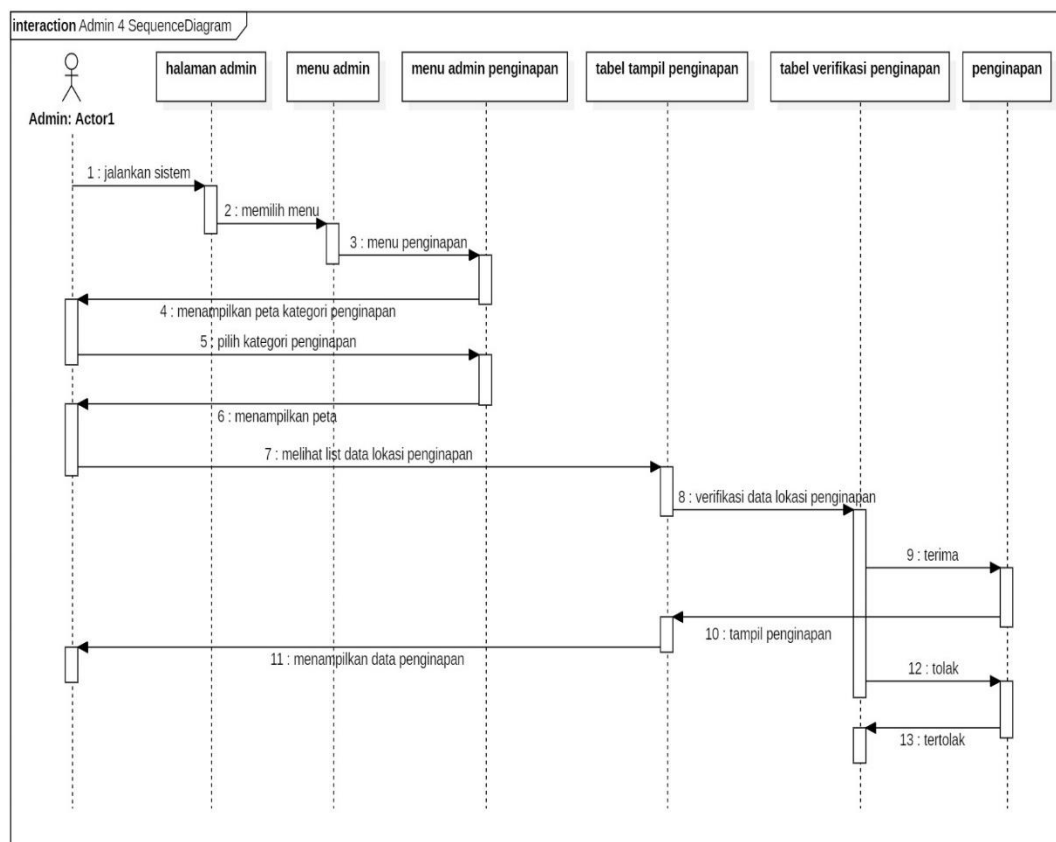
Gambar 4.12 *Sequence* Diagram Admin Register dan Login



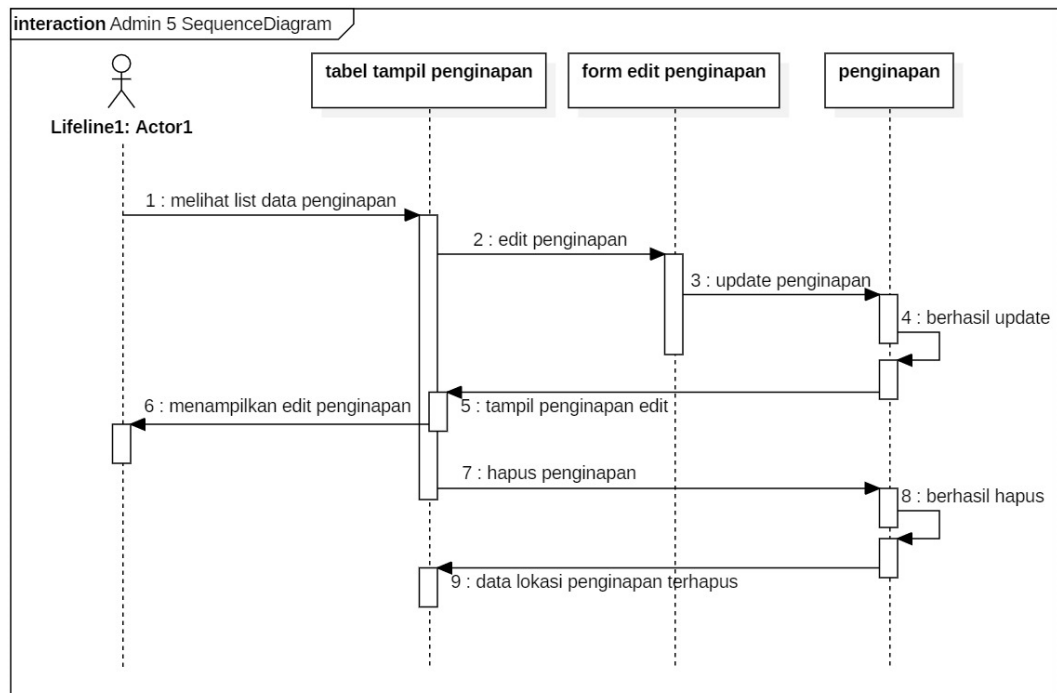
Gambar 4.13 *Sequence* Diagram Admin Wisata



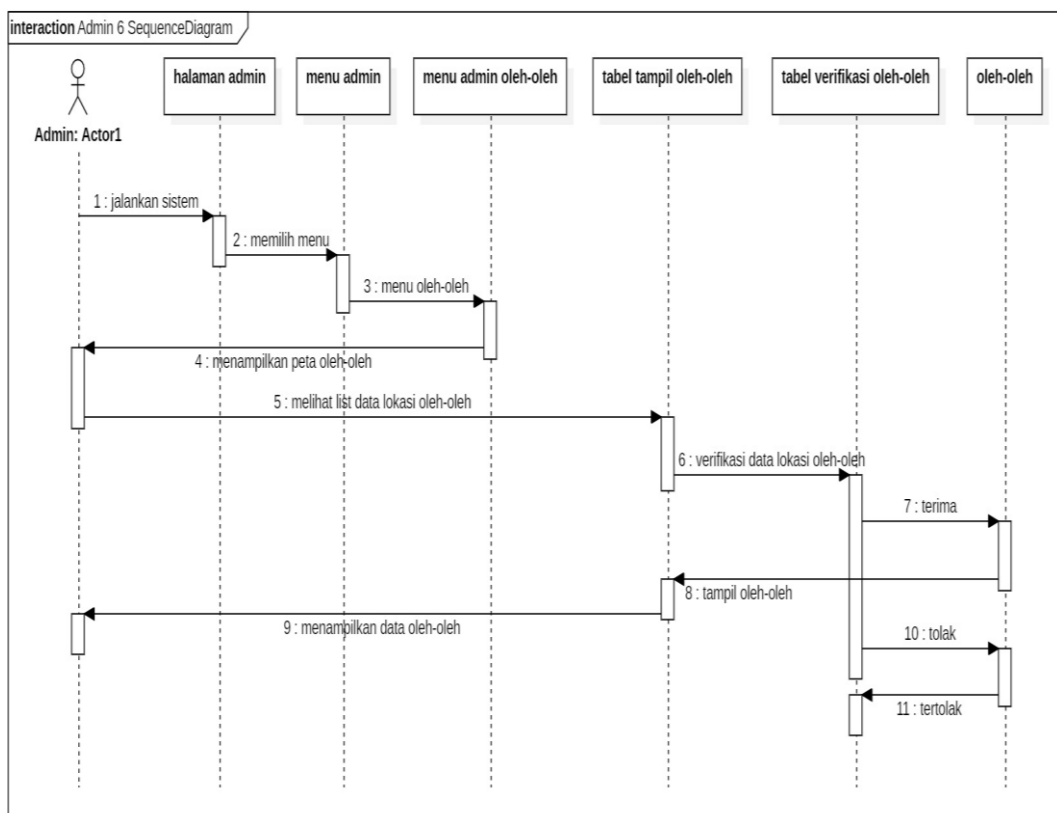
Gambar 4.14 Sequence Diagram Admin Wisata Edit Dan Hapus



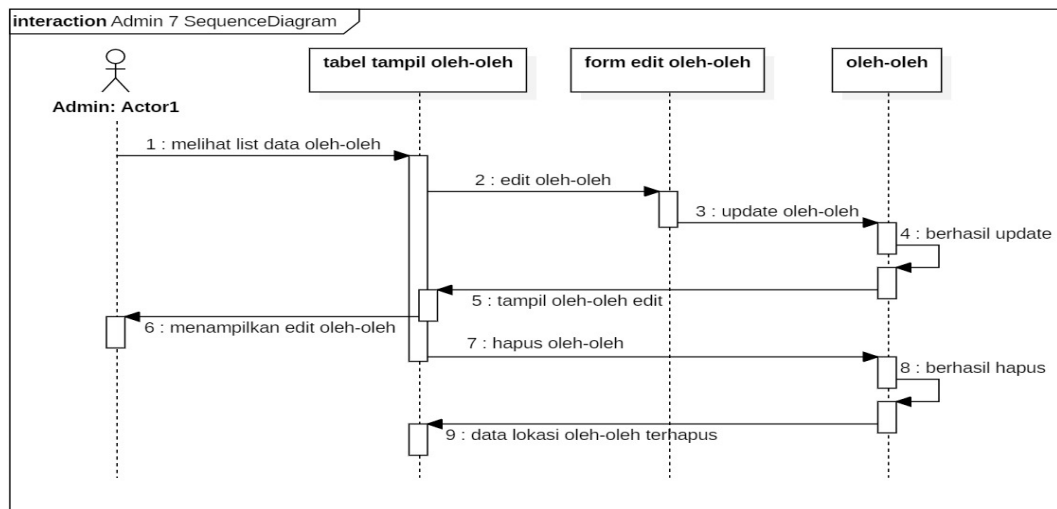
Gambar 4.15 Sequence Diagram Admin Penginapan



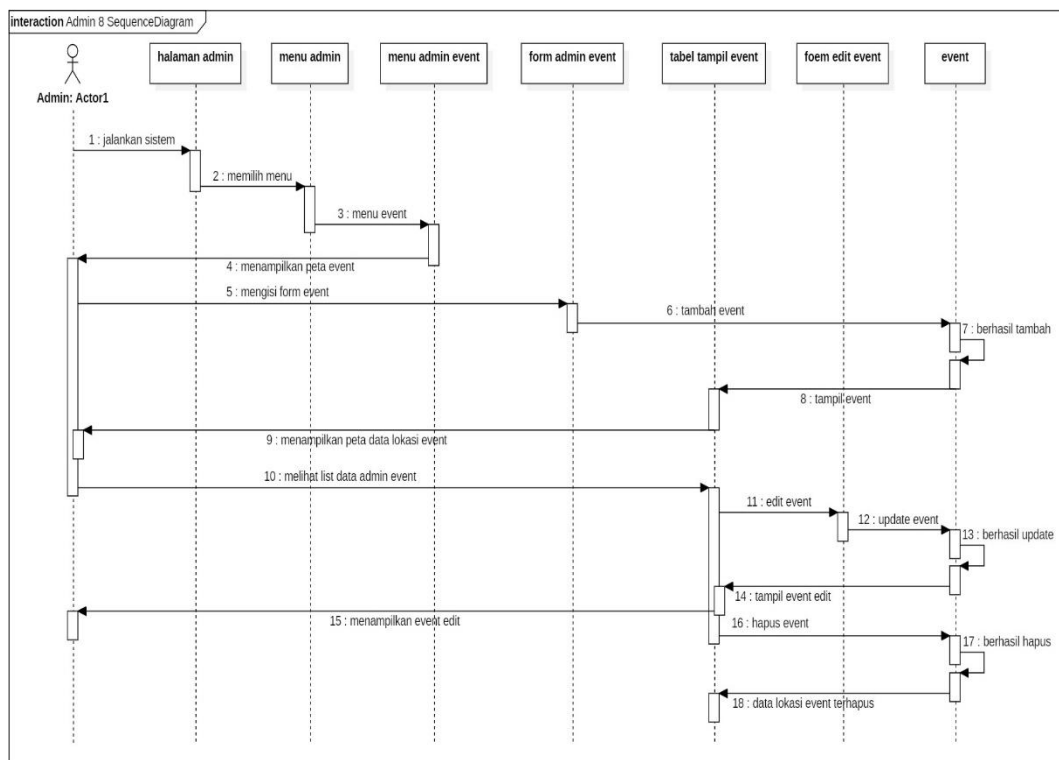
Gambar 4.16 Sequence Diagram Admin Penginapan Edit Dan Hapus



Gambar 4.17 Sequence Diagram Admin Oleh-oleh



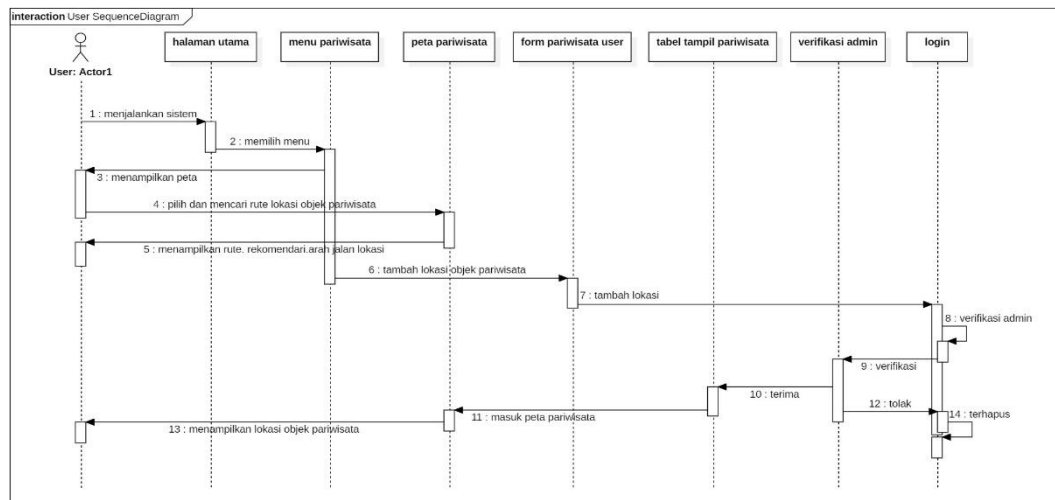
Gambar 4.18 *Sequence Diagram Admin Oleh-oleh Edit Dan Hapus*



Gambar 4.19 *Sequence Diagram Admin Event*

2. Sequence Diagram User

Sequence diagram user urutan pengguna menjelaskan bahwa ada kotak aktivasi di mana garis memberi tahu aktor atau objek aktivitas yang terjadi ketika berinteraksi dengan objek lain, berikut *Sequence diagram user*:



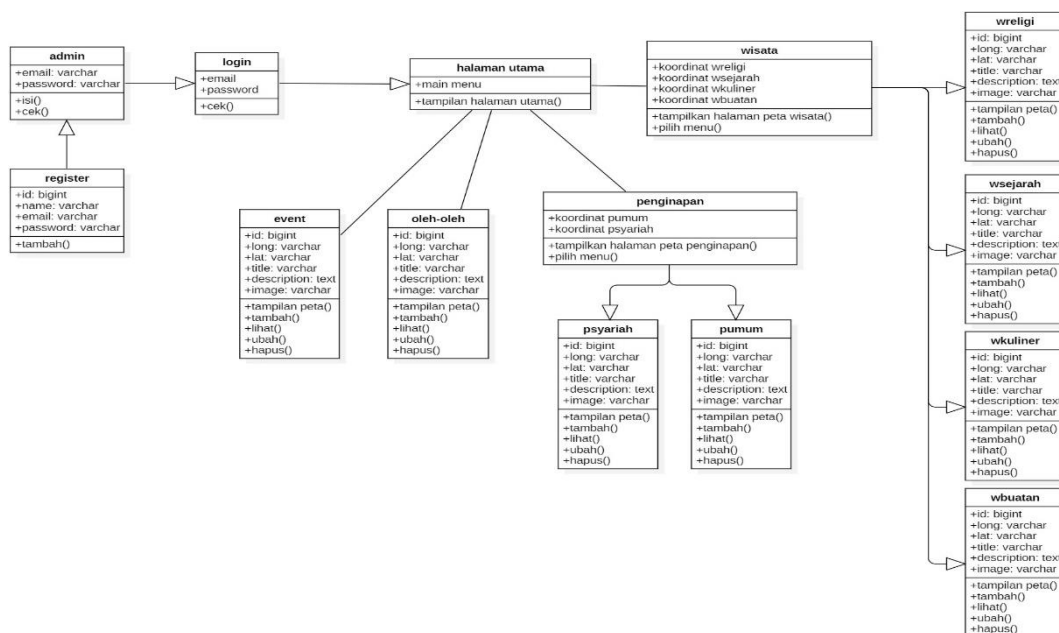
Gambar 4.20 Sequence Diagram User

4.2.2 Desain Database

Pada tahap ini menjelaskan tentang desain database seperti *class diagram*, dan spesifikasi tabel pada *database*.

4.2.2.1 Class Diagram

Class diagram merupakan hubungan yang saling berinteraksi untuk tujuan tertentu. Berikut adalah bentuk *class diagram* untuk sistem yang akan dibangun:



Gambar 4.21 Class Diagram SIG Pariwisata Kota Medan

4.2.2.2 Spesifikasi Tabel *Database*

Adapun rancangan database dalam pembangunan sistem ini adalah terdiri dari beberapa tabel yang diantaranya:

1. Tabel Admin

Nama tabel: users

Adapun struktur tabel users sebagai berikut:

Tabel 4.9 Atribut Users

No.	Nama <i>Field</i>	Tipe Data	Ukuran
1.	Id	Bigint	20
2.	Name	Varchar	255
3.	Email	Varchar	255
4.	Password	Varchar	255
5.	Created_at	Timestamp	-
6.	Updated_at	Timestamp	-

2. Tabel Wisata Religi

Nama tabel: religis

Adapun struktur tabel religis sebagai berikut:

Tabel 4.10 Atribut Religis

No.	Nama <i>Field</i>	Tipe Data	Ukuran
1.	Id	Bigint	20
2.	Long	Varchar	255
3.	Lat	Varchar	255

4.	Title	Varchar	255
5.	Description	Text	-
6.	Image	Varchar	255
7.	Status	Tinyint	1
8.	Created_at	Timestamp	-
9.	Updated_at	Timestamp	-

3. Tabel Sejarah

Nama tabel: sejarahs

Adapun struktur tabel sejarahs sebagai berikut:

Tabel 4.11 Atribut Sejarahs

No.	Nama <i>Field</i>	Tipe Data	Ukuran
1.	Id	Bigint	20
2.	Long	Varchar	255
3.	Lat	Varchar	255
4.	Title	Varchar	255
5.	Description	Text	-
6.	Image	Varchar	255
7.	Status	Tinyint	1
8.	Created_at	Timestamp	-
9.	Updated_at	Timestamp	-

4. Tabel Kuliner

Nama tabel: kuliners

Adapun struktur tabel kuliners sebagai berikut:

Tabel 4.12 Atribut Kuliners

No.	Nama <i>Field</i>	Tipe Data	Ukuran
1.	Id	Bigint	20
2.	Long	Varchar	255
3.	Lat	Varchar	255
4.	Title	Varchar	255
5.	Description	Text	-
6.	Image	Varchar	255
7.	Status	Tinyint	1
8.	Created_at	Timestamp	-
9.	Updated_at	Timestamp	-

5. Tabel Buatan

Nama tabel: buatans

Adapun struktur tabel buatans sebagai berikut:

Tabel 4.13 Atribut Buatans

No.	Nama <i>Field</i>	Tipe Data	Ukuran
1.	Id	Bigint	20
2.	Long	Varchar	255
3.	Lat	Varchar	255
4.	Title	Varchar	255

5.	Description	Text	-
6.	Image	Varchar	255
7.	Status	Tinyint	1
8.	Created_at	Timestamp	-
9.	Updated_at	Timestamp	-

6. Tabel Penginapan Syariah

Nama tabel: syariahs

Adapun struktur tabel syariahs sebagai berikut:

Tabel 4.14 Atribut Syariahs

No.	Nama <i>Field</i>	Tipe Data	Ukuran
1.	Id	Bigint	20
2.	Long	Varchar	255
3.	Lat	Varchar	255
4.	Title	Varchar	255
5.	Description	Text	-
6.	Image	Varchar	255
7.	Status	Tinyint	1
8.	Created_at	Timestamp	-
9.	Updated_at	Timestamp	-

7. Tabel Penginapan Umum

Nama tabel: umums

Adapun struktur tabel umums sebagai berikut:

Tabel 4.15 Atribut Umums

No.	Nama <i>Field</i>	Tipe Data	Ukuran
1.	Id	Bigint	20
2.	Long	Varchar	255
3.	Lat	Varchar	255
4.	Title	Varchar	255
5.	Description	Text	-
6.	Image	Varchar	255
7.	Status	Tinyint	1
8.	Created_at	Timestamp	-
9.	Updated_at	Timestamp	-

8. Tabel Oleh-Oleh

Nama tabel: oleh_olehs

Adapun struktur tabel oleh_olehs sebagai berikut:

Tabel 4.16 Atribut Oleh_olehs

No.	Nama <i>Field</i>	Tipe Data	Ukuran
1.	Id	Bigint	20
2.	Long	Varchar	255
3.	Lat	Varchar	255
4.	Title	Varchar	255
5.	Description	Text	-

6.	Image	Varchar	255
7.	Status	Tinyint	1
8.	Created_at	Timestamp	-
9.	Updated_at	Timestamp	-

9. Tabel *Event*

Nama tabel: events

Adapun struktur tabel events sebagai berikut:

Tabel 4.17 Atribut Events

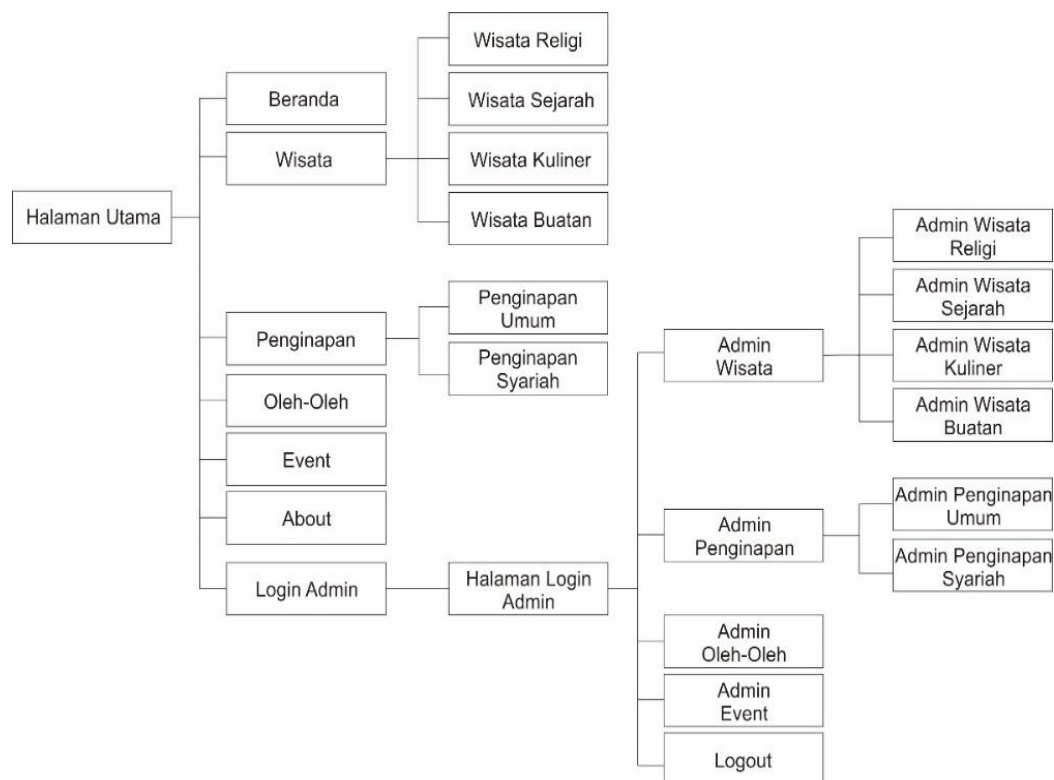
No.	Nama <i>Field</i>	Tipe Data	Ukuran
1.	Id	Bigint	20
2.	Long	Varchar	255
3.	Lat	Varchar	255
4.	Title	Varchar	255
5.	Description	Text	-
6.	Image	Varchar	255
7.	Status	Tinyint	1
8.	Created_at	Timestamp	-
9.	Updated_at	Timestamp	-

4.2.3 Desain Rancangan Antarmuka (*Interface*)

Desain rancangan antarmuka merupakan suatu gambaran dari sistem yang akan dibangun. Berikut ini merupakan *interface* dari sistem yang dibangun.

4.2.3.1 Struktur Menu

Struktur menu digunakan untuk memudahkan *user* dalam mengoperasikan Sistem Informasi Geografis Pariwisata Kota Medan pada Dinas Pariwisata Kota Medan.



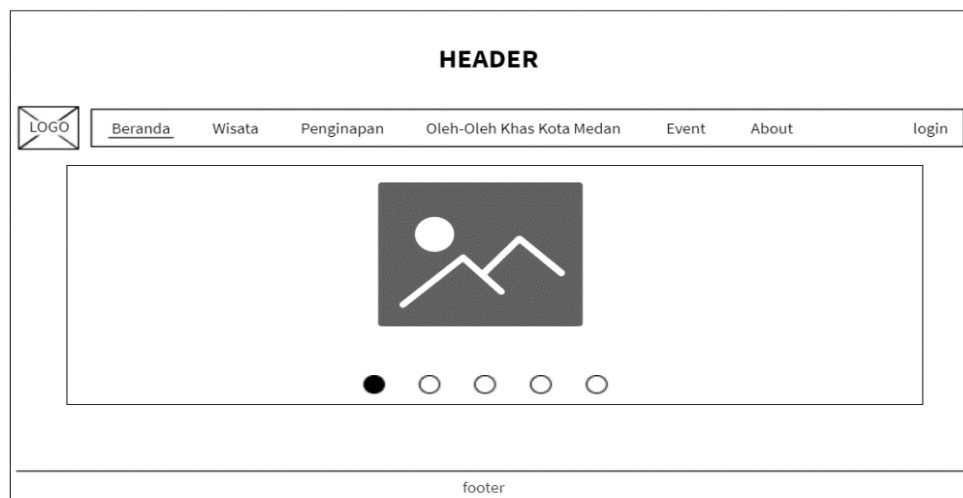
Gambar 4.22 Struktur Menu SIG Pariwisata Kota Medan

4.2.3.2 Perancangan Tampilan

Dibawah ini merupakan rancangan tampilan dari sistem yang akan dibangun agar pembuatan Sistem Informasi Geografis Pariwisata Kota Medan ini lebih tergambar.

1. Tampilan Beranda

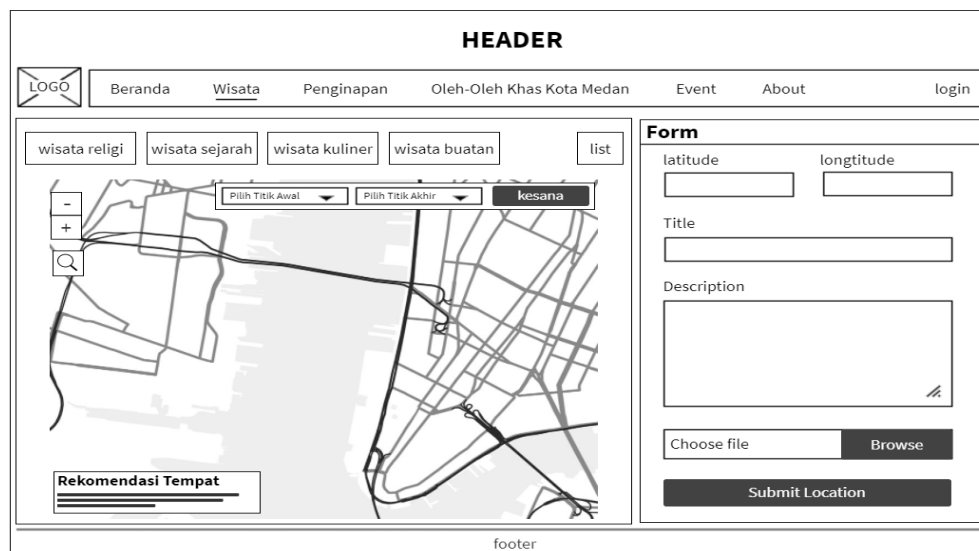
Tampilan beranda atau halaman utama adalah tampilan saat *user* membuka sebuah *website*. Berikut tampilan setelah membuka *website*:



Gambar 4.23 Tampilan Beranda

2. Tampilan Halaman Wisata

Terdapat halaman wisata yang di dalamnya terdapat 4 kategori wisata seperti wisata religi, wisata sejarah, wisata kuliner, wisata buatan. Dan terdapat *form* untuk input data wisata. Berikut tampilannya:




Gambar 4.24 Tampilan Halaman Wisata

3. Tampilan Halaman *List* Wisata

Setelah *user* menginputkan suatu data wisata dan sudah diverifikasi oleh admin, maka data tersebut akan masuk ke dalam *database*. Berikut tampilan tabel berdasarkan 4 kategori wisata.

HEADER



[Beranda](#)
[Wisata](#)
[Penginapan](#)
[Oleh-Oleh Khas Kota Medan](#)
[Event](#)
[About](#)
[login](#)

Wisata Religi Terbaru

[Kembali](#)

Show 10 entries

Search :


No	Title	Description	longtitude	latitude	foto
					<div style="background-color: #333; color: white; padding: 5px 10px; display: inline-block;">Lihat</div>

«
1
2
3
4
5
»

footer

Gambar 4.25 Tampilan Halaman *List* Wisata Religi

HEADER



[Beranda](#)
[Wisata](#)
[Penginapan](#)
[Oleh-Oleh Khas Kota Medan](#)
[Event](#)
[About](#)
[login](#)

Wisata Sejarah Terbaru

[Kembali](#)

Show 10 entries

Search :

No	Title	Description	longtitude	latitude	foto
					<div style="background-color: #333; color: white; padding: 5px 10px; display: inline-block;">Lihat</div>

«
1
2
3
4
5
»

footer

Gambar 4.26 Tampilan Halaman *List* Wisata Sejarah

HEADER

[Beranda](#)
[Wisata](#)
[Penginapan](#)
[Oleh-Oleh Khas Kota Medan](#)
[Event](#)
[About](#)
[login](#)

Wisata Kuliner Terbaru

[Kembali](#)

Show 10 entries

Search :

No	Title	Description	longitude	latitude	foto
					<div style="background-color: #333; color: white; padding: 5px 10px; display: inline-block;">Lihat</div>

«
1
2
3
4
5
»

footer

Gambar 4.27 Tampilan Halaman *List* Wisata Kuliner

HEADER

[Beranda](#)
[Wisata](#)
[Penginapan](#)
[Oleh-Oleh Khas Kota Medan](#)
[Event](#)
[About](#)
[login](#)

Wisata Buatan Terbaru

[Kembali](#)

Show 10 entries

Search :

No	Title	Description	longitude	latitude	foto
					<div style="background-color: #333; color: white; padding: 5px 10px; display: inline-block;">Lihat</div>

«
1
2
3
4
5
»

footer

Gambar 4.28 Tampilan Halaman *List* Wisata Buatan

4. Tampilan Halaman Penginapan

Terdapat halaman penginapan yang di dalamnya terdapat 2 kategori penginapan yaitu penginapan umum dan syariah. Dan terdapat *form* untuk mengisi data penginapan. Berikut tampilannya:

Gambar 4.29 Tampilan Halaman Penginapan

5. Tampilan Halaman *List* Penginapan

Setelah *user* menginputkan suatu data penginapan dan sudah diverifikasi oleh admin, maka data tersebut akan masuk ke dalam *database*. Berikut tampilan tabel berdasarkan 2 kategori penginapan.

No	Title	Description	longitude	latitude	foto
					Lihat

Gambar 4.30 Tampilan Halaman *List* Penginapan Umum

HEADER

LOGO Beranda Wisata Penginapan Oleh-Oleh Khas Kota Medan Event About login

Penginapan Syariah Terbaru Kembali

Show 10 entries Search:

No	Title	Description	longitude	latitude	foto
					Lihat

« 1 2 3 4 5 »

footer

Gambar 4.31 Tampilan Halaman *List* Penginapan Syariah

6. Tampilan Halaman Oleh-oleh

Terdapat halaman oleh-oleh khas kota medan dan terdapat *form* untuk *input* data oleh-oleh. Berikut tampilannya:

HEADER

LOGO Beranda Wisata Penginapan Oleh-Oleh Khas Kota Medan Event About login

Oleh-oleh Khas Medan list

- +

Pilih Titik Awal Pilih Titik Akhir kesana

Rekomendasi Tempat

Form

latitude longitude

Title

Description

Choose file Browse

Submit Location

footer

Gambar 4.32 Tampilan Halaman Oleh-oleh

7. Tampilan Halaman *List* Oleh-oleh

Setelah *user* menginputkan suatu data oleh-oleh khas kota medan dan sudah

diverifikasi oleh admin, maka data tersebut akan masuk ke dalam *database*. Berikut tampilan dari tabel oleh-oleh:

Gambar 4.33 Tampilan Halaman *List Oleh-oleh*

8. Tampilan Halaman *Event*

User dapat melihat pada *website* tentang *event* apa saja yang ada di Kota Medan. Berikut tampilannya:

Gambar 4.34 Tampilan Halaman *Event*

9. Tampilan Halaman *List Event*

Setelah admin menginputkan suatu data *event*, data tersebut akan masuk ke dalam *database* dan dapat dilihat oleh *user*. Berikut tampilannya:

Gambar 4.35 Tampilan Halaman *List Event*

10. Tampilan Halaman *About*

Terdapat halaman *about* untuk *user* dalam melihat tentang Dinas Pariwisata Kota Medan. Berikut tampilannya:

Gambar 4.36 Tampilan Halaman *About*

11. Tampilan Halaman *Login Admin*

Terdapat halaman *login* khusus untuk admin dalam mengakses atau mengelola suatu *website* instansi. Berikut tampilan halaman *login*.

HEADER

LOGO Beranda Wisata Penginapan Oleh-Oleh Khas Kota Medan Event About [login](#)

Login

Email

Password

Masuk

footer

Gambar 4.37 Tampilan Halaman *Login* Admin

12. Tampilan Halaman *Register* admin

Tampilan halaman *register* adalah halaman dimana admin harus registrasi terlebih dahulu. Setelah admin mengisi form registrasi, data tersebut akan masuk ke *database* dan admin dapat *login*.

HEADER

LOGO Beranda Wisata Penginapan Oleh-Oleh Khas Kota Medan Event About [login](#)

Register

Name

E-Mail Address

Password

Confirm Password

Register

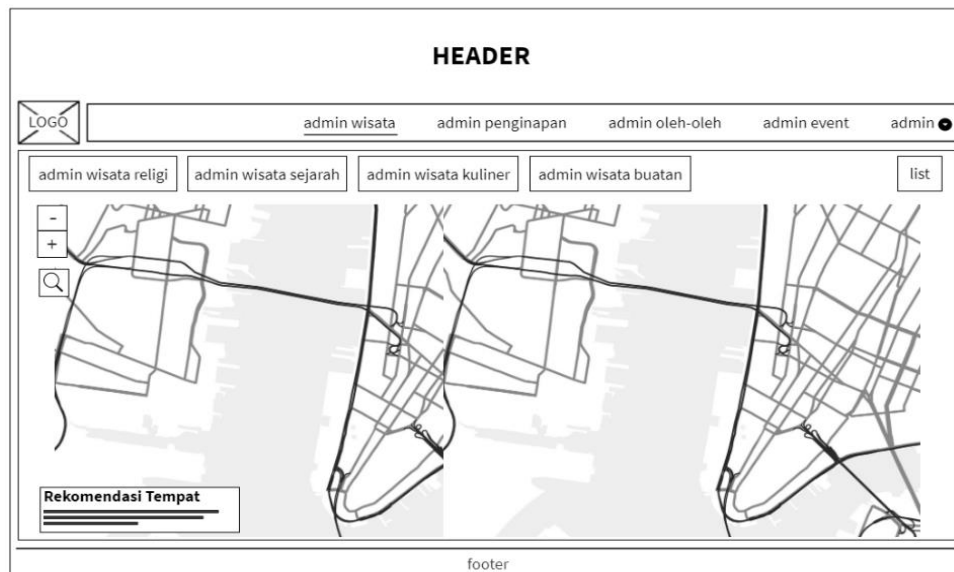
footer

Gambar 4.38 Tampilan Halaman *Register*

13. Tampilan Halaman Admin Wisata

Terdapat halaman admin wisata yang di dalamnya terdapat 4 kategori wisata seperti wisata religi, wisata sejarah, wisata kuliner, wisata buatan. Berikut

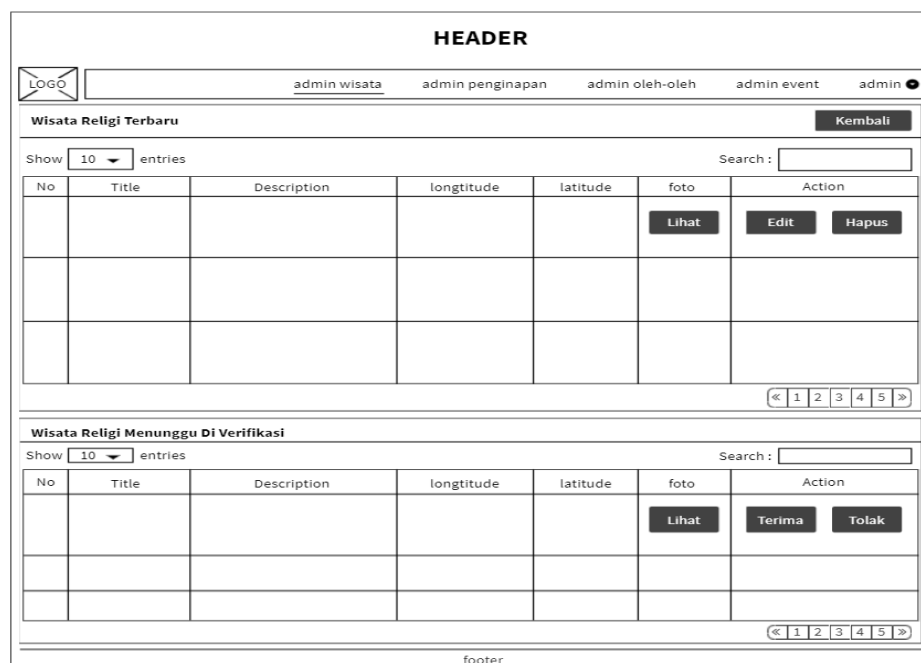
tampilannya:



Gambar 4.39 Tampilan Halaman Admin Wisata


14. Tampilan Halaman Tabel Verifikasi dan Tampil Wisata

Pada tabel verifikasi, admin dapat menyetujui suatu wisata dari 4 kategori wisata yang telah diinputkan oleh *user*. Pada tabel ini admin berhak untuk menerima dan menolak data yang diinputkan *user*. Berikut tampilannya:



Gambar 4.40 Tampilan Halaman Tabel Verifikasi & Tampil Wisata Religi

HEADER



[admin wisata](#) [admin penginapan](#) [admin oleh-oleh](#) [admin event](#) [admin](#)

Wisata Sejarah Terbaru

[Kembali](#)

Show 10 entries

Search :

No	Title	Description	longitude	latitude	foto	Action
					Lihat	Edit Hapus

« 1 2 3 4 5 »

Wisata Sejarah Menunggu Di Verifikasi

Show 10 entries

Search :


No	Title	Description	longitude	latitude	foto	Action
					Lihat	Terima Tolak

« 1 2 3 4 5 »

footer

Gambar 4.41 Tampilan Halaman Tabel Verifikasi & Tampil Wisata Sejarah

HEADER



[admin wisata](#) [admin penginapan](#) [admin oleh-oleh](#) [admin event](#) [admin](#)

Wisata Kuliner Terbaru

[Kembali](#)

Show 10 entries

Search :

No	Title	Description	longitude	latitude	foto	Action
					Lihat	Edit Hapus

« 1 2 3 4 5 »

Wisata Kuliner Menunggu Di Verifikasi

Show 10 entries

Search :

No	Title	Description	longitude	latitude	foto	Action
					Lihat	Terima Tolak

« 1 2 3 4 5 »

footer

Gambar 4.42 Tampilan Halaman Tabel Verifikasi & Tampil Wisata Kuliner

HEADER

admin wisata
admin penginapan
admin oleh-oleh
admin event
admin ●

Wisata Buatan Terbaru
Kembali

Show 10 entries
Search :

No	Title	Description	longitude	latitude	foto	Action
					Lihat	Edit Hapus

« 1 2 3 4 5 »

Wisata Buatan Menunggu Di Verifikasi

Show 10 entries
Search :

No	Title	Description	longitude	latitude	foto	Action
					Lihat	Terima Tolak

« 1 2 3 4 5 »

footer

Gambar 4.43 Tampilan Halaman Tabel Verifikasi & Tampil Wisata Buatan

15. Tampilan Halaman *Update* Wisata

Tampilan halaman *update* wisata adalah suatu *form* yang dapat mengubah atau mengedit data wisata dari 4 kategori wisata. Berikut tampilan *form*nya:

HEADER

admin wisata
admin penginapan
admin oleh-oleh
admin event
admin ●

Update Data Wisata Religi
Kembali

latitude

longitude

Title

Description

Image

Pilih File

Tidak ada file yang dipilih

*kosongkan jika tidak ingin mengubah gambar

Update Location

footer

Gambar 4.44 Tampilan Halaman *Update* Wisata Religi

HEADER

LOGO

admin wisata
admin penginapan
admin oleh-oleh
admin event
admin ●

Update Data Wisata Sejarah
Kembali

latitude

longtitude

Title

Description

Image

Pilih File
Tidak ada file yang dipilih

*kosongkan jika tidak ingin mengubah gambar

Update Location

footer

Gambar 4.45 Tampilan Halaman *Update* Wisata Sejarah

HEADER

LOGO

admin wisata
admin penginapan
admin oleh-oleh
admin event
admin ●

Update Data Wisata Kuliner
Kembali

latitude

longtitude

Title

Description

Image

Pilih File
Tidak ada file yang dipilih

*kosongkan jika tidak ingin mengubah gambar

Update Location

footer

Gambar 4.46 Tampilan Halaman *Update* Wisata Kuliner

HEADER

LOGO

[admin wisata](#)
[admin penginapan](#)
[admin oleh-oleh](#)
[admin event](#)
[admin ●](#)

Update Data Wisata Buatan
Kembali

latitude

longtitude

Title

Description

Image

Pilih File

Tidak ada file yang dipilih

*kosongkan jika tidak ingin mengubah gambar

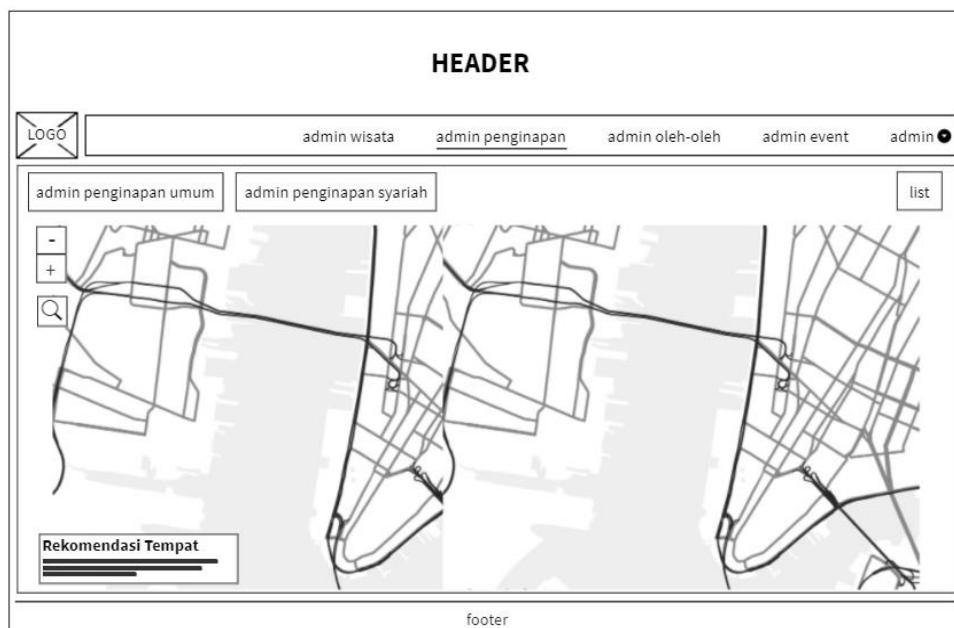
Update Location

footer

Gambar 4.47 Tampilan Halaman *Update* Wisata Buatan

16. Tampilan Halaman Admin Penginapan

Terdapat halaman admin penginapan yang di dalamnya terdapat 2 kategori penginapan. Berikut tampilannya:



Gambar 4.48 Tampilan Halaman Admin Penginapan

17. Tampilan Halaman Tabel Verifikasi dan Tampil Penginapan Admin

Pada tabel verifikasi, admin dapat menyetujui suatu penginapan dari 2 kategori penginapan yang telah diinputkan oleh *user*. Pada tabel ini admin berhak untuk menerima dan menolak data yang diinputkan *user*. Berikut tampilannya:

HEADER

admin wisata
admin penginapan
admin oleh-oleh
admin event
admin ●

Penginapan Umum Terbaru

[Kembali](#)

Show 10 entries

Search :

No	Title	Description	longitude	latitude	foto	Action
					Lihat	Edit Hapus

« 1 2 3 4 5 »

Penginapan Umum Menunggu Di Verifikasi

Show 10 entries

Search :

No	Title	Description	longitude	latitude	foto	Action
					Lihat	Terima Tolak

« 1 2 3 4 5 »

footer

Gambar 4.49 Tampilan Halaman Tabel Verifikasi & Tampil Penginapan Umum

HEADER

admin wisata
admin penginapan
admin oleh-oleh
admin event
admin ●

Penginapan Syariah Terbaru

[Kembali](#)

Show 10 entries

Search :

No	Title	Description	longitude	latitude	foto	Action
					Lihat	Edit Hapus

« 1 2 3 4 5 »

Penginapan Syariah Menunggu Di Verifikasi

Show 10 entries

Search :

No	Title	Description	longitude	latitude	foto	Action
					Lihat	Terima Tolak

« 1 2 3 4 5 »

footer

Gambar 4.50 Tampilan Halaman Tabel Verifikasi & Tampil Penginapan Syariah

18. Tampilan Halaman *Update* Penginapan

Tampilan halaman *update* penginapan adalah suatu *form* yang dapat mengubah atau mengedit data penginapan dari 2 kategori penginapan. Berikut tampilan *form*nya:

HEADER

LOGO admin wisata admin penginapan admin oleh-oleh admin event admin ●

Update Data Penginapan Umum Kembali

latitude longitude

Title

Description

Image Pilih File Tidak ada file yang dipilih

*kosongkan jika tidak ingin mengubah gambar

Update Location

footer

Gambar 4.51 Tampilan Halaman *Update* Penginapan Umum Admin

HEADER

LOGO admin wisata admin penginapan admin oleh-oleh admin event admin ●

Update Data Penginapan Syariah Kembali

latitude longitude

Title

Description

Image Pilih File Tidak ada file yang dipilih

*kosongkan jika tidak ingin mengubah gambar

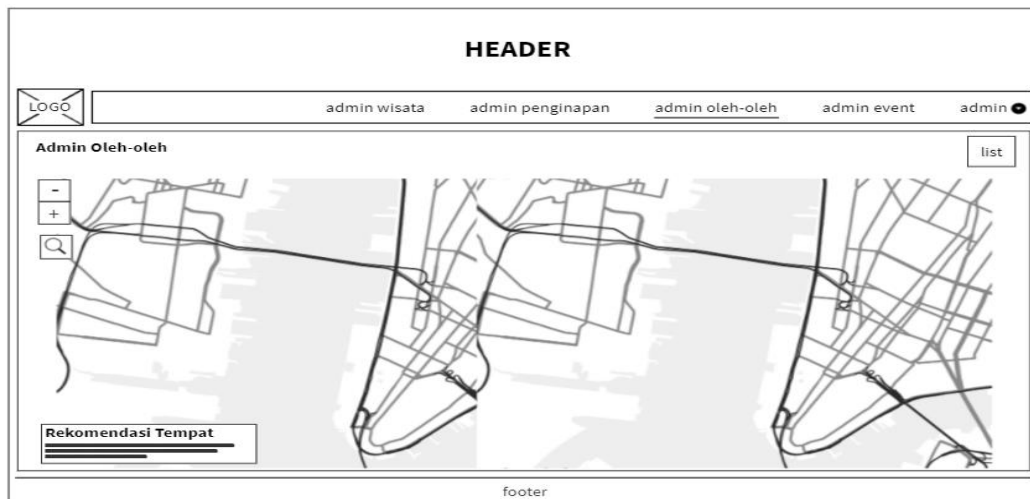
Update Location

footer

Gambar 4.52 Tampilan Halaman *Update* Penginapan Syariah Admin

19. Tampilan Halaman Admin Oleh-oleh

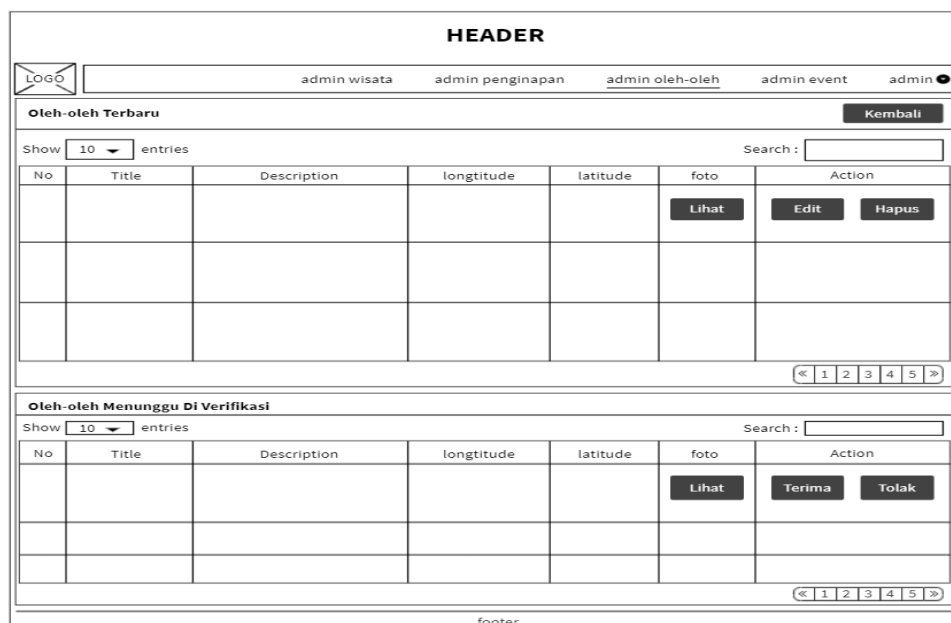
Terdapat halaman admin yang di dalamnya terdapat oleh-oleh khas Kota Medan. Berikut tampilannya:



Gambar 4.53 Tampilan Halaman Admin Oleh-oleh

20. Tampilan Halaman Tabel Verifikasi dan Tampil Oleh-oleh Admin

Pada tabel verifikasi, admin dapat menyetujui oleh-oleh khas Kota Medan yang telah diinputkan oleh *user*. Pada tabel ini admin berhak untuk menerima dan menolak data yang diinputkan *user*. Berikut tampilannya:



Gambar 4.54 Tampilan Halaman Tabel Verifikasi & Tampil Oleh-oleh Admin

21. Tampilan Halaman *Update* Oleh-oleh

Tampilan halaman *update* oleh-oleh adalah suatu *form* yang dapat mengubah atau mengedit data oleh-oleh khas Kota Medan. Berikut tampilan *form*nya:

The screenshot shows the 'Update Data Oleh-oleh' form within an admin dashboard. The dashboard has a header with a logo, navigation links (admin wisata, admin penginapan, admin oleh-oleh, admin event, admin), and a footer. The form itself has a title bar with a 'Kembali' button. It contains input fields for 'latitude' and 'longitude', a 'Title' field, a 'Description' field with a text area, and an 'Image' section with a 'Pilih File' button and a message 'Tidak ada file yang dipilih'. A note below the image section says '*kosongkan jika tidak ingin mengubah gambar'. At the bottom right of the form is an 'Update Location' button.

Gambar 4.55 Tampilan Halaman *Update* Oleh-oleh Admin

22. Tampilan Halaman Admin *Event*

Terdapat halaman admin yang di dalamnya terdapat *form* untuk menginputkan suatu *event*. Berikut tampilannya:

The screenshot shows the 'Admin Event' page within an admin dashboard. The dashboard has a header with a logo, navigation links (admin wisata, admin penginapan, admin oleh-oleh, admin event, admin), and a footer. The page is divided into two main sections: 'Admin Event' on the left and 'Form' on the right. The 'Admin Event' section features a map with a 'list' button and a 'Rekomendasi Tempat' button. The 'Form' section contains input fields for 'latitude' and 'longitude', a 'Title' field, a 'Description' field with a text area, a 'Choose file' button, a 'Browse' button, and a 'Submit Location' button.

Gambar 4.56 Tampilan Halaman Admin *Event*

23. Tampilan Halaman Tabel Tampil *Event* Admin


Setelah admin menginputkan suatu *event*, data tersebut akan masuk ke dalam *database*. Berikut tampilan tabel *event*:

HEADER						
<div>  admin wisata admin penginapan admin oleh-oleh <u>admin event</u> admin ● </div>						
Event Terbaru						Kembali
Show <input type="text" value="10"/> entries						Search: <input type="text"/>
No	Title	Description	longitude	latitude	foto	Action
					Lihat	Edit Hapus
						« 1 2 3 4 5 »
footer						

Gambar 4.57 Tampilan Halaman Tabel Tampil *Event* Admin

24. Tampilan Halaman *Update Event*

Tampilan halaman *update event* adalah suatu *form* yang dapat mengubah atau mengedit data *event* yang ada di Kota Medan. Berikut tampilan *form*nya:

HEADER	
<div>  admin wisata admin penginapan admin oleh-oleh <u>admin event</u> admin ● </div>	
Update Data Event	
Kembali	
latitude	longitude
<input type="text"/>	<input type="text"/>
Title	
<input type="text"/>	
Description	
<input type="text"/>	
Image	
Pilih File	Tidak ada file yang dipilih
*kosongkan jika tidak ingin mengubah gambar	
Update Location	
footer	

Gambar 4.58 Tampilan Halaman *Update Event* Admin

4.3 Implementation

Pada tahap implementasi sistem akan dikembangkan sesuai desain proses, desain *database* dan desain *interface* yang telah ditentukan dan dengan menggunakan bahasa pemrograman *PHP*, *HTML*, dan *MySQL* sebagai *database* serta peta yang digunakan adalah *OSM (Open Street Map)*.

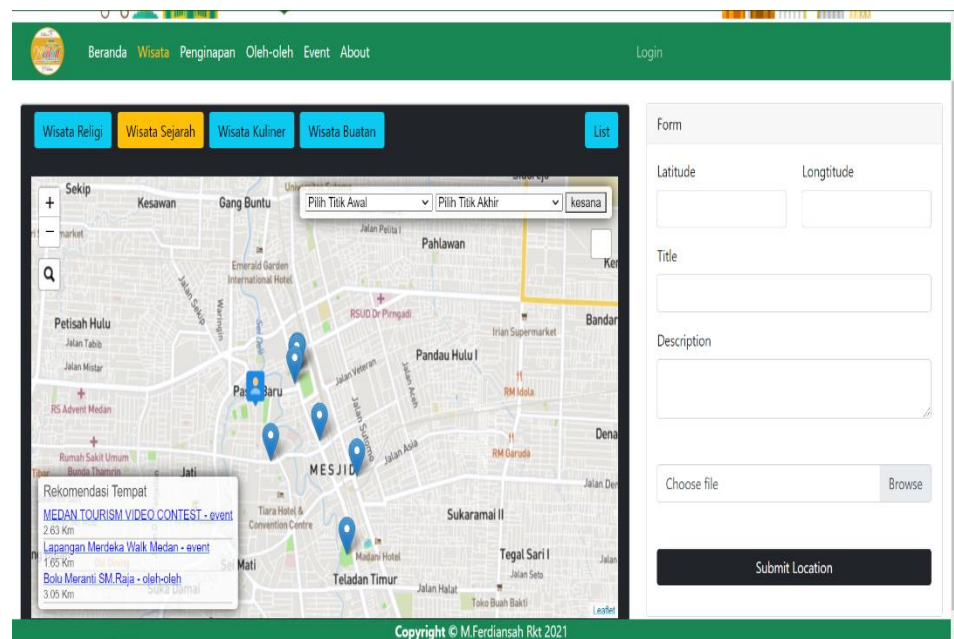
4.4 Verification

Pada tahap ini dilakukan proses pengujian sistem yang dibangun. Adapun pelaksanaan dan hasil dari pengujian sistem adalah sebagai berikut:

Tabel 4.18 Pengujian Metode *Blackbox* Level *User*

No.	Rancangan Input/Output	Hasil Yang Diharapkan	Hasil Aktual
1.	Menjalankan <i>website</i>	Masuk ke halaman beranda	Berhasil
	Tampilan <i>website</i> : 		
2.	Klik→Menu wisata	Menampilkan halaman wisata Dengan 4 kategori wisata dan rekomendasi yang terdekat dari <i>user</i> berada	Berhasil

Tampilan *website*:

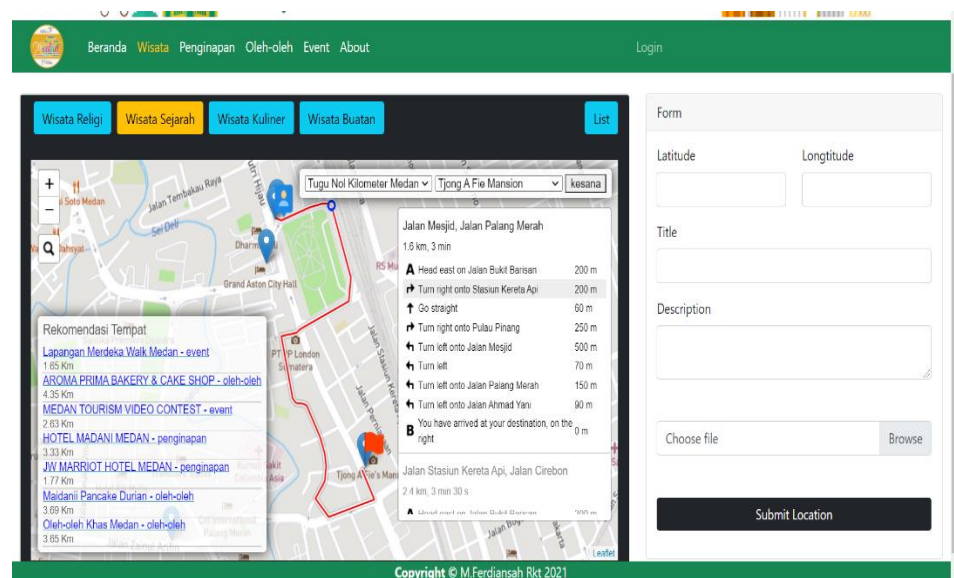


3. Pilih titik awal dan akhir, lalu klik → *button* kesana

Menampilkan *route* titik lokasi wisata sejarah

Berhasil

Tampilan *website*:



4. Klik titik lokasi berada, lalu klik → *button* arahkan kesini

Menampilkan *popup* titik lokasi dan *route* lokasi wisata dari *user* berada

Berhasil

Tampilan *website*:

Copyright © M.Ferdiansah Rkt 2021


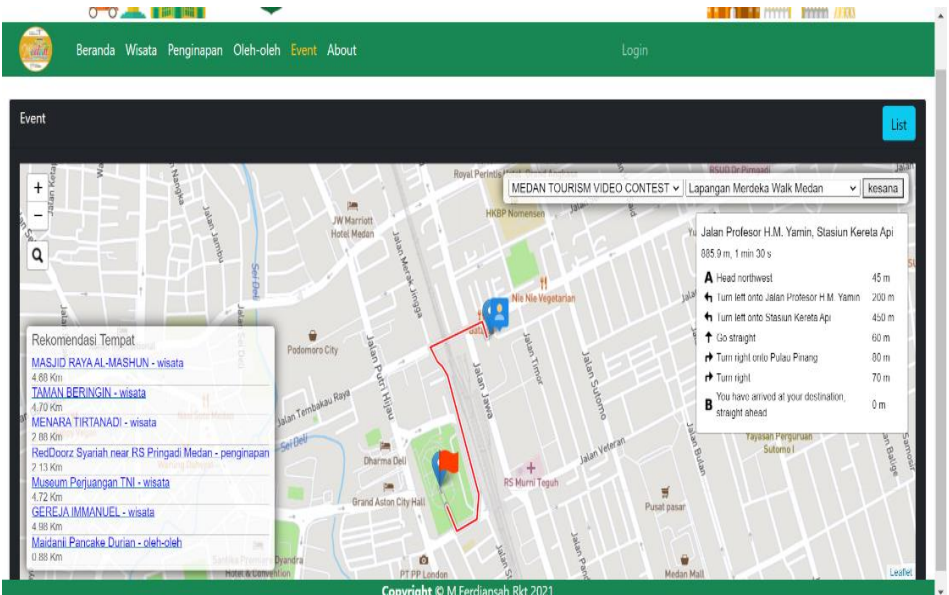
Copyright © M.Ferdiansah Rkt 2021

5.	Klik→ <i>Button list</i>	Menampilkan data wisata sejarah	Berhasil
----	--------------------------	---------------------------------	----------

Tampilan *website*:



Copyright © M.Ferdiansah Rkt 2021

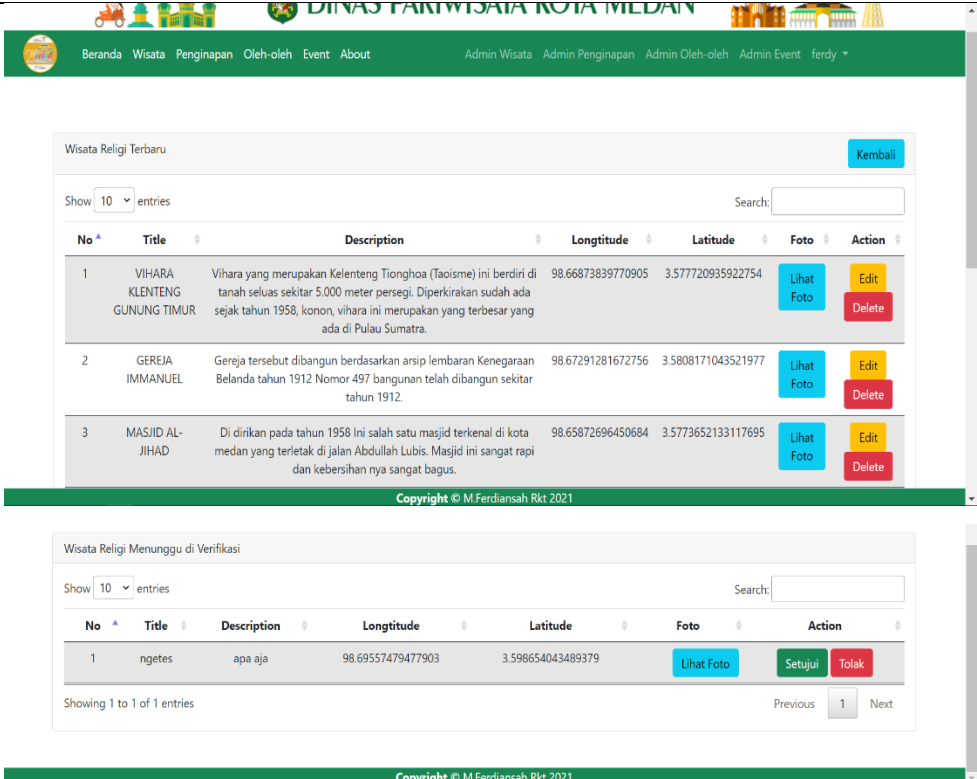
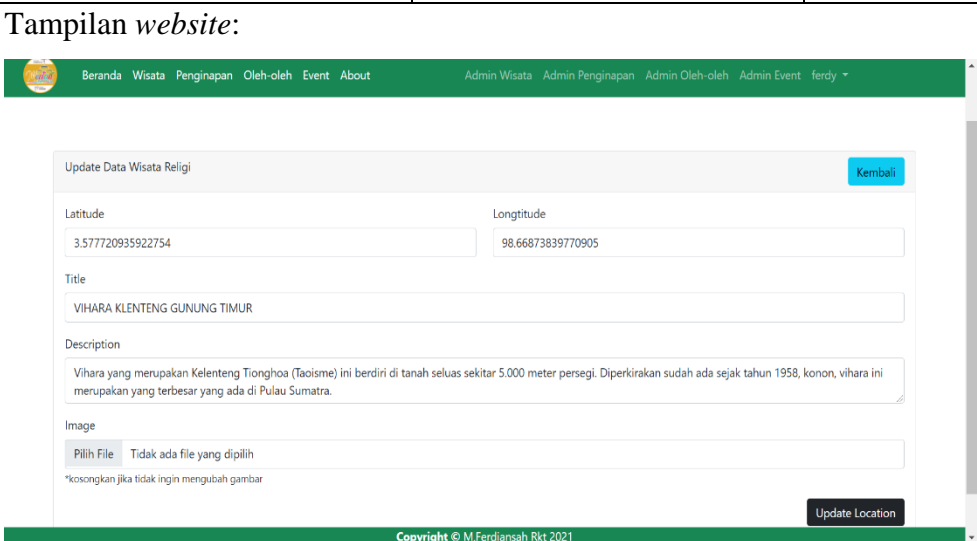
No *	Title	Description	Longitude	Latitude	Foto
1	Tugu Nol Kilometer Medan	Tugu Nol Kilometer Medan merupakan patung air pancur. Selain itu, tugu tersebut dari masa Kolonial sampai saat ini terletak tepat di hadapan pintu masuk Kantor Pos Medan. Berdasarkan data dari Deli Data, pembangunan air pancur ini dibuat tahun 1915.	98.67739043340718	3.591472290508836	Lihat Foto
2	Tjong A Fie Mansion	Rumah Tjong A Fie (Hanzi tradisional: 張耀軒故居; Hanzi sederhana: 张耀轩故居; Pinyin: Zhāng Yàoxuān Gùjū) adalah rumah dua lantai di Jalan Ahmad Yani di Kesawan, Medan, Sumatra Utara, yang dibangun oleh Tjong A Fie (1860–1921), pedagang Hakka yang memiliki banyak tanah perkebunan di Medan.	98.68030416346674	3.585421528910968	Lihat Foto
3	MENARA	Menara Air Tirtanadi (Aksara Jawi: مینارا ایر تیرتاندی) merupakan salah satu ikon kota	98.68515932413514	3.5821939783357966	Lihat Foto

6.	Form user input titik lokasi wisata	Tampilan form user tambah wisata	Berhasil
<p>Tampilan website:</p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid #ccc; padding: 10px; width: 45%;"> <p>Form</p> <p>Latitude <input type="text"/> Longitude <input type="text"/></p> <p>Title <input type="text"/></p> <p>Description <input type="text"/></p> <p>Choose file <input type="button" value="Browse"/></p> <p><input type="button" value="Submit Location"/></p> </div> <div style="border: 1px solid #ccc; padding: 10px; width: 45%;"> <p>Form</p> <p>Latitude <input type="text" value="3.598554043480570"/> Longitude <input type="text" value="98.65557470477003"/></p> <p>Title <input type="text" value="ngarten"/></p> <p>Description <input type="text" value="apa aja"/></p> <p>Choose file <input type="button" value="Browse"/></p>  <p><input type="button" value="Submit Location"/></p> </div> </div>			
7.	Klik→menu <i>event</i> lalu pilih titik lokasi	Menampilkan halaman peta, <i>route</i> , dan rekomendasi terdekat	Berhasil
<p>Tampilan website:</p> 			

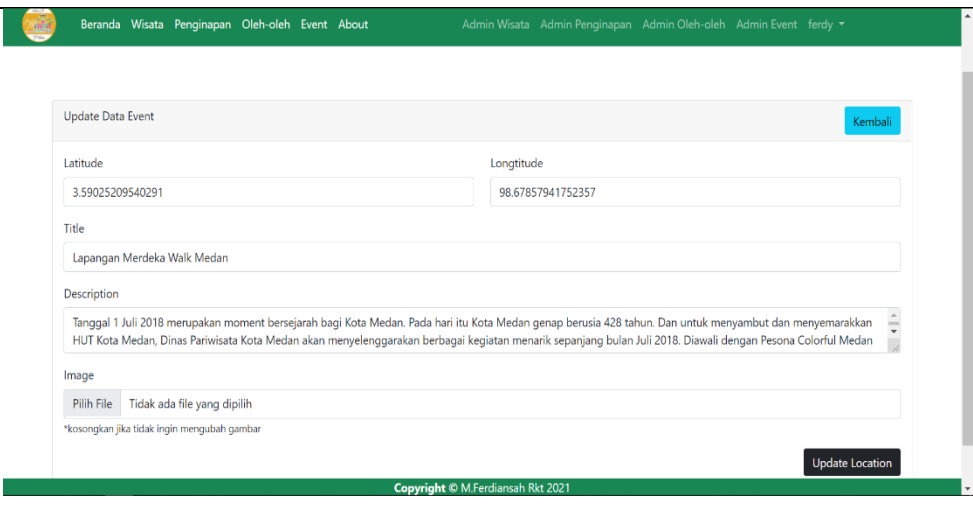
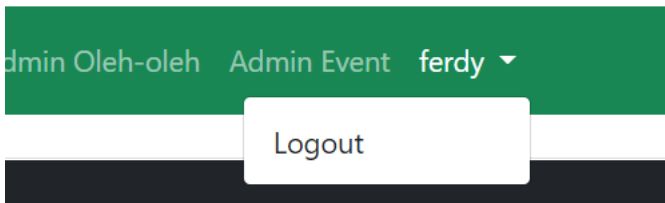
Tabel 4.19 Pengujian Metode *Blackbox* Level Admin

No.	Rancangan Input/Output	Hasil Yang Diharapkan	Hasil Aktual
1.	Menjalankan <i>website</i>	Masuk ke halaman beranda	Berhasil

	<p>Tampilan <i>website</i>:</p> 		
2.	Klik→ <i>Login</i>	Masuk ke halaman <i>login</i> admin	Berhasil
	<p>Tampilan <i>website</i>:</p> 		
3.	Klik→Menu admin wisata	Menampilkan halaman wisata dengan 4 kategori wisata	Berhasil
	Tampilan <i>website</i> :		

			
6.	Klik→ <i>Button</i> edit	Menampilkan <i>form</i> edit wisata admin	Berhasil
	<p>Tampilan <i>website</i>:</p> 		
7.	Klik→Menu admin <i>event</i>	Menampilkan peta <i>event</i> , <i>rute</i> , dan <i>form</i> tambah admin <i>event</i>	Berhasil
	Tampilan <i>website</i> :		

			
8.	Klik→ <i>Button list</i>	Menampilkan data <i>event</i> admin	Berhasil
Tampilan <i>website</i> :			
			
9.	Klik→ <i>Button edit</i>	Menampilkan <i>form</i> edit admin <i>event</i>	Berhasil
Tampilan <i>website</i> :			

			
10.	Klik→ <i>Button logout</i>	Menampilkan <i>button logout</i>	Berhasil
Tampilan <i>website</i> : 			

4.5 Maintenance

Pada tahap ini Sistem Informasi Geografis Pariwisata Kota Medan yang dibangun sudah berhasil untuk digunakan dan dilakukan pemeliharaan sistem termasuk memperbaiki perubahan sistem kembali atau *update* sistem sesuai kebutuhan yang akan mendatang.

Pada pemeliharaan sistem, perlu dilakukan beberapa cara untuk keamanan dari sistem yang telah dibangun. Adapun cara yang dapat digunakan seperti berikut:

1. Melakukan Pemeriksaan Fungsi Pada Sistem

Dalam melakukan pemeriksaan pada sistem, admin perlu memeriksa fungsi pada sistem dalam setiap halaman *website* secara perbulan. Hal ini untuk menghindari *error* yang mungkin akan terjadi pada sistem, sehingga seluruh halaman dapat berjalan dengan baik.

2. Melakukan *Back Up* Data

Pada pemeliharaan *website* perlu dilakukan *back up* data. Hal ini dilakukan untuk menghindari serangan dari berbagai macam virus atau *hacker* yang dapat merusak atau mengganggu kerahasiaan dari data pariwisata kota medan.

3. Melakukan *Update* Tampilan Sistem

Pada pemeliharaan sistem yang telah dibangun, perlu dilakukan perubahan pada tampilan *website*. Hal ini dikarenakan agar pengunjung tidak bosan dalam melakukan pencarian pariwisata di kota medan ketika melihat tampilan yang menarik.

4. Melakukan Penambahan Fitur Terbaru

Dalam pemeliharaan sistem pastinya diperlukan fitur-fitur terbaru yang dikeluarkan dari *platform website* yang digunakan untuk mengoptimalkan tampilan website seperti penambahan fitur rekomendasi mode perjalanan, fitur suara navigasi, dan lain-lain.

5. Melakukan Pemeriksaan *Hosting* dan *Domain Website*

Perlu dilakukan pemeriksaan hosting *website* agar kecepatan tetap stabil untuk digunakan. Hal ini dikarenakan berbagai aktivitas pengunjung pada *website* dapat menyebabkan *CPU usage* dan *loading website* menjadi lambat. Kemudian pada sebuah *domain* diperlukan pengecekan masa aktif *domain*. Hal ini untuk menghindari tanggal kadaluarsa *domain* agar pengunjung dapat mengakses sistem dengan baik tanpa ada gangguan.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan pembahasan dari bab sebelumnya, maka dapat diperoleh kesimpulan, yaitu:

1. Dibangunnya sistem informasi geografis ini menghasilkan suatu lokasi wisata yang berada di Kota Medan, lokasi tempat penginapan, serta lokasi oleh-oleh khas Kota Medan dan *Event* yang diadakan di Kota Medan.
2. Dengan adanya sistem informasi geografis ini memudahkan wisatawan atau masyarakat Kota Medan dalam memilih suatu lokasi yang dituju dengan *route* terpendek.

5.2. Saran

Berdasarkan kesimpulan dan analisa yang telah dilakukan, maka terdapat beberapa saran untuk pengembangan sistem selanjutnya seperti sebagai berikut:

1. Pada sistem yang dibangun saat ini masih menggunakan pemrograman berbasis web. Diharapkan penelitian selanjutnya mengembangkan sistem berbasis android.
2. Untuk peneliti selanjutnya diharapkan dapat menambahkan fitur lokasi-lokasi lain sehingga sistem ini benar-benar memiliki banyak kelebihan.
3. Selanjutnya diharapkan selalu melakukan backup data untuk menghindari terjadinya hilang data yang disebabkan hal yang tak terduga.

DAFTAR PUSTAKA

- Adil, A. (2017). Sistem Informasi Geografis. Penerbit Andi, Yogyakarta.
- Andalia, F., & Setiawan, E. B. (2015). Pengembangan Sistem Informasi Pengolahan Data Pencari Kerja Pada Dinas Sosial Dan Tenaga Kerja Kota Padang. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, 4(2), 93–98.
- Anggraeni, E. Y., & Irviani, R. (2017). Pengantar Sistem Informasi, Yogyakarta: CV. Andi Offset.
- Ardana, D., & Saputra, R. (2016). Penerapan Algoritma *Dijkstra* pada Aplikasi Pencarian Rute Bus Trans Semarang. *Seminar Nasional Ilmu Komputer (SNIK 2016)*, 1, 299–306.
- Hamdi, H., Usman, U., & Samsudin, S. (2018). Perancangan Sistem Informasi Geografis Pemetaan Taman di Kabupaten Indragiri Hilir Berbasis *Web*. *SISTEMASI: Jurnal Sistem Informasi*, 7(2), 87–94.
- Hendra, A. (2012). Pengantar Sistem Informasi. Edisi Pertama, Penerbit Andi, Yogyakarta.
- Hidayatullah, R. (2020). Pemrograman *WEB PHP*. In Yogyakarta: *START UP. START UP*.
- Hutagaol, V., & Sudarsono, B. (2015). Penentuan Potensi Lokasi Atm Bni Menggunakan *Analytical Hierarchy Process (Ahp)* Dan Sistem Informasi Geografis (Studi Kasus: Kecamatan Tembalang). *Jurnal Geodesi Undip*, 4(2), 25–32.
- Hutahaean, J. (2014). Konsep Sistem Informasi, Yogyakarta: CV. Budi Utama.
- Immasari, I. R. (2018). Sistem Informasi Geografis Perdagangan Tanah Abang Dengan Menggunakan *Arcview 3.3*. *Journal of Information System, Informatics and Computing*, 2(2), 4–9.
- Kadir, A. (2002). Pengenalan Sistem Informasi (Vol. Edisi 2). Yogyakarta: Andi Offset.

- Kadir, A. (2014). *Pengenalan Sistem Informasi Edisi Revisi*. Yogyakarta: Andi Offset.
- Kirom, M. (2014). Sistem informasi geografis Pemetaan suara pemelukada berbasis open source di kabupaten jombang. *Edutic-Scientific Journal of Informatics Education*, 1(1).
- Kusuma, R. C. S. D. (2016). Modul Manajemen Event. In Modul Manajemen Event. Yogyakarta.
- Lubis, K. H. (2018). Roman Medan. In Jakarta: Gramedia Pustaka Medan. Gramedia Pustaka Medan.
- Nirwansyah, A. W. (2017). *Dasar Sistem Informasi Geografi dan Aplikasinya Menggunakan ARCGIS 9.3*. Yogyakarta: Deepublish.
- Noor, A. (2013). Manajemen Event. In Bandung: Alfabeta. Alfabeta.
<https://opac.perpusnas.go.id/DetailOpac.aspx?id=911967>
- Nugroho, A. S. (2017). *Analisis dan Perancangan Sistem Informasi*. Yogyakarta: Trans Tekno.
- Nurwanto. (2020). Evaluasi Dampak Pembangunan Pariwisata Menggunakan Konsep *Community Based Tourism (CBT)* di Kawasan Wisata Tebing Breksi. *Jurnal Kepariwisata Indonesia (JKI)*, 12(1), 1411–9862.
- Oktarian, S., & Usman, U. (2016). Sistem Informasi Geografis Pemetaan Fasilitas Kota Tembilahan. *Sistemasi: Jurnal Sistem Informasi*, 5(1), 1–9.
- Rachman, S. N. M., & Maulana, N. (2012). *Sistem informasi Geografis Parawisata Kota Yogyakarta Berbasis Mobile Android 2.2*. Sekolah Tinggi Ilmu Manajemen Informatika Dan Komputer Amikom Yogyakarta.
- Rahma, A. A. (2020). Potensi Sumber Daya Alam dalam Mengembangkan Sektor Pariwisata di Indonesia. *Jurnal Nasional Pariwisata*, 12(1), 1–8.
- Rahmi, M. A. (2018). Analisis City Branding Kota Medan Terhadap Pertumbuhan Pariwisata.

- Ramadhani, H. A., & Awaluddin, M. (2016). Aplikasi *Webgis* untuk Informasi Persebaran Sekolah Menengah Atas dan Madrasah Aliyah di Kabupaten Kudus Menggunakan *Here Map Api*. *Jurnal Geodesi Undip*, 5(1), 164–173.
- Rezky Nur Awalia, N. H. . K. A. (2017). Kajian Karakter Pembentuk Lanskap Budaya Masyarakat Adat Kajang Di Sulawesi Selatan. *Jurnal Lanskap Indonesia*, 9(2), 91–100.
- Riyanto, Ekaputra, P., & Indelarko, H. (2019). Pengembangan Aplikasi Sistem Informasi Geografis Berbasis *Desktop* Dan *Web*. Yogyakarta: Gava Media.
- Samsudin, S., Irawan, M. D., & Harahap, A. H. (2019). *Mobile App Education* Gangguan Pencernaan Manusia Berbasis Multimedia Menggunakan *Adobe Animate CC*. *JurTI (Jurnal Teknologi Informasi)*, 3(2), 141–148.
- Sasmito, G. W. (2017). Penerapan Metode *Waterfall* Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Informatika: Jurnal Pengembangan IT*, 2(1), 6–12.
- Setiawan, E. B. (2020). Sistem Informasi Geografis Berbasis *Web*, Menggunakan *Google Maps* dan *Mapbox API*. Bandung, Informatika Bandung, 387.
- Sidik, B. (2019). HTML5 Dasar-dasar Untuk Pengembangan Aplikasi Berbasis *Web*. In Bandung: Informatika Bandung. Informatika Bandung.
- Suendri, S. (2019). Implementasi Diagram *UML (Unified Modelling Language)* Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan *Database Oracle* (Studi Kasus: UIN Sumatera Utara Medan). *Algoritma: Jurnal Ilmu Komputer Dan Informatika*, 2(2), 1.
- Susepti, A., Hamid, D., & Kusumawati, A. (2017). Pengaruh Kualitas Pelayanan Terhadap Kepuasan Dan Loyalitas Tamu Hotel (Studi tentang Persepsi Tamu Hotel Mahkota Plengkung Kabupaten Banyuwangi). *Jurnal Administrasi Bisnis*, 50(5), 27–36.
- Syam, F. H., Arifin, N., & Arifin, H. S. (2019). Kajian Potensi Lanskap Kota Medan

untuk Pengembangan Wisata Sejarah. *Jurnal Lanskap Indonesia*, 11(2).

Tanjaya, E. J., Rostianingsih, S., & Handojo, A. (2016). Pemetaan Surabaya Heritage Dengan *Geographic Information System*. *Jurnal Infra*, 4(2), 149–152.

Triase, T., & Aprilia, R. (2020). Implementasi Penyaluran Paket Online Shop Menggunakan Algoritma *FIFO* dan *Dijkstra*. *Query: Journal of Information Systems*, 4(1).

Yudhanto, Y., & Prasetyo, H. A. (2019). Mudah menguasai framework laravel. *Elex Media Komputindo*.

Zufria, I. (2013). Pemodelan Berbasis *UML (Unified Modeling Language)* dengan Strategi Teknik Orientasi Objek *User Centered Design (UCD)* dalam Sistem Administrasi Pendidikan. *Research Gate*.
<https://www.researchgate.net/publication/306037464>

LAMPIRAN I

Surat Izin Riset Balitbang



PEMERINTAH KOTA MEDAN BADAN PENELITIAN DAN PENGEMBANGAN

Jalan Kapten Maulana Lubis Nomor 2 Medan Kode Pos 20112
Telepon. (061) 4555693 Faks. (061) 4555693
E-mail : balitbangmedan@yahoo.co.id. Website : balitbang.pernkoredan.go.id

SURAT REKOMENDASI RISET

NOMOR : 070/2700/Balitbang/2021

Berdasarkan Surat Keputusan Walikota Medan Nomor: 57 Tahun 2001, Tanggal 13 November 2001 dan Peraturan Walikota Medan Nomor: 55 Tahun 2010, tanggal 24 November 2010 tentang Tugas Pokok dan Fungsi Badan Penelitian dan Pengembangan Kota Medan dan setelah membaca/memperhatikan surat dari: Wakil Dekan Bidang Akademik Dan Kelembagaan Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara Medan. Nomor: B.539/ST.I/ST.V.2/TL.00/7/2021. Tanggal: 02 Juli 2021. Hal: Izin Riset.

Badan Penelitian dan Pengembangan Kota Medan dengan ini memberikan Surat Rekomendasi Riset Kepada :

Nama : M. Ferdiansah Rkt.
NIM : 0702171050.
Prodi : Sistem Informasi.
Lokasi : Dinas Pariwisata Kota Medan.
Judul : " Sistem Informasi Geografis Pariwisata Kota Medan Menggunakan Metode Algoritma Dijkstra ".
Lamanya : 8 Juli s.d 8 Agustus 2021.
Penanggung Jawab : Wakil Dekan Bidang Akademik Dan Kelembagaan Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.

Dengan ketentuan sebagai berikut :

1. Sebelum melakukan Penelitian terlebih dahulu harus melapor kepada pimpinan Organisasi Perangkat Daerah lokasi Yang ditetapkan.
 2. Mematuhi peraturan dan ketentuan yang berlaku di lokasi Penelitian.
 3. Tidak dibenarkan melakukan Penelitian atau aktivitas lain di luar lokasi yang telah direkomendasikan.
 4. Hasil penelitian diserahkan kepada Kepala Balitbang Kota Medan selambat lambatnya 2 (dua) bulan setelah penelitian dalam bentuk soft copy atau melalui Email (balitbangmedan@yahoo.co.id).
 5. Surat rekomendasi penelitian dinyatakan batal apabila pemegang surat rekomendasi tidak mengindahkan ketentuan atau peraturan yang berlaku pada Pemerintah Kota Medan.
 6. Surat rekomendasi penelitian ini berlaku sejak tanggal dikeluarkan.
- Demikian Surat ini diperbuat untuk dapat dipergunakan sebagaimana mestinya.

Dikeluarkan di : Medan.
Pada Tanggal : 8 Juli 2021

Sebelum Kepala Balitbang Kota Medan
Sekretaris,

Dra. SITI MAHRANI HASIBUAN
PEMBAKTA TK. I
NIP. 19661208 198603 2 002

Tembusan :

1. Walikota Medan (sebagai Laporan).
2. Kepala Dinas Pariwisata Kota Medan.
3. Wakil Dekan Bidang Akademik Dan Kelembagaan Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
4. Pertinggal.

LAMPIRAN II

Surat Izin Riset Dinas Pariwisata Kota Medan



PEMERINTAH KOTA MEDAN DINAS PARIWISATA

Jl. Prof. H.M. Yamin, SH. No. 40, Medan 20231

Telp. 061-4525248 Fax. 061-4557417

website : www.medantourism.pemkomedan.go.id email : pariwisatamedan@gmail.com

Nomor : 070/2317
Sifat :
Lampiran :
Perihal : Riset di Dinas Pariwisata
Kota Medan

Medan, 10 Agustus 2021

Kepada Yth,
Wakil Dekan Bid. Akademik & Kelembagaan
Fak. Sains dan Teknologi UINSU
di-
Medan

Berdasarkan surat Kepala Badan Penelitian dan Pengembangan Kota Medan Nomor : 070/2700/Balitbang/2021 tanggal 08 Juli 2021 perihal Rekomendasi Riset di Dinas Pariwisata Kota Medan.

Berkenaan dengan hal tersebut dengan ini disampaikan bahwa :

Nama : M. Ferdiansah RKT
NIM : 0702171050
Program Studi : Sistem Informasi
Judul : "Sistem Informasi Geografis Pariwisata Kota Medan
Menggunakan Metode Algoritma Dijkstra"
Lamanya : 8 Juli s.d 8 Agustus 2021

Telah menyelesaikan penelitian dan pengambilan data di Dinas Pariwisata Kota Medan.

Data yang diambil dari Dinas Pariwisata Kota Medan hendaknya hanya digunakan untuk keperluan penelitian dan penyelesaian skripsi yang bersangkutan dan tidak digunakan untuk kepentingan lainnya.

Demikian hal ini disampaikan untuk dimaklumi.

KEPALA DINAS PARIWISATA
KOTA MEDAN



Drs. AGUS SURIYONO
PEMBINA TK. I

NIP. 19650830 198712 1 003

Tembusan :

1. Bapak Wali Kota Medan (sebagai laporan)
2. Kepala Balitbang Kota Medan
3. Yang Bersangkutan
4. Peringgal

Visi : Terwujudnya Masyarakat Kota Medan Yang Berkah, Maju Dan Kondusif

Please follow our social media account



Dinas Pariwisata Kota Medan



DisparMedan



pariwisatamedan



Medan Tourism



Dinas Pariwisata Kota Medan



Medan Creative Market



LAMPIRAN III

Formulir Hasil Wawancara

LEMBAR VALIDASI WAWANCARA

Judul Skripsi : Sistem Informasi Geografis Pariwisata Kota Medan Menggunakan Metode Algoritma *Dijkstra*
Nama Penulis : M. Ferdiansah Rkt
NIM : 0702171050
Nama Narasumber : Susanto, S.Kom
Jabatan : Pengelola Teknologi Informasi Sub Bagian Umum
Tanggal Wawancara : April 2021

No	Pertanyaan	Jawaban
1.	Apakah dinas pariwisata saat ini sudah memiliki sistem informasi geografis terkait tentang pariwisata?	Sudah ada
2.	Bagaimana alur berjalannya sistem pada saat ini?	Sistem saat ini pemetaan hanya beberapa titik dan hanya memiliki beberapa informasi terkait pariwisata
3.	Apakah terdapat kendala pada sistem saat ini?	Kendala saat ini tidak memberikan <i>user</i> <i>rule</i> jarak terdekat untuk menuju wisata. Tidak memberikan informasi rekomendasi terkait pariwisata Kota Medan kepada wisatawan.
4.	Apakah ada harapan untuk perbaikan sistem?	Harapan selanjutnya sistem memiliki <i>rule</i> jarak terdekat untuk <i>user</i> menuju wisata yang ada di Kota Medan dan memberikan rekomendasi terkait pariwisata yang ada di Kota Medan.

Medan, Agustus 2021
Pengelola Teknologi Informasi
Sub Bagian Umum
Dinas Pariwisata Kota Medan


Susanto, S.Kom

NIP.19810416 201101 1 009

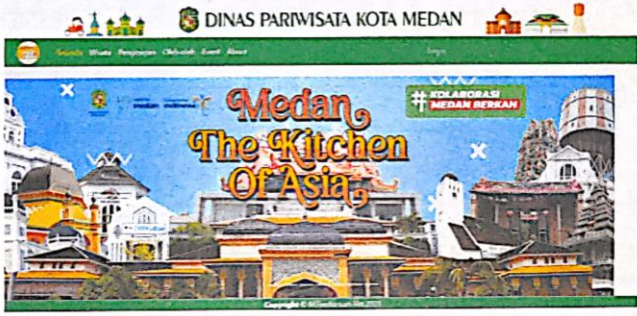
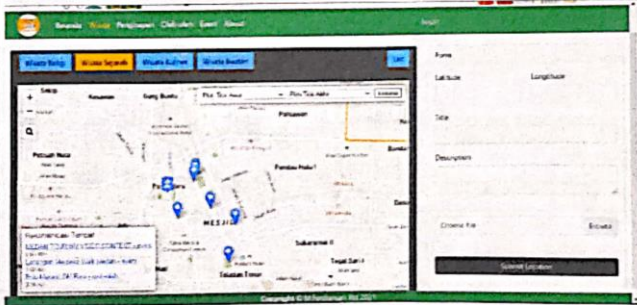
LAMPIRAN IV

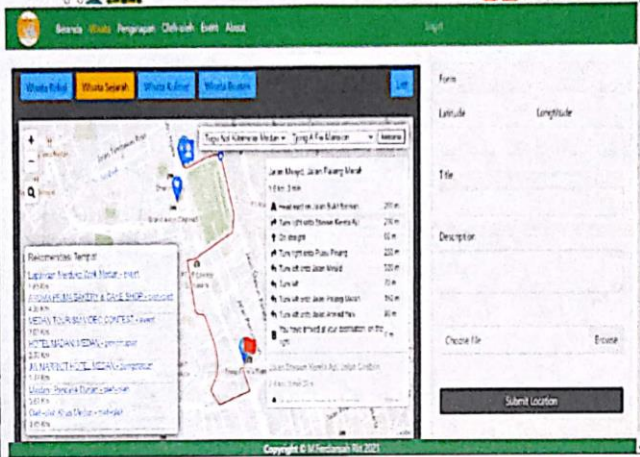
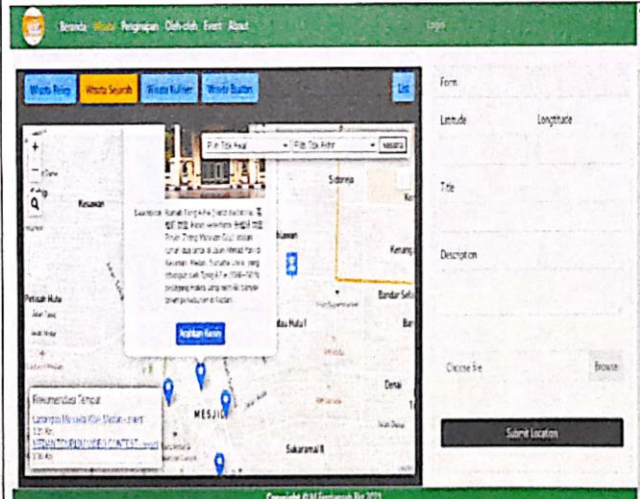
Formulir *Black Box Testing*

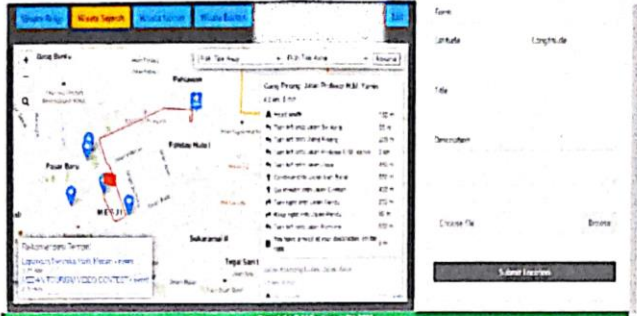


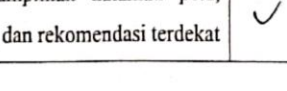
FORMULIR PENGUJIAN BLACKBOX

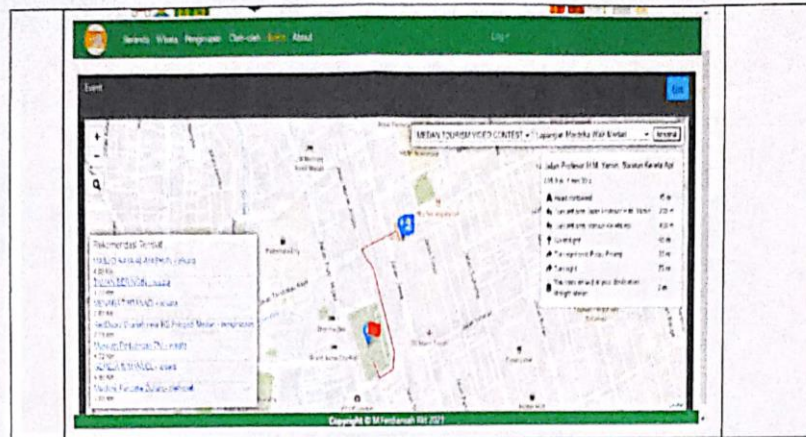
Tanggal Pengujian : 09/ Agustus/ 2021
 Nama Sistem : Sistem Informasi Geografis Pariwisata Kota Medan Menggunakan Metode Algoritma Dijkstra
 Penguji : Susanto, S.Kom

1. Tabel Pengujian Metode *Blackbox* Level User

No.	Rancangan Input/Output	Hasil Yang Diharapkan	Hasil Aktual	
			Berhasil	Tidak Berhasil
1.	Menjalankan <i>website</i>	Masuk ke halaman beranda	✓	
	Tampilan <i>website</i> : 			
2.	Klik→Menu wisata	Menampilkan halaman wisata Dengan 4 kategori wisata dan rekomendasi yang terdekat dari <i>user</i> berada	✓	
	Tampilan <i>website</i> : 			


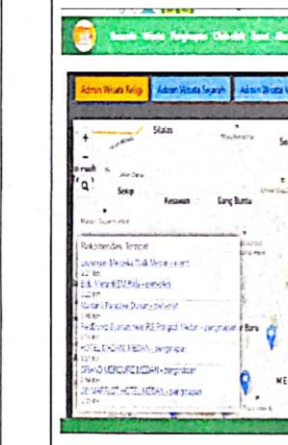
3.	Pilih titik awal dan akhir, lalu klik→button kesana	Menampilkan <i>route</i> titik lokasi wisata sejarah	✓
Tampilan website:			
			
4.	Klik titik lokasi berada, lalu klik→button arahkan kesini	Menampilkan <i>popup</i> titik lokasi dan <i>route</i> lokasi wisata dari user berada	✓
Tampilan website:			
			

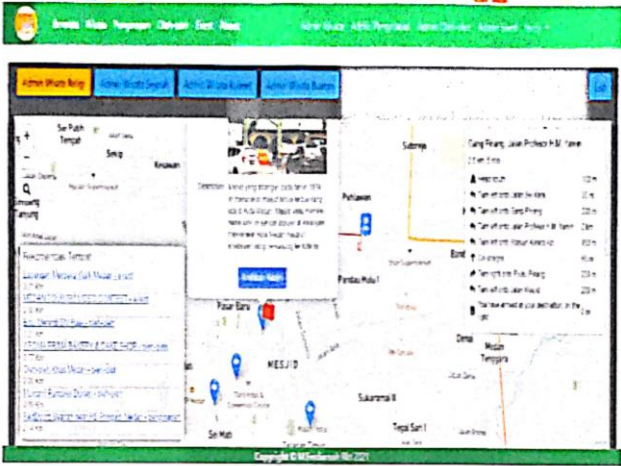
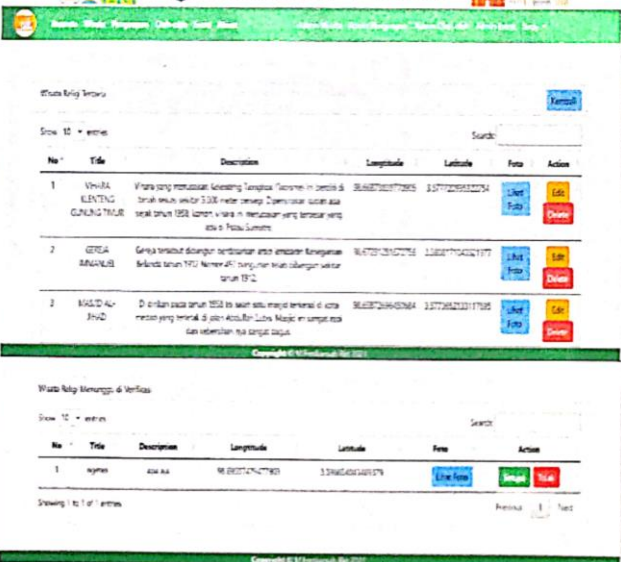
	
<p>5. Klik→<i>Button list</i></p> <p>Menampilkan data wisata sejarah</p> <p>Tampilan website:</p>	
<p>6. <i>Form user input</i> titik lokasi wisata</p> <p>Tampilan form user tambah wisata</p> <p>Tampilan website:</p>	
<p>7. Klik→<i>menu event</i> lalu pilih titik lokasi</p> <p>Menampilkan halaman peta, <i>route</i>, dan rekomendasi terdekat</p> <p>Tampilan website:</p>	





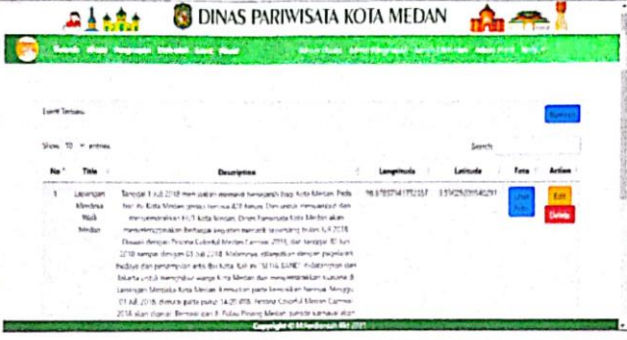


2. Tabel Pengujian Metode *Blackbox* Level Admin

No.	Rancangan Input/Output	Hasil Yang Diharapkan	Hasil Aktual	
			Berhasil	Tidak Berhasil
1.	Menjalankan <i>website</i>	Masuk ke halaman beranda	✓	
	Tampilan <i>website</i> : 			
2.	Klik→ <i>Login</i>	Masuk ke halaman <i>login</i> admin	✓	
	Tampilan <i>website</i> :			

		
<p>3. Klik→Menu admin wisata</p>	<p>Menampilkan halaman wisata dengan 4 kategori wisata</p>	<p>✓</p>
<p>Tampilan website:</p>		
	<p>Menampilkan <i>popup</i> titik lokasi dan <i>route</i> lokasi wisata dari user berada</p>	<p>✓</p>
<p>Tampilan website:</p>		

		
<p>5. Klik→<i>Button list</i></p> <p>Menampilkan data wisata sejarah dan data wisata yang harus diverifikasi</p> <p>✓</p> <p>Tampilan website:</p> 	<p>6. Klik→<i>Button edit</i></p> <p>Menampilkan <i>form</i> edit wisata admin</p> <p>✓</p>	<p>Tampilan website:</p>

			
7.	Klik→Menu admin event	Menampilkan peta event, route, dan form tambah admin event	✓
	Tampilan website: 		
8.	Klik→Button list	Menampilkan data event admin	✓
	Tampilan website:		

			
9.	Klik→ <i>Button edit</i>	Menampilkan <i>form edit</i> admin event	✓
	Tampilan website: 		
10.	Klik→ <i>Button logout</i>	Menampilkan <i>button logout</i>	✓
	Tampilan website: 		

Medan, 09 Agustus 2021
 Pengelola Teknologi Informasi
 Sub Bagian Umum
 Dinas Pariwisata Kota Medan



Susanto, S.Kom
 NIP.19810416 201101 1 009

LAMPIRAN V

Source Code

Layout/app.blade.php

```
<!doctype html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>{{ config('app.name', 'Laravel') }}</title>

    <!-- Scripts -->
    <script src="{{ asset('js/app.js') }}" defer></script>

    <!-- Fonts -->
    <link rel="dns-prefetch" href="//fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">

    <!-- Styles -->
    <link href="{{ asset('css/app.css') }}" rel="stylesheet">
    <link href="{{ asset('css/style.css') }}" rel="stylesheet">

    <!-- Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

    <!-- FontAwesome -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/awesome/5.15.3/css/all.min.css" />

    <!-- Leaflet -->
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />

    <link rel="stylesheet" type="text/css" href="{{ asset('js/leaflet-search/dist/leaflet-search.min.css') }}">
    <link rel="stylesheet" href="{{ asset('js/leaflet-routing-machine/dist/leaflet-routing-machine.css') }}" />

    <!-- DataTable -->
    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/1.10.25/css/jquery.dataTables.css">

    @livewireStyles

    <script src="https://unpkg.com/leaflet@1.3.4/dist/leaflet.js"></script>
    <script src="{{ asset('js/leaflet-routing-machine.js') }}"></script>
```

```

machine/dist/leaflet-routing-
machine.js')?>"</script>
    <script src="<?=asset('js/l
eaflet-search/dist/leaflet-
search.src.js')?>"></script>

```

```

</head>
<body>
    <div id="app">
        <div class="d-
flex justify-content-center">
            
        </div>
        <nav class="navbar navb
ar-expand-lg navbar-dark bg-
success py-0">
            <div class="contain
er-fluid">
                <a class="navba
r-brand" href="#">
                    
                </a>
                <button class="
navbar-
toggler" type="button" data-bs-
toggle="collapse" data-bs-
target="#navbarSupportedContent
" aria-
controls="navbarSupportedConten
t" aria-expanded="false" aria-
label="Toggle navigation">
                    <span class
="navbar-toggler-icon"></span>
                </button>
                <div class="con
tainer">
                    <div class=
"collapse navbar-
collapse" id="navbarSupportedCo
ntent">

```

```

            <ul cla
ss="navbar-nav me-auto mb-2 mb-
lg-0">
                <li
class="nav-item">
                    <a class="nav-
link active {{ Request::path()
=== '/' ? 'text-
warning ' : '' }}" aria-
current="page" href="{{ url('/')
 }}">Beranda</a>
                </li>
                <li
class="nav-item">
                    <a class="nav-
link active {{ Request::path()
=== 'wisata-
religi' || Request::path() ===
'wisata-
sejarah' || Request::path() ===
'wisata-
kuliner' || Request::path() ===
'wisata-buatan' ? 'text-
warning ' : '' }}" aria-
current="page" href="{{ url('/w
isata-religi') }}">Wisata</a>
                </li>
                <li
class="nav-item">
                    <a class="nav-
link active {{ Request::path()
=== 'penginapan-
umum' || Request::path() === 'p
enginapan-syariah' ? 'text-
warning ' : '' }}" aria-
current="page" href="{{ url('/p
enginapan-
umum') }}">Penginapan</a>
                </li>
            </ul>

```

```

</li>
class="nav-item">

<a class="nav-
link active {{ Request::path()
=== 'tempat-oleh-oleh' ? 'text-
warning ' : '' }}" aria-
current="page" href="{{ url('/t
empat-oleh-oleh') }}">Oleh-
oleh</a>
</li>
i>

<li class="nav-item">

<a class="nav-
link active {{ Request::path()
=== 'event' ? 'text-
warning ' : '' }}" aria-
current="page" href="{{ url('/e
vent') }}">Event</a>
</li>
i>

<li class="nav-item">

<a class="nav-
link active {{ Request::path()
=== 'about' ? 'text-
warning ' : '' }}" aria-
current="page" href="{{ url('/a
bout') }}">About</a>
</li>
i>
</ul>
<ul cla
ss="navbar-nav me-auto mb-2 mb-
lg-0">
@gu
est

```

```
<li class="nav-item">
```

```
<a class="nav-
```

```
link" href="{{ route('login') }}
}">{{ __('Login') }}</a>
```

```
</li>
```

```
se
```

```
<li class="nav-item">
```

```
<a class="nav-
link" href="{{ url('admin-
wisata-
religi') }}">Admin Wisata</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-
link" href="{{ url('admin-
penginapan-
umum') }}">Admin Penginapan</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-
link" href="{{ url('admin-oleh-
oleh') }}">Admin Oleh-oleh</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-
link" href="{{ url('admin-
event') }}">Admin Event</a>
```

```
</li>
```

```
<li class="nav-item dropdown">
```

```
<a id="navbarDropdown" cla
ss="nav-link dropdown-
```

```

toggle" href="#" role="button"
data-toggle="dropdown" aria-
haspopup="true" aria-
expanded="false" v-pre>

    {{ Auth::user()-
>name }} <span class="caret"></
span>

</a>

<div class="dropdown-
menu dropdown-menu-right" aria-
labelledby="navbarDropdown">

    <a class="dropdown-
item" href="{{ route('logout')
}}"

        onclick="event.prevent
Default();

        document.
getElementById('logout-
form').submit();">

        {{ __('Logout') }}

    </a>

    <form id="logout-
form" action="{{ route('logout'
) }}" method="POST" style="disp
lay: none;">

        @csrf

    </form>

</div>

</li>

```

```

@en
dguest

</ul>
</div>
</div>
</div>
</nav>

<main class="py-4">
    @yield('content')
    {{isset($slot) ? $s
lot : null}}
</main>
<footer class="text-
center fixed-bottom bg-
success text-white mt-2">
    <strong>Copyright &
#169</strong> M.Ferdiansah Rkt
<script>document.write(new Date
()).getFullYear())</script>
</footer>
</div>

<!-- bootstrap -->
<script src="https://cdn.js
delivr.net/npm/bootstrap@5.0.2/
dist/js/bootstrap.bundle.min.js
"></script>
@livewireScripts
@stack('scripts')
<script src="https://code.j
query.com/jquery-
3.6.0.min.js"></script>
<script src="https://cdn.da
tatables.net/1.10.25/js/jquery.
dataTables.min.js"></script>
<script>
    var jq14 = jQuery.noCon
flict(true);
    (function ($) {
        $(document).ready(f
unction () {
            $('table').Data
Table();
        });
    });

```



```
}(jq14));
```

```
</script>
```

```
</body>
```

```
</html>
```

Beranda/Index.blade.php

```
@extends('layouts.app')
```

```
@section('judul-
```

```
halaman', 'Beranda')
```

```
@section('content')
```

```
<div class="bd-example d-  
flex justify-content-center">
```

```
<div id="carouselExampleCap  
tions" class="carousel slide" d  
ata-ride="carousel">
```

```
<ol class="carousel-  
indicators">
```

```
<li data-  
target="#carouselExampleCaption  
s" data-slide-  
to="0" class="active"></li>
```

```
<li data-  
target="#carouselExampleCaption  
s" data-slide-to="1"></li>
```

```
<li data-  
target="#carouselExampleCaption  
s" data-slide-to="2"></li>
```

```
<li data-  
target="#carouselExampleCaption  
s" data-slide-to="3"></li>
```

```
<li data-  
target="#carouselExampleCaption  
s" data-slide-to="4"></li>
```

```
<li data-  
target="#carouselExampleCaption  
s" data-slide-to="5"></li>
```

```
<li data-  
target="#carouselExampleCaption  
s" data-slide-to="6"></li>
```

```
</ol>
```

```
<div class="carousel-  
inner">
```

```
<div class="carousel-  
item active">
```

```

```

```
</div>
```

```
<div class="carousel-  
item">
```

```

```

```
</div>
```

```
<div class="carousel-  
item">
```

```

```

```
</div>
```

```
<div class="carousel-  
item">
```

```

```

```
</div>
```

```
<div class="carousel-  
item">
```

```

```

```
</div>
```

```
<div class="carousel-  
item">
```

```

```

```
</div>
```

```
<div class="carousel-  
item">
```

```

```

```
</div>
```

```
</div>
```

```
<a class="carousel-  
control-  
prev" href="#carouselExampleCap
```



```

tions" role="button" data-
slide="prev">
    <span class="carousel-
control-prev-icon" aria-
hidden="true"></span>
    <span class="sr-
only">Previous</span>
</a>
<a class="carousel-
control-
next" href="#carouselExampleCap
tions" role="button" data-
slide="next">
    <span class="carousel-
control-next-icon" aria-
hidden="true"></span>
    <span class="sr-
only">Next</span>
</a>
</div>
</div>

```

@endsection

Wisata sejarah/index.blade.php

@extends('layouts.app')

@section('judul-
halaman', 'List Wisata Sejarah'
)

@section('content')

```

<div class="container-fluid">
    <div class="row m-5">
        <div class="col-md-12">
            <div class="card">
                <div class="car
d-header">
                    <div>
                        Wisata
Sejarah Terbaru
                    <a href
="{{ url('admin-wisata-
sejarah') }}" class="btn btn-
info float-right">Kembali</a>
                </div>
            </div>
        </div>
    </div>

```

```

<div class="car
d-body">
    <table clas
s="table table-striped">
        <thead
class="text-center">
            <tr>
                <th scope="col">No</th>
                <th scope="col">Title</th>
                <th scope="col">Description</th>
                <th scope="col">Longitude</th>
                <th scope="col">Latitude</th>
                <th scope="col">Foto</th>
                <th scope="col">Action</th>
            </thead>
            <tbody
class="text-center">
                @fo
reach ($list as $key => $item)
                <tr>
                    <td>{{ $item->id }}</td>
                    <td>{{ $item-
>title }}</td>
                    <td>{{ $item-
>description }}</td>
                    <td>{{ $item->long }}</td>

```

```

        <td>{{ $item->lat }}</td>

        <td><a href="<?=> asset('storage/images/'.$item->image); ?>" class="btn btn-info float-right">Lihat Foto</a></td>

    <td>

        <a href="{{ route('sejarah.edit', $item->id) }}" class="btn btn-warning">Edit</a>

        <form action="{{ route('sejarah.destroy', $item->id) }}" method="POST" class="inline-block">

            {!! method_field('delete') . csrf_field() !!}

            <button type="submit" class="btn btn-danger">

                Delete

            </button>

        </form>

    </td>

</tr>

@en
dforeach

</tbody>

>

</table>

</div>

</div>

</div>

```

```

</div>
<div class="row m-5">
    <div class="col-md-12">
        <div class="card">
            <div class="card-header">

                <div>

                    Wisata
                    Sejarah Menunggu di Verifikasi

                </div>

            </div>
            <div class="card-body">

                <table class="table table-striped">
                    <thead
                        class="text-center">
                            <tr>

                                <th scope="col">No</th>

                                <th scope="col">Title</th>

                                <th scope="col">Description</th>

                                <th scope="col">Longitude</th>

                                <th scope="col">Latitude</th>

                                <th scope="col">Foto</th>

                                <th scope="col">Action</th>

                            </tr>
                        </thead>
                        <tbody
                            class="text-center">
                                @foreach ($list2 as $key => $item)

                                    <tr>

```

```

<td>{{ ++$key }}</td>

<td>{{ $item-
>title }}</td>

<td>{{ $item-
>description }}</td>

<td>{{ $item->long }}</td>

<td>{{ $item->lat }}</td>

<td><a target="_blank" href=
f="<?= asset('storage/images/'.
$item-
>image); ?>" class="btn btn-
info float-
right">Lihat Foto</a></td>

<td>

<form action="{{ route
('sejarah.update', $item-
>id) }}" method="POST" class="d
-inline">

        {!! method_field('
PUT') . csrf_field() !!}

        <input type="hidde
n" name="setujui" value="1">

        <button type="subm
it" class="btn btn-success">

            Setujui

        </button>

    </form>

    <form action="{{ route
('sejarah.destroy', $item-

```

```

>id) }}" method="POST" class="d
-inline">

        {!! method_field('
delete') . csrf_field() !!}

        <button type="subm
it" class="btn btn-danger">

            Tolak

        </button>

    </form>

</td>

</tr>

@en
dforeach

</tbody>
>

</table>
</div>
</div>
</div>
</div>
</div>
</div>
@endsection
Wisata sejarah/edit.blade.php
@extends('layouts.app')
@section('judul-
halaman', 'Update Wisata Sejara
h')
@section('content')

<div class="container-fluid">
    <div class="row m-5">
        <div class="col-md12">
            <div class="card">
                <div class="car
d-header">

                    <div>

```

```

                                Update
Data Wisata Sejarah
                                <a href
                                ="{{ url('sejarah') }}" class="
                                btn btn-info float-
                                right">Kembali</a>
                                </div>
                                </div>
                                <div class="car
                                d-body">
                                <form actio
                                n="{{ route('sejarah.update', $
                                wisata-
                                >id) }}" method="POST" enctype=
                                "multipart/form-data">
                                @csrf
                                @method
                                ('PUT')
                                <div cl
                                ass="row">
                                <di
                                v class="col-sm-6">
                                <div class="form-group">
                                <label for="">Latitude</la
                                bel>
                                <input name="lat" type="te
                                xt" class="form-
                                control" value="{{ $wisata-
                                >lat }}" required>
                                @error('lat') <small class
                                ="text-
                                danger">{{ $message }}</small>@en
                                derror
                                </div>
                                </div>
                                </div>
                                <div
                                v class="col-sm-6">
                                <div class="form-group">

```

```

                                <label for="">Longitude</
                                label>
                                <input name="long" type="t
                                ext" class="form-
                                control" value="{{ $wisata-
                                >long }}" required>
                                @error('long') <small clas
                                s="text-
                                danger">{{ $message }}</small>@en
                                derror
                                </div>
                                </div>
                                <div cl
                                ass="form-group">
                                <la
                                bel for="">Title</label>
                                <in
                                put name="title" type="text" cl
                                ass="form-
                                control" value="{{ $wisata-
                                >title }}" required>
                                @er
                                ror('title') <small class="text
                                -
                                danger">{{ $message }}</small>@en
                                derror
                                </div>
                                <div cl
                                ass="form-group">
                                <la
                                bel for="">Description</label>
                                <te
                                xtaarea name="description" type=
                                "text" class="form-
                                control">{{ $wisata-
                                >description }}</textarea>
                                @er
                                ror('description') <small class
                                ="text-

```



```

        return div;
    }else if(route[i].type == 'event'){
        alamat = "/event";

    }else{
        alamat = ""
    }

    labels.push(
        `

<a href="${alamat}"><p
            style="margin-bottom:0; color: blue;">${route
            [i].title} - ${route[i].type}</p></a>

            <small>${route[i].jarak.toFixed(2)} Km</small>

            <hr style="margin:0">

        </div>`
    );
    }
    }else{
        lab
    els.push(
        `

Tidak terdapat rekomendasi tempat</div>`
    );
    }
    }

    div.innerHTML = labels.join("");


```

```

    });

    legend.addTo(map)
}

function getRecommendation(geoJson, latLng){
    let data = geoJson.recommended;
    let type = ['event', 'oleh-oleh', 'penginapan'];

    const jumlah = data[type[0]].length + data[type[1]].length + data[type[2]].length;

    let posisi = 0;

    let terbaik = [];

    type.forEach((el, idx)=>{
        data[type[idx]].forEach((el) => {
            let wp1 = L.Routing.waypoint(latLng)
            let wp2 = L.Routing.waypoint([el.lat, el.long])

            let routesUs = L.Routing.osrmv1();
            routesUs.route([wp1, wp2], (err, routes)=>{
                posisi++;

                if(!err){
                    let best = 100000000000;

                    let bestRoute = 0;

```

```

for(i in routes){

    if(routes[i].summary.total
Distance < best){

        bestRoute = i;

        best = routes[i].summa
ry.totalDistance;

    }

}

let jarak = best/1000;

if(terbaik.length < 7){

    if(jarak.toFixed(2) >= 0.5
0 && jarak.toFixed(2) <= 5 ){

        terbaik.push({

            id: el.id,

            title: el.title,

            type: type[idx],

            jarak: routes[best
Route].summary.totalDistance/10
00,

        })

    }

}

console.log(`Best Route ${type
[idx]}: ${best / 1000} Km`);
}el
se{

        console.error(err)
    }
    console.log(posisi, jumlah)
    if(
posisi === jumlah){

        tampilRekomendasi(terbaik, err
)

    }

})

})

})

}

    var markersLayer =
new L.LayerGroup();
    map.addLayer(marker
sLayer);

    function loadLocati
ons(geoJson){

        let data = geoJ
son.features;

        for (i in data)
        {

            var title =
data[i].properties.title,

            loc = [
Number(data[i].geometry.coordin
ates[1]), Number(data[i].geomet
ry.coordinates[0])],

            marker
= new L.Marker(new L.LatLng(loc
), {

                tit
le: data[i].properties.title,

            });

            var imageSt
orage = '{asset("/storage/imag

```

```

es"))}}' + '/' + data[i].properties.image
marker.bind
Popup(`
    <div class="overflow-7, auto; max-height:400px, width: 100%">
        <table class="table table-sm mt-2">
            <tbody>
                <tr>
                    <td>Title</td>
                    <td>${title}</td>
                </tr>
                <tr>
                    <td>Picture</td>
                    <td></td>
                </tr>
                <tr>
                    <td>Description</td>
                    <td>${data[i].properties.description}</td>
                </tr>
            </tbody>
        </table>
    </div>
</div>
<div class="text-center">

```

```

    <button class="btn btn-sm btn-primary keSini" data-lat="${data[i].geometry.coordinates[1]}" data-lng="${data[i].geometry.coordinates[0]}">Arahkan Kesini</button>
</div>
</div>
`);
markersLayer.addLayer(marker);
}
}

loadLocations({!! $geoJson !!})

//Search
var controlSearch =
new L.Control.Search({
    position: "topLeft",
    layer: markersLayer,
    initial: false,
    zoom: 16,
    marker: false,
});
map.addControl(controlSearch);

// rute
var routeControl =
L.Routing.control({
    waypoints: [latLng],
    addWaypoints: false,
    routeWhileDragging: false,

```

```

        draggableWaypoi
nts: false,
        createMarker: f
unction (i, waypoint, n) {
            let urlIcon
;
            var pos=i+1
;
            if(pos==1){
                urlIcon
='<?=asset('icons/user.png')?>'
;
            }else{
                urlIcon
='<?=asset('icons/dest.png')?>'
;
            }
            const marke
r = L.marker(waypoint.latLng,{
                icon: L
.icon({
                    ico
nUrl: urlIcon,
                    ico
nAnchor: [5, 45],
                    ico
nSize: [38, 45]
                })
            });
            return mark
er;
        }
    });
    routeControl.addTo(
map)
    routeControl.hide()

    $(document).on("cli
ck",".keSini",function(){
        let newLat = L.
latLng($(this).data('lat'), $(t
his).data('lng'));
        routeControl.sp
liceWaypoints(routeControl.getW

```

```

aypoints().length - 1, 1, newLa
t);
        routeControl.sh
ow()
    })
    var status = true;
    $('<div>.leaflet-
routing-
container</div>').click(function(e) {
        if(status){
            status = fa
lse;
            routeContro
l.hide()
        }else{
            status = tr
ue;
            routeContro
l.show()
        }
    })
    window.addEventList
ener('locationAdded', (e) => {
        console.log(e.d
etail)
        loadLocations(J
SON.parse(e.detail))
    })
    function setDataPil
ihempat(geoJson) {
        let data = geoJ
son.features;
        for (i in data)
        {
            let title =
data[i].properties.title
            let loc = [
Number(data[i].geometry.coordinates[1]), Number(data[i].geomet
ry.coordinates[0])];
            $('#titik-
awal').append(`

```

```

                <option
value="{{$loc}}">{{$title}}</option>
n>
                `)
                $('#titik-
akhir').append(`
                <option
value="{{$loc}}">{{$title}}</option>
n>
                `)
            }
        }
        setDataPilihtempat(
{!! $geoJson !!})
        $('#btn-
kesana').on('click', function()
        {
            let loc = $('#t
itik-akhir').val();
            let newLoc = lo
c.split(',');
            let newLat = L.
latLng(newLoc[0], newLoc[1]);
            let loc2 = $('#
titik-awal').val();
            let newLoc2 = l
oc2.split(',');
            let newLat2 = L
.latLng(newLoc2[0], newLoc2[1])
;
            routeControl.se
tWaypoints([newLat2, newLat]);
            routeControl.sh
ow()
        })
    })
</script>
@endpush
Livewire/p-umum.blade.php
<div class="container-fluid">
    <div class="row">
        <div class="col-md-8">
            <div class="card">

```

```

                <div class="car
d-header bg-dark text-white">
                    <a href="{{$
url('penginapan-
umum') }}" class="btn btn-
warning ml-1 mb-
1">Penginapan Umum</a>
                    <a href="{{$
url('penginapan-
syariah') }}" class="btn btn-
info ml-1 mb-
1">Penginapan Syariah</a>
                    <a href="{{$
route('penginapan.umum') }}" c
lass="btn btn-info float-
right">List</a>
                </div>
            <div class="car
d-body bg-dark text-white">
                <div wire:i
gnore id="map" style="width: 10
0%; height: 70vh;"></div>
            </div>
        </div>
        <div class="col-md-4">
            <div class="card">
                <div class="car
d-header">
                    Form
                </div>
                <div class="car
d-body">
                    <form wire:
submit.prevent="saveLocation">
                        <div cl
ass="row">
                            <di
v class="col-sm-6">
                                <div class="form-group">
                                    <label for="">Latitude</la
bel>

```

```

        <input wire:model="lat" type="text" class="form-control">

        @error('lat') <small class="text-danger">{{ $message }}</small>@enderror

    </div>
</div>
</div>
<div class="col-sm-6">

    <div class="form-group">

        <label for="">Longitude</label>

        <input wire:model="long" type="text" class="form-control">

        @error('long') <small class="text-danger">{{ $message }}</small>@enderror

    </div>
</div>
</div>
<div class="form-group">

    <label for="">Title</label>

    <input wire:model="title" type="text" class="form-control">

    @error('title') <small class="text-danger">{{ $message }}</small>@enderror

    </div>
</div>
<div class="form-group">

    <label for="">Description</label>

    <textarea wire:model="description" type="text" class="form-control"></textarea>

    @error('description') <small class="text-danger">{{ $message }}</small>@enderror

    </div>
<div class="form-group">

    <label class="text-white">Image</label>

    <div class="custom-file dark-input">

        <input wire:model="image" type="file" class="custom-file-input" id="customFile">

        <label class="custom-file-label dark-input" for="customFile">Choose file</label>

    </div>

    <label class="text-white">Picture of Location</label>

    @error('image') <small class="text-danger">{{ $message }}</small>@enderror

    @if ($image)

```



```

        latLng = [position.coords.latitude, position.coords.longitude];
        map.setView(latLng, 14, {animation: true});
        routeControl.spliceWaypoints(0, 1, latLng);
        getRecomendation({!! $geoJson !!}, latLng);
    }

    var legend = L.control({ position: "bottomleft" })
    ,
        labels = [];
        labels2 = [];

    legend.onAdd = function (map) {
        var div = L.DomUtil.create("div", "info legend");

        labels2.push(`<h6>Rekomendasi Tempat</h6>`)
        labels2.push(`<p>Sedang memuat rekomendasi tempat</p>`)

        div.innerHTML = labels2.join("");
        return div;
    };

    legend.addTo(map)

    var pilihTempat = L.control({ position: "topright" }),
        items = [];
        pilihTempat.onAdd = function (map) {
            var div = L.DomUtil.create("div", "info");
            items.push(`
                <select id=
"titik-awal">

```

```

                <option
selected disabled>Pilih Titik
Awal</option>

                </select>
                <select id=
"titik-akhir">

                <option
selected disabled>Pilih Titik
Akhir</option>

                </select>
                <button id=
"btn-kesana">kesana</button>

            `)
            div.innerHTML =
items.join("");
            return div;
        };
        pilihTempat.addTo(m
ap)

        function tampilReko
mendasi(route, err){
            map.removeContr
ol(legend);
            legend.onAdd =
function (map) {
                var div = L
.DomUtil.create("div", "info le
gend");

                labels.push
(`<h6>Rekomendasi Tempat</h6>`)
                if(err){
                    labels.
push(`<p>Maaf, terjadi kesalaha
n saat mendapatkan lokasi</p>`)
                }else{
                    let ala
mat=""

                    if(rout
e.length > 0){
                        for
(var i = 0; i < route.length;
i++) {

```

```

    if(route[i].type == 'penginapan') {
        alamat = "/penginapan-umum";

    } else if(route[i].type == 'oleh-oleh') {
        alamat = "/tempat-oleh-oleh";

    } else if(route[i].type == 'wisata') {
        alamat = "/wisata-religi";

    } else if(route[i].type == 'event') {
        alamat = "/event";

    } else {
        alamat = ""
    }

    labels.push(
        `<div>

            <a href="${alamat}"><p style="margin-bottom:0; color: blue;">${route[i].title} - ${route[i].type}</p></a>

            <small>${route[i].jarak.toFixed(2)} Km</small>

            <hr style="margin:0">

```

```

        </div>`
    );
    }
    } else {
        lab

    els.push(
        `<div>Tidak terdapat rekomendasi tempat</div>`
    );
    }
    }

    div.innerHTML
    ML = labels.join("");
    return div;
    };

    legend.addTo(map)
    }

    function getRecommendation(geoJson, latLng){
        let data = geoJson.recommended;
        let type = ['event', 'oleh-oleh', 'wisata'];
        const jumlah = data[type[0]].length + data[type[1]].length + data[type[2]].length;

        let posisi = 0;

        let terbaik = [
        ];

        type.forEach((el, idx)=>{
            data[type[idx]].forEach((el) => {
                let wp1 = L.Routing.waypoint(latLng)

```

```

        let wp2
        = L.Routing.waypoint([el.lat,
        el.long])

        let routesUs = L.Routing.osrmv1();
        routesUs.route([wp1, wp2], (err, routes) => {
            posisi = routes[bestRoute].summary.totalDistance/1000,
            type: type[idx],
            jarak: routes[bestRoute].summary.totalDistance/1000,
        })

            isi++;

            if(!err){
                let best = 100000000000;
                let bestRoute = 0;

                for(i in routes){
                    if(routes[i].summary.totalDistance < best){
                        bestRoute = i;
                        best = routes[i].summary.totalDistance;
                    }
                }

                let jarak = best/1000;

                if(terbaik.length < 7){
                    if(jarak.toFixed(2) >= 0.5 && jarak.toFixed(2) <= 5 ){
                        terbaik.push({
                            id: el.id,
                            title: el.title,
                            type: type[idx],
                            jarak: routes[bestRoute].summary.totalDistance/1000,
                        })
                    }
                }

                console.log(`Best Route ${type[idx]}: ${best / 1000} Km`);
            } else {
                console.error(err)
            }
        })

        console.log(posisi, jumlah)
        if(posisi === jumlah){
            tampilRekomendasi(terbaik, err)
        }
    })
})

var markersLayer =
new L.LayerGroup();
map.addLayer(markersLayer);

function loadLocations(geoJson){
    let data = geoJson.features;

```

```

        for (i in data)
        {
            var title =
            data[i].properties.title,
            loc = [
            Number(data[i].geometry.coordinates[1]), Number(data[i].geometry.coordinates[0])],
            marker
            = new L.Marker(new L.LatLng(loc
            ), {
            title: data[i].properties.title,
            });
            var imageStorage = '{{asset("/storage/images")}}' + '/' + data[i].properties.image
            marker.bind
            Popup(`
            <div class="overflow-7, auto; max-height:400px, width: 100%">
            <table class="table table-sm mt-2">

            <tbody>

            <tr>

            <td>Title</td>

            <td>${title}</td>

            </tr>

            <tr>

            <td>Picture</td>

            <td></td>

```

```

            </tr>

            <tr>

            <td>Description</td>

            <td>${data[i].properties.description}</td>

            </tr>

            </tbody>

            </table>

            <div class="text-center">

            <button class="btn btn-sm btn-primary keSini" data-lat="${data[i].geometry.coordinates[1]}" data-
            lng="${data[i].geometry.coordinates[0]}">Arahkan Kesini</button>

            </div>

            `);
            markersLayer.addLayer(marker);
        }

        loadLocations({!! $geoJson !!})

        //Search
        var controlSearch =
        new L.Control.Search({
            position: "topleft",
            layer: markersLayer,
            initial: false,

```

```

        zoom: 16,
        marker: false,
    });
    map.addControl(controlSearch);

    // route
    var routeControl =
    L.Routing.control({
        waypoints: [lat
    Lng],
        addWaypoints: false,
        routeWhileDragging: false,
        draggableWaypoints: false,
        createMarker: function (i, waypoint, n) {
            let urlIcon
            ;
            var pos=i+1
            ;
            if(pos==1){
                urlIcon
                ='<?=asset('icons/user.png')?>'
            ;
            }else{
                urlIcon
                ='<?=asset('icons/dest.png')?>'
            ;
            }
            const marker = L.marker(waypoint.latLng,{
                icon: L
                .icon({
                    iconUrl: urlIcon,
                    iconAnchor: [5, 45],
                    iconSize: [38, 45]
                })
            });
        },
        return marker;
    });
    routeControl.addTo(
    map)
    routeControl.hide()

    $(document).on("click",".keSini",function(){
        let newLat = L.
        latLng($(this).data('lat'), $(this).data('lng'));
        routeControl.spliceWaypoints(routeControl.getWaypoints().length - 1, 1, newLat);
        routeControl.show()
    })

    var status = true;
    $('.leaflet-routing-container').click(function(e) {
        if(status){
            status = false;
            routeControl.hide()
        }else{
            status = true;
            routeControl.show()
        }
    })

    window.addEventListener('locationAdded', (e) => {
        console.log(e.detail)
        loadLocations(JSON.parse(e.detail))
    })

```



```

<div class="form-group">
    <label for="">Latitude</label>

    <input wire:model="lat" type="text" class="form-control">

    @error('lat') <small class="text-danger">{{ $message }}</small>@enderror

</div>
</div>
<div class="col-sm-6">

    <div class="form-group">

        <label for="">Longitude</label>

        <input wire:model="long" type="text" class="form-control">

        @error('long') <small class="text-danger">{{ $message }}</small>@enderror

    </div>
</div>
</div>
<div class="form-group">
    <label for="">Title</label>

    <input wire:model="title" type="text" class="form-control">

    @error('title') <small class="text-danger">{{ $message }}</small>@enderror
</div>
</div>
<div class="form-group">
    <label for="">Description</label>

    <textarea wire:model="description" type="text" class="form-control"></textarea>

    @error('description') <small class="text-danger">{{ $message }}</small>@enderror
</div>
<div class="form-group">
    <label class="text-white">Image</label>

    <div class="custom-file dark-input">

        <input wire:model="image" type="file" class="custom-file-input" id="customFile">

        <label class="custom-file-label dark-input" for="customFile">Choose file</label>

    </div>

    <label class="text-white">Picture of Location</label>

    @error('image') <small class="text-

```



```

                                for
    (var i = 0; i < route.length;
    i++) {

        if(route[i].type == 'penginapan') {

            alamat = "/penginapan-
            umum"

        } else if(route[i].type == 'oleh-
        oleh') {

            alamat = "/tempat-oleh-
            oleh";

        } else if(route[i].type == 'wisata') {

            alamat = "/wisata-religi";

        } else if(route[i].type == 'event') {

            alamat = "/event";

        } else {

            alamat = ""

        }

        labels.push(

            `

<a href="${alamat}"><p
                style="margin-
                bottom:0; color: blue;">${route
                [i].title} - ${route[i].type}</
                p></a>


```

```

                <small>${route[i].jara
                k.toFixed(2)} Km</small>

                <hr style="margin:0">

            </div>`

        );

    }

    } else {

        lab

        els.push(

            `

```

```

        data[type[idx]].forEach((el) => {
            let wp1
            = L.Routing.waypoint(latLng)
            let wp2
            = L.Routing.waypoint([el.lat,
            el.long])

            let routesUs = L.Routing.osrmv1();
            routesUs.route([wp1, wp2], (err, routes) => {
                posisi
                isi++;
                if(!err){
                    let best = 100000000000;
                    let bestRoute = 0;

                    for(i in routes){
                        if(routes[i].summary.totalDistance < best){
                            bestRoute = i;
                            best = routes[i].summary.totalDistance;
                        }
                    }

                    let jarak = best/1000;

                    if(terbaik.length < 7){
                        if(jarak.toFixed(2) >= 0.5
                        0 && jarak.toFixed(2) <= 5 ){
                            terbaik.push({
                                id: el.id,
                                title: el.title,
                                type: type[idx],
                                jarak: routes[bestRoute].summary.totalDistance/1000,
                            })

                            console.log(`Best Route ${type[idx]}: ${best / 1000} Km`);
                        }el
                        se{
                            console.error(err)
                        }
                        console.log(posisi, jumlah)
                        if(posisi === jumlah){
                            tampilRekomendasi(terbaik, err)
                        }
                    })
                })
            })

            var markersLayer =
            new L.LayerGroup();
            map.addLayer(markersLayer);

```

```

        function loadLocations(geoJson){
            let data = geoJson.features;
            for (i in data)
            {
                var title = data[i].properties.title,
                    loc = [Number(data[i].geometry.coordinates[1]), Number(data[i].geometry.coordinates[0])],
                    marker = new L.Marker(new L.LatLng(loc), {
                        title: data[i].properties.title,
                    });
                var imageStorage = '{{asset("/storage/images")}}' + '/' + data[i].properties.image;
                marker.bindPopup(`
                    <div class="overflow-7, auto; max-height:400px, width: 100%">
                        <table class="table table-sm mt-2">
                            <tbody>
                                <tr>
                                    <td>Title</td>
                                    <td>${title}</td>
                                </tr>
                                <tr>
                                    <td>Picture</td>

```

```

                                    <td></td>
                                </tr>
                                <tr>
                                    <td>Description</td>
                                    <td>${data[i].properties.description}</td>
                                </tr>
                            </tbody>
                        </table>
                        <div class="text-center">
                            <button class="btn btn-sm btn-primary keSini" data-lat="${data[i].geometry.coordinates[1]}" data-
                                lng="${data[i].geometry.coordinates[0]}">Arahkan Kesini</button>
                        </div>
                    `);
                markersLayer.addLayer(marker);
            }
        }

        loadLocations({!! $geoJson !!})

        //Search
        var controlSearch = new L.Control.Search({

```

```

        position: "topLeft",
        iconSize: [38, 45],
        layer: markersLayer,
        initial: false,
        zoom: 16,
        marker: false,
    });
    map.addControl(controlSearch);

    // route
    var routeControl = L.Routing.control({
        waypoints: [latLng, lng],
        addWaypoints: false,
        routeWhileDragging: false,
        draggableWaypoints: false,
        createMarker: function(i, waypoint, n) {
            let urlIcon;

            var pos=i+1;

            if(pos==1){
                urlIcon = '<?=' + asset('icons/user.png') + '?>';
            } else {
                urlIcon = '<?=' + asset('icons/dest.png') + '?>';
            }
            const marker = L.marker(waypoint.latLng, {
                icon: L.icon({
                    imageUrl: urlIcon,
                    iconAnchor: [5, 45],
                    iconSize: [38, 45]
                })
            });
            map.addControl(routeControl.addTo(map));
            routeControl.hide();

            $(document).on("click", ".keSini", function() {
                let newLat = L.latLng(this.data('lat'), this.data('lng'));
                routeControl.spliceWaypoints(routeControl.getWaypoints().length - 1, 1, newLatLng);
                routeControl.show();
            });

            var status = true;
            $('.leaflet-routing-container').click(function(e) {
                if(status) {
                    status = false;
                    routeControl.hide();
                } else {
                    status = true;
                    routeControl.show();
                }
            });
        },
        nAnchor: [5, 45],
        nSize: [38, 45]
    });
    return marker;
}
});

```

```

        window.addEventListener('locationAdded', (e) => {
            console.log(e.detail)

            loadLocations(JSON.parse(e.detail))
        })

        function setDataPilihTempat(geoJson) {
            let data = geoJson.features;

            for (i in data)
            {
                let title = data[i].properties.title
                let loc = [Number(data[i].geometry.coordinates[1]), Number(data[i].geometry.coordinates[0])];
                $('#titik-awal').append(`
                    <option
                    value="${loc}">${title}</option>
                `)
                $('#titik-akhir').append(`
                    <option
                    value="${loc}">${title}</option>
                `)
            }
            setDataPilihTempat({!! $geoJson !!})
            $('#btn-kesana').on('click', function()
            {
                let loc = $('#titik-akhir').val();
                let newLoc = loc.split(',');
                let newLat = L.latLng(newLoc[0], newLoc[1]);

```

```

                let loc2 = $('#titik-awal').val();
                let newLoc2 = loc2.split(',');
                let newLat2 = L.latLng(newLoc2[0], newLoc2[1]);
                routeControl.setWaypoints([newLat2, newLat]);
                routeControl.show()
            })
        }
    </script>
@endpush
Livewire/acara.blade.php
<div class="container-fluid">
    <div class="row">
        <div class="col-md-12">
            <div class="card">
                <div class="card-header bg-dark text-white">
                    <div class="Event">
                        <a href="{{ url('list-event') }}" class="btn btn-info float-right">List</a>
                    </div>
                </div>
                <div class="card-body bg-dark text-white">
                    <div wire:ignore id="map" style="width: 100%; height: 70vh;"></div>
                </div>
            </div>
        </div>
    </div>
</div>
@push('scripts')
    <script>

```

```

        document.addEventListener(
        'livewire:load', () => {
            let latLng = [3.591
            393241276196, 98.67736409332781
            ];

            let user = [];
            var map = L.map("map", {
                fullscreenControl: {
                    pseudoFullscreen: false,
                },
            }).setView(latLng,
            14);

            L.tileLayer(
                "https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token=pk.eyJ1IjoibWFwYm94IiwiaWUiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ3N3gifQ.rJcFIG214AriISLbB6B5aw",
                {
                    maxZoom: 18
                },
                {
                    id: "mapbox/streets-v11",
                    tileSize: 512,
                    zoomOffset: -1,
                }
            ).addTo(map);

            getUserLoc()
            // setInterval(function(){ getUserLoc(); }, 3000);

            function getUserLoc()
            {
                if (navigator.geolocation) {

```

```

                navigator.geolocation.getCurrentPosition(showPosition)
                } else {
                    alert("Geolocation is not supported by this browser.")
                }
            }

            function showPosition(position) {
                let latLng = [position.coords.latitude, position.coords.longitude];
                map.setView(latLng, 14, {animation: true});
                routeControl.spliceWaypoints(0, 1, latLng);
                getRecommendation({!! $geoJson !!}, latLng);
            }

            var legend = L.control({ position: "bottomleft" })
            ,
                labels = [];
                labels2 = [];

            legend.onAdd = function (map) {
                var div = L.DomUtil.create("div", "info legend");

                labels2.push(`<h6>Rekomendasi Tempat</h6>`)
                labels2.push(`<p>Sedang memuat rekomendasi tempat</p>`)

                div.innerHTML = labels2.join("");
                return div;
            };

            legend.addTo(map)

```

```

        var pilihTempat = L
        .control({ position: "topright"
        })),
        items = [];
        pilihTempat.onAdd =
        function (map) {
            var div = L.Dom
            Util.create("div", "info");
            items.push(`
                <select id=
                "titik-awal">
                    <option
                    selected disabled>Pilih Titik
                    Awal</option>
                </select>
                <select id=
                "titik-akhir">
                    <option
                    selected disabled>Pilih Titik
                    Akhir</option>
                </select>
                <button id=
                "btn-kesana">kesana</button>
            `)
            div.innerHTML =
            items.join("");
            return div;
        };
        pilihTempat.addTo(m
        ap)

        function tampilReko
        mendasi(route, err){
            map.removeContr
            ol(legend);
            legend.onAdd =
            function (map) {
                var div = L
                .DomUtil.create("div", "info le
                gend");
                labels.push
                (`<h6>Rekomendasi Tempat</h6>`)
                if(err){

```

```

                labels.
                push(`<p>Maaf, terjadi kesalaha
                n saat mendapatkan lokasi</p>`)
            }else{
                let ala
                mat=""

                if(rout
                e.length > 0){
                    for
                    (var i = 0; i < route.length;
                    i++) {

                        if(route[i].type == 'penginapa
                        n'){

                            alamat = "/penginapan-
                            umum"

                        }else if(route[i].type == 'ole
                        h-oleh'){

                            alamat = "/tempat-oleh-
                            oleh";

                        }else if(route[i].type == 'wis
                        ata'){

                            alamat = "/wisata-religi";

                        }else if(route[i].type == 'eve
                        nt'){

                            alamat = "/event";

                        }else{

                            alamat = ""

                        }

                        labels.push(

                            `<div>

```



```

        <a href="{alamat}"><p
        style="margin-
        bottom:0; color: blue;">{route
        [i].title} - {route[i].type}</
        p></a>

        <small>{route[i].jara
        k.toFixed(2)} Km</small>

        <hr style="margin:0">

        </div>`

    );

    }
    }else{
        lab
    els.push(

        `<div>Tidak terdapat rekomenda
        si tempat</div>`

    );
    }

    div.innerHT
    ML = labels.join("");
    return div;
    };

    legend.addTo(ma
    p)

    }

    function getRecomen
    dation(geoJson, latLng){
        let data = geoJ
        son.recommended;
        let type = ['wi
        sata', 'oleh-
        oleh', 'penginapan'];
        const jumlah =
        data[type[0]].length + data[typ

```

```

e[1]].length + data[type[2]].le
ngth;

        let posisi = 0;

        let terbaik = [
    ];

        type.forEach((e
        l, idx)=>{
            data[type[i
            dx]].forEach((el) => {
                let wp1
                = L.Routing.waypoint(latLng)
                let wp2
                = L.Routing.waypoint([el.lat,
                el.long])

                let rou
                tesUs = L.Routing.osrmv1();
                routesU
                s.route([wp1, wp2], (err, route
                s)=>{
                    pos
                    isi++;

                    if(
                    !err){

                        let best = 1000000000000;

                        let bestRoute = 0;

                        for(i in routes){

                            if(routes[i].summary.total
                            Distance < best){

                                bestRoute = i;

                                best = routes[i].summa
                                ry.totalDistance;

                            }

                        }
                    }
                }
            }
        }
    }

```



```

        <td>${title}</td>

    </tr>

    <tr>

        <td>Picture</td>

        <td></td>

    </tr>

    <tr>

        <td>Description</td>

        <td>${data[i].properties.description}</td>

    </tr>

</tbody>

</table>

<div class="text-center">

    <button class="btn btn-sm btn-primary keSini" data-lat="${data[i].geometry.coordinates[1]}" data-lng="${data[i].geometry.coordinates[0]}">Arahkan Kesini</button>

</div>

`);
markersLayer.addLayer(marker);
}
}

```

```

        loadLocations({!! $geoJson !!})

//Search
var controlSearch =
new L.Control.Search({
    position: "topLeft",
    layer: markersLayer,
    initial: false,
    zoom: 16,
    marker: false,
});
map.addControl(controlSearch);

// rute
var routeControl =
L.Routing.control({
    waypoints: [latLng],
    addWaypoints: false,
    routeWhileDragging: false,
    draggableWaypoints: false,
    createMarker: function (i, waypoint, n) {
        let urlIcon
        ;
        var pos=i+1
        ;
        if(pos==1){
            urlIcon
            ='<?=asset('icons/user.png')?>'
            ;
        }else{
            urlIcon
            ='<?=asset('icons/dest.png')?>'
            ;
        }
    }
}

```

```

const marker = L.marker(waypoint.latLng, {
  icon: L.icon({
    iconUrl: urlIcon,
    iconSize: [38, 45],
    iconAnchor: [5, 45],
  })
});
return marker;
});
routeControl.addTo(map);
routeControl.hide();

$(document).on("click", ".keSini", function() {
  let newLat = L.latLng($(this).data('lat'), $(this).data('lng'));
  routeControl.spliceWaypoints(routeControl.getWaypoints().length - 1, 1, newLat);
  routeControl.show();
});

var status = true;
$('#leaflet-routing-container').click(function(e) {
  if(status){
    status = false;
  }else{
    status = true;
  }
});

```

```

routeControl.show();
});

window.addEventListener('locationAdded', (e) => {
  console.log(e.detail);
  loadLocations(JSON.parse(e.detail));
});

function setDataPilihTempat(geoJson) {
  let data = geoJson.features;
  for (i in data) {
    let title = data[i].properties.title;
    let loc = [Number(data[i].geometry.coordinates[1]), Number(data[i].geometry.coordinates[0])];
    $('#titik-awal').append(`
      <option value="${loc}">${title}</option>
    `);
    $('#titik-akhir').append(`
      <option value="${loc}">${title}</option>
    `);
  }
  setDataPilihTempat(!! $geoJson !!);
  $('#btn-kesana').on('click', function() {
    {

```

```

        let loc = $('#titik-akhir').val();
        let newLoc = loc.split(',');
        let newLat = L.latLng(newLoc[0], newLoc[1]);
        let loc2 = $('#titik-awal').val();
        let newLoc2 = loc2.split(',');
        let newLat2 = L.latLng(newLoc2[0], newLoc2[1]);
        ;
        routeControl.setWaypoints([newLat2, newLat]);
        routeControl.show();
    })

})
</script>
@endpush

```