

**PEMBANGUNAN APLIKASI *SOCIAL GEOTAGGING* DESTINASI
WISATA BERBASIS ANDROID**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ulfani Defitria

NIM: 145150207111043



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PEMBANGUNAN APLIKASI *SOCIAL GEOTAGGING* DESTINASI WISATA BERBASIS
ANDORID

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ulfani Defitria
NIM: 145150207111043

Skripsi ini telah diuji dan dinyatakan lulus pada
30 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


Bayu Priyambadha, S.Kom, M.Kom
NIP: 19820909200812 1 004


Denny Sagita R., S.Kom, M.Kom
NIP: 19851124 201504 1 001

Mengetahui

Ketua Jurusan Teknik Informatika




OTC Kornianwan, S.T, M.T, Ph.D
NIP: 19710518200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Juli 2018



Ulfani Defitria

NIM: 145150207111043

KATA PENGANTAR

Puji dan syukur kehadiran Allah Subhanahu Wa Ta'ala yang telah melimpahkan rahmat dan karunia-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi dengan judul : "PEMBANGUNAN APLIKASI SOCIAL GEOTAGGING DESTINASI WISATA BERBASIS ANDROID".

Penulisan skripsi ini diajukan untuk memenuhi salah satu persyaratan untuk meraih gelar sarjana pada Jurusan Informatika Fakultas Ilmu Komputer Universitas Brawijaya. Kelancaran dalam proses penelitian serta penulisan skripsi ini tidak lepas dari bimbingan serta kerja sama dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Kedua orang tua penulis, Alm. Ir. M. Azmi dan Almh. Deni Marni yang telah memberi motivasi, kasih sayang serta dukungan kepada Penulis.
2. Bapak Bayu Priyambadha, S.Kom, M.Kom selaku Pembimbing I, yang telah membimbing serta mengarahkan penulis dalam proses pengerjaan skripsi ini sehingga skripsi ini dapat terselesaikan dengan baik.
3. Bapak Denny Sagita R., S.Kom, M.Kom selaku Pembimbing II, yang telah membimbing dan mengarahkan penulis dalam proses penyusunan laporan skripsi sehingga skripsi ini dapat terselesaikan dengan baik.
4. Kedua saudara Penulis, Milza Apriliani dan Raafi Adepangga yang telah memberikan semangat dan dukungan kepada penulis sampai akhir pengerjaan skripsi ini.
5. Segenap Bapak dan Ibu Dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
6. Sasha, Imelda, miracle dan teman-teman lainnya yang telah membantu penulis jika penulis mengalami masalah dalam proses pengerjaan skripsi ini.
7. Teman-teman seperjuangan informatika 2014 yang telah banyak memberikan segala cerita, kenangan, dan support dalam menempuh pendidikan bersama-sama di Universitas Brawijaya.
8. Seluruh pihak yang telah membantu penulis hingga skripsi ini terselesaikan dengan baik yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun akan penulis terima dengan senang hati untuk penyempurnaan dari segala kekurangan yang ada pada penelitian ini. Akhir kata, penulis harap skripsi ini dapat memberikan manfaat dan menambah wawasan bagi pembaca dan penulis sendiri.

Malang, 17 Juli 2018

Penulis

udefitria@gmail.com



ABSTRAK

Berwisata sudah menjadi gaya hidup seseorang untuk mengisi waktu luang ataupun sekedar hobi. Aktivitas pariwisata tidak lepas dari dokumentasi sebagai suatu sarana untuk mengenang peristiwa wisata. Perkembangan teknologi *mobile* saat ini telah dilengkapi dengan kamera digital di dalamnya sehingga dapat digunakan secara mudah untuk mengambil foto wisata bagi para pengunjung wisata. Dengan dukungan jaringan internet, memungkinkan pula bagi wisatawan untuk mendistribusikan foto agar bisa dinikmati oleh banyak orang dalam waktu singkat melalui perantara media social. Perangkat *mobile* berupa *smartphone* saat ini sudah mendukung kamera yang memiliki fitur *geotagging* sehingga letak pengambilan foto dapat dengan mudah diketahui. Berbagai referensi berserakan di berbagai media cetak dan elektronik. Media-media tersebut kurang efektif dalam memberikan informasi, hal tersebut dikarenakan penelusuran informasi dari sumber yang tidak tertata sehingga membutuhkan waktu yang lama untuk mencarinya. Sehingga agar informasi tersebut dapat dengan mudah diperoleh sebaiknya referensi wisata dikumpulkan dalam suatu wadah berupa komunitas yang mudah di akses dengan memberikan rekomendasi berdasarkan faktor pendukung wisata seperti jarak, waktu, dan biaya. Penelitian ini dimulai dengan menganalisis kebutuhan sistem, kemudian melakukan tahap perancangan dan implementasi. Pada penelitian ini menerapkan fitur *geotagging* untuk mendapatkan lokasi dari foto wisata dan rumus haversine untuk pencarian lokasi terdekat. Selanjutnya dilakukan tahap pengujian sistem dengan menggunakan pengujian unit, pengujian validasi, serta pengujian *compatibility*. Hasil dari pengujian validasi menunjukkan bahwa sistem yang dibangun sudah berjalan sesuai dengan fungsinya baik dalam kondisi normal maupun kondisi alternatif, dan hasil pengujian *compatibility* menunjukkan bahwa sistem yang dibangun berjalan dengan baik pada versi sistem operasi Android yang berbeda.

Kata kunci: *geotagging*, haversine, destinasi wisata

ABSTRACT

Traveling has become a lifestyle for someone to fill their spare time or hobby. Tourism activity can not be separated from the documentation as a means to commemorate the tourist event. Nowadays, the development of mobile technology has been equipped with a digital camera in it, so it can be used easily to take photos of tours for the visitors. With the support of Internet network, it is possible for tourists to distribute photos in order to be enjoyed by many people in a short time through intermediaries of social media. Mobile devices such as smartphones currently supports cameras that have geotagging feature, so that the location of shooting can easily to known. There are some various references are scattered in various print and electronic media. These media are less effective in providing information, it is because of the searching of information from unregulated sources, so it takes a long time to look for them. So that the information can be obtained easily, with a reference tours are collected in a container in the form of community is easily to accessed by recommendations provide based on supporting factors such as distance, time, and cost. This research begins by analyzing system requirements, then performing design and implementation phases. In this study apply the geotagging feature to get the location of the tourist photos and haversine formula to search the nearest location. The testing is done by using unit testing, validation testing, and compatibility testing. The results of the validation testing indicate that the built system has been run in accordance with its function in both of normal function and alternative conditions, and compatibility test results indicate that the built system runs well on different versions of Android operating system.

Keywords: *geotagging, haversine, tourist destination.*



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan Penelitian	3
1.4 Manfaat.....	3
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.1.1 Penelitian Aplikasi <i>Geotagging</i>	6
2.1.2 Penelitian Aplikasi Pencarian Lokasi Terdekat.....	7
2.2 Wisata	7
2.3 Geotagging.....	8
2.4 Android	9
2.5 Google Maps API.....	9
2.6 Rumus Haversine	10
2.7 PhoneGap.....	10
2.8 CodeIgniter	11
2.9 Pengembangan Perangkat Lunak	12
2.9.1 Model Pengembangan Perangkat Lunak.....	12
2.9.2 Model <i>Waterfall</i>	12
2.10 <i>Unified Modeling Language</i> (UML).....	14



2.11 Pengujian Perangkat Lunak	14
2.11.1 Pengujian <i>Black-Box</i>	15
2.11.2 Pengujian <i>White-Box</i>	15
BAB 3 METODOLOGI	17
3.1 Studi Literatur	17
3.2 Analisis Kebutuhan	18
3.3 Perancangan Sistem.....	18
3.4 Implementasi	19
3.5 Pengujian	19
3.6 Kesimpulan Dan Saran	20
BAB 4 Analisis kebutuhan.....	21
4.1 Gambaran Umum Aplikasi	21
4.2 Analisis Kebutuhan	21
4.2.1 Identifikasi Aktor	21
4.2.2 Analisis Kebutuhan Fungsional	22
4.2.3 Analisis Kebutuhan Non Fungsional.....	45
BAB 5 Perancangan dan implementasi sistem.....	46
5.1 Perancangan Sistem.....	46
5.1.1 Perancangan Arsitektur Sistem.....	46
5.1.2 Pemodelan <i>Sequence Diagram</i>	46
5.1.3 Pemodelan <i>Class Diagram</i>	49
5.1.4 Perancangan Komponen	51
5.1.5 Perancangan Data	56
5.1.6 Perancangan Antarmuka.....	60
5.2 Implementasi Sistem	66
5.2.1 Spesifikasi Sistem	66
5.2.2 Batasan Implementasi.....	67
5.2.3 Implementasi <i>Database</i>	67
5.2.4 Implementasi Antarmuka	68
BAB 6 Pengujian sistem.....	72
6.1 Pengujian Unit.....	72
6.1.1 Pengujian Unit Klas Agenda Untuk Operasi kirimagenda().....	72



6.1.2 Pengujian Unit Klas Pos Untuk Operasi kirimpos()	74
6.1.3 Pengujian Unit Klas Admin Untuk Operasi blokir()	76
6.2 Pengujian Validasi	78
6.2.1 Pengujian Validasi Register	78
6.2.2 Pengujian Validasi Login	79
6.2.3 Pengujian Validasi Lihat Post Terbaru	80
6.2.4 Pengujian Validasi Lihat Profil	81
6.2.5 Pengujian Validasi Tambah Post	81
6.2.6 Pengujian Validasi Ambil Foto Dari Kamera	82
6.2.7 Pengujian Validasi Ambil Foto Dari Galeri	82
6.2.8 Pengujian Validasi Tambah Lokasi	82
6.2.9 Pengujian Validasi Lihat Lokasi Terdekat	83
6.2.10 Pengujian Validasi Lihat Wisata Termurah	84
6.2.11 Pengujian Validasi Lihat Wisata Termahal	84
6.2.12 Pengujian Validasi Lihat Agenda	84
6.2.13 Pengujian Validasi Tambah Agenda	85
6.2.14 Pengujian Validasi Edit Agenda	86
6.2.15 Pengujian Validasi Hapus Agenda	86
6.2.16 Pengujian Validasi Cari Pengguna	87
6.2.17 Pengujian Validasi Lihat Profil Pengguna Lain	88
6.2.18 Pengujian Validasi Ikuti Pengguna	88
6.2.19 Pengujian Validasi Berhenti Mengikuti Pengguna	88
6.2.20 Pengujian Validasi Cari Lokasi	89
6.2.21 Pengujian Validasi Lihat Post Berdasarkan Lokasi	90
6.2.22 Pengujian Validasi Tambah Komentar	90
6.2.23 Pengujian Validasi Lihat Komentar	91
6.2.24 Pengujian Validasi Edit Profil	91
6.2.25 Pengujian Validasi Hapus Post	91
6.2.26 Pengujian Validasi Laporkan Post	92
6.2.27 Pengujian Validasi Lihat Notifikasi	92
6.2.28 Pengujian Validasi Logout	92
6.2.29 Pengujian Validasi Login Admin	93



6.2.30 Pengujian Validasi Lihat Daftar Laporan Post	94
6.2.31 Pengujian Validasi Hapus Post Dari Daftar Laporan	94
6.2.32 Pengujian Validasi Hapus Post	95
6.3 Pengujian <i>Compatibility</i>	95
6.4 Analisis Hasil Pengujian.....	97
6.4.1 Analisis Hasil Pengujian Unit	97
6.4.2 Analisis Hasil Pengujian Validasi	97
6.4.3 Analisis Hasil Pengujian <i>Compatibility</i>	98
BAB 7 Penutup	99
11.1 Kesimpulan.....	99
11.2 Saran	99
DAFTAR PUSTAKA.....	100



DAFTAR TABEL

Tabel 2.1 Penelitian Aplikasi Geotagging.....	6
Tabel 2.2 Penelitian Aplikasi Lokasi Terdekat.....	7
Tabel 2.3 Fitur PhoneGap.....	11
Tabel 2.4 Tabel Nilai Kompleksitas.....	16
Tabel 4.1 Identifikasi Aktor.....	22
Tabel 4.2 Judul Penelitian dan Fitur yang Diterapkan.....	22
Tabel 4.3 Spesifikasi Kebutuhan Fungsional Aplikasi Pengguna.....	23
Tabel 4.4 Spesifikasi Kebutuhan Fungsional Aplikasi Admin.....	29
Tabel 4.5 <i>Use Case</i> Skenario <i>Register</i>	32
Tabel 4.6 <i>Use Case</i> Skenario Login.....	32
Tabel 4.7 <i>Use Case</i> Skenario Lihat Post Terbaru.....	33
Tabel 4.8 <i>Use Case</i> Skenario Lihat Profil.....	33
Tabel 4.9 <i>Use Case</i> Skenario Tambah Post.....	34
Tabel 4.10 <i>Use Case</i> Skenario Ambil Foto Dari Kamera.....	34
Tabel 4.11 <i>Use Case</i> Skenario Ambil Foto Dari Galeri.....	34
Tabel 4.12 <i>Use Case</i> Skenario Tambah Lokasi.....	35
Tabel 4.13 <i>Use Case</i> Skenario Lihat Lokasi Terdekat.....	35
Tabel 4.14 <i>Use Case</i> Skenario Lihat Wisata Termurah.....	36
Tabel 4.15 <i>Use Case</i> Skenario Lihat Wisata Termahal.....	36
Tabel 4.16 <i>Use Case</i> Skenario Lihat Agenda.....	36
Tabel 4.17 <i>Use Case</i> Skenario Tambah Agenda.....	37
Tabel 4.18 <i>Use Case</i> Skenario Edit Agenda.....	37
Tabel 4.19 <i>Use Case</i> Skenario Hapus Agenda.....	38
Tabel 4.20 <i>Use Case</i> Skenario Cari Pengguna.....	38
Tabel 4.21 <i>Use Case</i> Skenario Lihat Profil Pengguna Lain.....	38
Tabel 4.22 <i>Use Case</i> Skenario Ikuti Pengguna.....	39
Tabel 4.23 <i>Use Case</i> Skenario Berhenti Mengikuti Pengguna.....	39
Tabel 4.24 <i>Use Case</i> Skenario Cari Lokasi.....	40
Tabel 4.25 <i>Use Case</i> Skenario Lihat Post Berdasarkan Lokasi.....	40
Tabel 4.26 <i>Use Case</i> Skenario Tambah Komentar.....	40

Tabel 4.27 <i>Use Case</i> Skenario Lihat Komentar.....	41
Tabel 4.28 <i>Use Case</i> Skenario Edit Profil.....	41
Tabel 4.29 <i>Use Case</i> Skenario Hapus Post	42
Tabel 4.30 <i>Use Case</i> Skenario Laporkan Post	42
Tabel 4.31 <i>Use Case</i> Skenario Lihat Notifikasi	42
Tabel 4.32 <i>Use Case</i> Skenario <i>Logout</i>	43
Tabel 4.33 <i>Use Case</i> Skenario Login.....	43
Tabel 4.34 <i>Use Case</i> Skenario Lihat Daftar Laporan Post	44
Tabel 4.35 <i>Use Case</i> Skenario Hapus Post Dari Daftar Laporan	44
Tabel 4.36 <i>Use Case</i> Skenario Hapus Post	44
Tabel 4.37 Kebutuhan Non-Fungsional.....	45
Tabel 5.1 Struktur Tabel User	57
Tabel 5.2 Struktur Tabel Komentar.....	57
Tabel 5.3 Struktur Tabel Lokasi.....	58
Tabel 5.4 Struktur Tabel Pos	58
Tabel 5.5 Struktur Tabel Follow	59
Tabel 5.6 Struktur Tabel Agenda.....	59
Tabel 5.7 Struktur Tabel Pos_temp.....	59
Tabel 5.8 Struktur Tabel Notif.....	60
Tabel 5.9 Penjelasan Antarmuka Post Terbaru.....	61
Tabel 5.10 Penjelasan Antarmuka Lokasi Terdekat	62
Tabel 5.11 Penjelasan Antarmuka Daftar Agenda	63
Tabel 5.12 Penjelasan Antarmuka Daftar Post Berdasarkan Lokasi	64
Tabel 5.13 Penjelasan Antarmuka Daftar Laporan	65
Tabel 6.1 Pengujian Unit Klas Agenda Operasi kirimagenda().....	73
Tabel 6.2 Pengujian Unit Klas Pos Operasi kirimpos()	76
Tabel 6.3 Pengujian Unit Klas Admin Operasi blokir()	77
Tabel 6.4 Kasus Uji Berhasil Register	78
Tabel 6.5 Kasus Uji Gagal Register 1	78
Tabel 6.6 Kasus Uji Gagal Register 2	79
Tabel 6.7 Kasus Uji Berhasil <i>Login</i>	79
Tabel 6.8 Kasus Uji Gagal <i>Login</i> 1.....	79



Tabel 6.9 Kasus Uji Gagal <i>Login</i> 2.....	80
Tabel 6.10 Kasus Uji Lihat Post Terbaru.....	80
Tabel 6.11 Kasus Uji Lihat Profil.....	81
Tabel 6.12 Kasus Uji Berhasil Tambah Post.....	81
Tabel 6.13 Kasus Uji Gagal Tambah Post.....	81
Tabel 6.14 Kasus Uji Ambil Foto dari Kamera.....	82
Tabel 6.15 Kasus Uji Ambil Foto dari Galeri.....	82
Tabel 6.16 Kasus Uji Tambah Lokasi.....	82
Tabel 6.17 Kasus Uji Berhasil Lihat Lokasi Terdekat.....	83
Tabel 6.18 Kasus Uji Gagal Lihat Lokasi Terdekat.....	83
Tabel 6.19 Kasus Uji Lihat Wisata Termurah.....	84
Tabel 6.20 Kasus Uji Lihat Wisata Termahal.....	84
Tabel 6.21 Kasus Uji Lihat Agenda.....	84
Tabel 6.22 Kasus Uji Berhasil Tambah Agenda.....	85
Tabel 6.23 Kasus Uji Gagal Tambah Agenda.....	85
Tabel 6.24 Kasus Uji Edit Agenda.....	86
Tabel 6.25 Kasus Uji Hapus Agenda.....	86
Tabel 6.26 Kasus Uji Berhasil Cari Pengguna.....	87
Tabel 6.27 Kasus Uji Gagal Cari Pengguna 1.....	87
Tabel 6.28 Kasus Uji Gagal Cari Pengguna 2.....	87
Tabel 6.29 Kasus Uji Lihat Profil Pengguna Lain.....	88
Tabel 6.30 Kasus Uji Ikuti Pengguna.....	88
Tabel 6.31 Kasus Uji Berhenti Mengikuti Pengguna.....	88
Tabel 6.32 Kasus Uji Berhasil Cari Lokasi.....	89
Tabel 6.33 Kasus Uji Gagal Cari Lokasi 1.....	89
Tabel 6.34 Kasus Uji Gagal Cari Lokasi 2.....	89
Tabel 6.35 Kasus Uji Lihat Post Berdasarkan Lokasi.....	90
Tabel 6.36 Kasus Uji Tambah Komentar.....	90
Tabel 6.37 Kasus Uji Lihat Komentar.....	91
Tabel 6.38 Kasus Uji Edit Profil.....	91
Tabel 6.39 Kasus Uji Hapus Post.....	91
Tabel 6.40 Kasus Uji Laporkan Post.....	92



Tabel 6.41 Kasus Uji Lihat Notifikasi 92

Tabel 6.42 Kasus Uji Logout Pengguna 92

Tabel 6.43 Kasus Uji Logout Admin..... 93

Tabel 6.44 Kasus Uji Berhasil *Login* admin..... 93

Tabel 6.45 Kasus Uji Gagal *Login* Admin 1 93

Tabel 6.46 Kasus Uji Gagal *Login* Admin 2 94

Tabel 6.47 Kasus Uji Lihat Daftar Laporan Post 94

Tabel 6.48 Kasus Uji Hapus Post dari Daftar Laporan 94

Tabel 6.49 Kasus Uji Hapus Post 95

Tabel 6.50 Karakteristik Pengujian *Compatibility* 97



DAFTAR GAMBAR

Gambar 2.1 Diagram Alir Model <i>Waterfall</i>	13
Gambar 3.1 Diagram Alir Metodologi Penelitian	17
Gambar 4.1 Use Case Diagram Pengguna	31
Gambar 4.2 Use Case Diagram Admin	31
Gambar 5.1 Perancangan Arsitektur Sistem	46
Gambar 5.2 <i>Sequence Diagram</i> Lihat Lokasi Terdekat	47
Gambar 5.3 <i>Sequence Diagram</i> Lihat Agenda	47
Gambar 5.4 <i>Sequence Diagram</i> Tambah Agenda	48
Gambar 5.5 <i>Sequence Diagram</i> Hapus Laporan	48
Gambar 5.6 <i>Sequence Diagram</i> Tambah Post	49
Gambar 5.7 <i>Class Diagram</i> Aplikasi Pengguna	50
Gambar 5.8 <i>Class Diagram</i> Aplikasi Admin	50
Gambar 5.9 Entity Relationship Diagram	56
Gambar 5.10 Perancangan Antarmuka Post Terbaru	60
Gambar 5.11 Perancangan Antarmuka Lokasi Terdekat	62
Gambar 5.12 Perancangan Antarmuka Daftar Agenda	63
Gambar 5.13 Perancangan Antarmuka Daftar Post Berdasarkan Lokasi	64
Gambar 5.14 Perancangan Antarmuka Daftar Laporan	65
Gambar 5.15 Implementasi <i>Database</i>	68
Gambar 5.16 Implementasi Antarmuka Post Terbaru	69
Gambar 5.17 Implementasi Antarmuka Lokasi Terdekat	69
Gambar 5.18 Implementasi Antarmuka Daftar Agenda	70
Gambar 5.20 Implementasi Antarmuka Daftar Laporan	71
Gambar 6.1 Tampilan Aplikasi pada Android Versi 4.2 (<i>Jelly Bean</i>)	95
Gambar 6.2 Tampilan Aplikasi pada Android Versi 7.1 (<i>Nougat</i>)	96
Gambar 6.3 Tampilan Aplikasi pada Android Versi 8.0 (<i>Oreo</i>)	96



BAB 1 PENDAHULUAN

1.1 Latar belakang

Kemajuan teknologi di bidang informasi saat ini berkembang dengan sangat cepat. Kemajuan tersebut memudahkan seseorang untuk mengirim atau pun menerima informasi. Proses komunikasi dapat dilakukan dengan mudah, cepat, dan dimana saja.

Perkembangan teknologi informasi telah berdampak terhadap aktivitas *traveling*. Perkembangan akitivitas *traveling* per tahun dewasa ini terus meningkat secara signifikan. Tingginya tingkat aktivitas *traveling* membuat aktivitas tersebut menjadi sebuah *trend* di bebagai kalangan. *Traveling* sendiri sudah menjadi gaya hidup seseorang untuk mengisi waktu luang ataupun sekedar hobi. Dengan perkembangan teknologi informasi membuat aktivitas *traveling* menjadi semakin mudah dalam mencari informasi yang dibutuhkan (Sulaiman, 2016).

Aktivitas pariwisata tidak lepas dari dokumentasi sebagai suatu sarana untuk mengenang peristiwa wisata. Foto atau gambar merupakan produk dari aktivitas fotografi. Perkembangan teknologi *mobile* saat ini telah dilengkapi dengan kamera digital di dalamnya sehingga dapat digunakan secara mudah untuk mengambil foto wisata bagi para pengunjung wisata. Dengan dukungan jaringan internet, memungkinkan pula bagi wisatawan untuk mendistribusikan foto agar bisa dinikmati oleh banyak orang dalam waktu singkat melalui perantara media sosial (Antopani, 2015).

Perangkat *mobile* berupa *smartphone* saat ini sudah mendukung kamera yang memiliki fitur *geotagging*. *Geotagging* adalah sebuah proses penambahan informasi posisi data pada GPS berupa informasi *latitude* dan *longitude* dalam sebuah foto *digital*. Dengan adanya fitur *geotagging* dalam informasi sebuah foto maka letak pengambilan foto tersebut dapat dengan mudah diketahui. Penerapan fitur *geotagging* telah banyak dimanfaatkan dalam berbagai penelitian seperti pada penelitian yang berjudul Rancang Bangun Aplikasi Sosial *Geotagging* Keindahan Alam Indonesia pada Sistem Operasi Android (Efendy, 2016). Pemanfaatan *geotagging* pada penelitian tersebut digunakan untuk memberikan informasi lokasi pada foto keragaman budaya yang dibagikan oleh pengguna.

Salah satu hasil kemajuan teknologi dalam berkomunikasi adalah media sosial. Pada tahun 2016, 68.3% pengguna *internet* di dunia merupakan pengguna media sosial yang kurang lebih setara dengan 2.34 miliar orang (Statista, 2016). Di Indonesia, pengguna *internet* sebesar 63 juta orang dan 95% dari angka tersebut menggunakan internet untuk mengakses media sosial yang kurang lebih setara dengan 59 juta orang (KOMINFO, 2013).

ITB World Travel Trend Report Tahun 2015/2016 mengungkapkan bahwa jumlah wisatawan yang merupakan pengguna Facebook aktif mencapai sekitar

1,5 miliar orang di seluruh dunia sementara platform sosial lainnya seperti Twitter, LinkedIn, dan Google+ terus tumbuh di berbagai negara. Oleh karena itu, media sosial sangat populer di kalangan wisatawan, dan sekitar 70% dari mereka adalah pengguna aktif media sosial. Laporan ini juga mengungkapkan bahwa 30 persen wisatawan internasional yang menggunakan internet mendapatkan informasi mengenai perjalanan wisata mereka melalui media sosial (Made, et al., 2016).

Sebanyak 53% konten yang di *posting* oleh pengguna di Indonesia pada media sosial instagram adalah foto *traveling* (Parikesit, 2016). Foto-foto tempat wisata yang diunggah di *Instagram* dapat mempengaruhi pengguna lainnya dan menimbulkan respon bagi pengguna Instagram lain. Kecenderungan pengguna untuk membagikan dan mencari informasi dengan menggunakan akun berbasis foto memudahkan para *traveler* untuk menentukan tujuan wisata. Foto yang di *upload* mampu memberikan gambaran yang *real* dengan kondisi nyata sesuai dengan kondisi asli (Baskara, et al., 2017).

Pengaruh yang ditimbulkan yaitu munculnya berbagai referensi yang berserakan pada media cetak dan elektronik. Media-media tersebut kurang efektif dalam memberikan informasi, hal tersebut dikarenakan penelusuran informasi dari sumber yang tidak tertata sehingga membutuhkan waktu yang lama untuk mencarinya (Mulyatiningsih, et al., 2014). Selain itu pengguna masih kesulitan untuk mendapatkan informasi wisata sesuai dengan kriteria yang diinginkan ketika tidak mengetahui kata kunci yang tepat untuk mencari informasi yang dimaksud (Trihatmaja, 2015). Sehingga agar informasi tersebut dapat dengan mudah diperoleh sebaiknya referensi wisata dikumpulkan dalam suatu wadah yang mudah di akses.

Dalam penelitian yang dilakukan oleh Analis Eryani dengan judul “Faktor-Faktor Pengambilan Keputusan Wisatawan Untuk Berkunjung Ke Objek Wisata Istana Siak Sri Indrapura Kabupaten Siak Provinsi Riau” dijelaskan berbagai faktor yang dapat mempengaruhi seseorang untuk berwisata, salah satu faktor yang berperan adalah gambaran perjalanan yang meliputi jarak, waktu, dan juga biaya. Dari 330 responden terdapat 30 orang yang merasa sangat setuju jika jarak merupakan faktor pendukung untuk mempengaruhi seseorang dalam melakukan perjalanan wisata, 216 orang merasa setuju dengan hal tersebut. Untuk faktor waktu dan juga biaya terdapat 45 orang yang merasa sangat setuju dan 348 orang merasa setuju dari 456 responden. Hal ini menunjukkan bahwa jarak orang yang akan melakukan wisata dengan tempat wisata tersebut, waktu dan biaya yang dibutuhkan menjadi pertimbangan untuk berwisata.

Keberadaan suatu komunitas memungkinkan anggotanya untuk saling berinteraksi satu sama lain sehingga menyebabkan adanya proses berbagi informasi (Trihatmaja, 2009). Dengan adanya komunitas wisata dapat menjadi wadah tersendiri bagi orang-orang yang antusias di bidang wisata baik dalam berbagi atau mencari informasi.

Berdasarkan latar belakang diatas maka penulis mengangkat skripsi dengan judul “Pembangunan Aplikasi *Social Geotagging* Destinasi wisata” dengan memanfaatkan fitur *geotagging* untuk penyampaian informasi destinasi wisata dalam bentuk sosial media yang dapat digunakan sebagai wadah berbagi informasi wisata sehingga pengguna nantinya dapat saling berbagi informasi dan membantu pengguna lain menemukan lokasi destinasi wisata terdekat dalam radius tertentu, dapat menampilkan harga, dan juga membantu pengguna membuat agenda wisata.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah diuraikan, dapat dirumuskan masalah-masalah yang akan dibahas sebagai berikut:

1. Bagaimana analisis dan perancangan aplikasi *social geotagging* destinasi wisata yang melibatkan pengguna sebagai kontributor?
2. Bagaimana hasil implementasi aplikasi *social geotagging* destinasi wisata yang memudahkan seseorang memperoleh dan berbagi informasi destinasi wisata?
3. Bagaimana hasil pengujian aplikasi *social geotagging* destinasi wisata ditinjau dari pengujian *white box* dan pengujian *black box*?

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah membuat aplikasi *social geotagging* destinasi wisata sebagai wadah bagi penggunanya untuk saling berbagi dan mendapatkan informasi seputar destinasi wisata yang disertai dengan dokumentasi foto yang di bagikan oleh pengguna lainnya.

1.4 Manfaat

Manfaat dari pembangunan aplikasi *social geotagging* destinasi wisata ini adalah sebagai berikut:

1. Masyarakat dapat saling berbagi informasi seputar tempat wisata yang mereka kunjungi.
2. Masyarakat diharapkan dapat berperan aktif serta berpartisipasi dalam membantu pemerintah terutama pada bidang pariwisata dengan membagikan informasi destinasi wisata.

1.5 Batasan masalah

Berdasarkan rumusan masalah yang telah dipaparkan diatas, maka penulis memberikan batasan masalah dari penelitian ini hanya pada:

1. Sistem yang dikembangkan berjalan pada platform Android.
2. Data peta digital diperoleh berdasarkan data yang telah disediakan Google Map.
3. Kondisi internet dan GPS dianggap telah aktif.
4. Informasi yang disediakan oleh aplikasi dikhususkan pada informasi terkait tempat wisata di Indonesia.
5. Aplikasi yang akan dibangun membutuhkan koneksi internet untuk dapat dijalankan
6. Aplikasi dibangun dengan pendekatan *web hybrid* menggunakan bahasa pemrograman CSS, HTML, JavaScript, dan framework *PhoneGap* untuk mengubah menjadi aplikasi Android.

1.6 Sistematika pembahasan

Bagian ini berisi struktur skripsi ini yang dimulai dari bab pendahuluan sampai dengan bab penutup dan dijelaskan deskripsi singkat dari setiap bab. Bagian ini diharapkan dapat membantu pembaca untuk memahami sistematika pembahasan isi pada skripsi ini.

Dalam penulisan penelitian ini, sistematika penulisannya adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisikan uraian dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, sistematika penulisan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisikan kajian terhadap beberapa teori dan referensi yang menjadi landasan dalam mendukung studi penelitian ini.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi penjelasan mengenai metode yang digunakan di dalam penelitian serta menjelaskan studi literatur, analisis kebutuhan sistem yang akan dibuat, perancangan sistem, implementasi sistem, pengujian sistem serta kesimpulan dan saran.

BAB 4 ANALISIS KEBUTUHAN

Dalam bab ini berisikan pembahasan mengenai proses analisis kebutuhan aplikasi *social geotagging* destinasi wisata.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Dalam bab ini berisikan pembahasan mengenai proses perancangan dan implementasi aplikasi *social geotagging* destinasi wisata.

BAB 6 PENGUJIAN DAN ANALISIS

Dalam bab ini menjelaskan proses pengujian aplikasi *social geotagging* destinasi wisata yang telah dibuat.

BAB 7 PENUTUP

Berisi kesimpulan yang diperoleh dari pengujian sistem berdasarkan hasil implementasi sistem serta saran-saran untuk pengembangan sistem selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

2.1.1 Penelitian Aplikasi *Geotagging*

Terdapat beberapa penelitian sebelumnya yang memanfaatkan penggunaan *geotagging* dalam membuat sebuah aplikasi yang menjadi referensi dalam penelitian ini

Arik Achmad Efendy (2016), dengan judul Rancang Bangun Aplikasi Sosial *Geotagging* Keindahan Alam Indonesia pada Sistem Operasi Android. Permasalahan pada penelitian ini terkait dengan masih banyaknya wilayah Indonesia yang masih berpotensi menjadi objek wisata yang belum di kembangkan karena kurangnya informasi. Hasil dari penelitian ini adalah pemanfaatan fitur *geotagging* untuk membangun sebuah aplikasi berbasis Android yang melibatkan pengguna untuk berbagi informasi dalam bentuk foto seputar wilayah yang berpotensi sebagai objek wisata sehingga dapat diketahui posisinya oleh masyarakat lainya.

Miftahul Ulum Mubiin (2016), dengan judul Rancang Bangun Aplikasi Mobile *Social Geotagging* Transportasi Publik pada Platform Android. Permasalahan pada penelitian ini terkait dengan aktivitas penyampaian keluhan masyarakat tentang transportasi umum yang sebelumnya tidak ada wadah yang memfasilitasinya. Peneliti menerapkan metode *social report* dengan menggunakan fitur *geotagging* untuk mendapatkan data posisisi dari keluhan pengguna aplikasi. Hasil dari penelitian ini adalah sebuah sistem berbasis Android yang dapat menampung keluhan-keluhan masyarakat terkait dengan transportasi umum dan memudahkan pemerintah untuk memetakan permasalahan transportasi umum sehingga dapat mewujudkan pembangunan dengan meningkatkan sarana dan prasarana transportasi publik yang ada dengan memanfaatkan fitur *geotagging*.

Tabel 2.1 Penelitian Aplikasi *Geotagging*

No	Judul	Penulis	Perbandingan	
			Kajian Pustaka	Skripsi Penulis
1	Rancang Bangun Aplikasi Sosial <i>Geotagging</i> Keindahan Alam Indonesia pada Sistem Operasi Android.	(Efendy, 2016)	Implementasi fitur <i>geotagging</i> untuk melaporkan tempat yang berpotensi sebagai tempat wisata	Implementasi fitur <i>geotagging</i> untuk berbagi tempat wisata dalam bentuk sosial media
2	Rancang Bangun	(Mubiin,	Implementasi	Implementasi



	Aplikasi Mobile <i>Social Geotagging</i> Transportasi Publik pada Platform Android	2016)	metode <i>social report</i> dengan menggunakan fitur <i>geotagging</i> untuk mendapatkan data posisi dari keluhan pengguna aplikasi.	fitur <i>geotagging</i> untuk mendapatkan posisi dokumentasi wisata pengguna.
--	--	-------	--	---

2.1.2 Penelitian Aplikasi Pencarian Lokasi Terdekat

Terdapat penelitian yang mengangkat masalah pencarian lokasi terdekat yang menjadi referensi sebagai bahan perbandingan dalam penelitian ini.

Yustika Rillaningrum (2014) dengan judul Rancang Bangun Aplikasi Android untuk Pencarian Lokasi ATM Terdekat di Kota Malang. Permasalahan pada penelitian ini terkait dengan banyaknya mesin ATM yang tersebar di berbagai sudut kota malang, namun terkadang masyarakat masih merasa kesulitan untuk menemukan mesin ATM yang dibutuhkan. Peneliti menerapkan metode Haversine untuk membuat fitur pencarian lokasi ATM terdekat. Metode Haversine digunakan untuk menghitung jarak antara pengguna dengan lokasi ATM. Hasil dari penelitian tersebut adalah sebuah sistem berbasis Android yang mampu memberikan rekomendasi mesin ATM terdekat dari pengguna dengan memanfaatkan metode Haversine dalam menghitung jarak antara pengguna dengan mesin ATM.

Tabel 2.2 Penelitian Aplikasi Lokasi Terdekat

No	Judul	Penulis	Perbandingan	
			Kajian Pustaka	Skripsi Penulis
1	Rancang Bangun Aplikasi Android untuk Pencarian Lokasi ATM Terdekat di Kota Malang	(Rillaningrum, 2014)	Implementasi metode Haversine untuk mencari lokasi mesin ATM terdekat	Implementasi metode Haversine untuk mencari lokasi wisata terdekat

2.2 Wisata

Dalam Undang-Undang Republik Indonesia No.10 tahun 2009 menerangkan bahwa Wisata adalah kegiatan perjalanan yang dilakukan oleh seseorang atau sekelompok orang dengan mengunjungi tempat tertentu untuk tujuan rekreasi,



pengembangan pribadi, atau mempelajari keunikan daya tarik wisata yang dikunjungi dalam jangka waktu sementara.

Daerah tujuan pariwisata yang selanjutnya disebut Destinasi Pariwisata adalah kawasan geografis yang berada dalam satu atau lebih wilayah administratif yang di dalamnya terdapat daya tarik wisata, fasilitas umum, fasilitas pariwisata, aksesibilitas, serta masyarakat yang saling terkait dan melengkapi terwujudnya kepariwisataan.

2.3 Geotagging

Geotagging adalah sebuah proses penambahan informasi berupa posisi data pada GPS terkait *latitude* dan *longitude* pada sebuah foto *digital*. Dengan adanya fitur *geotagging* dalam informasi sebuah foto maka letak pengambilan foto tersebut dapat dengan mudah diketahui. Terdapat tiga metode yang dapat dilakukan untuk melakukan *geotagging* pada suatu media, antara lain (Nandipati, 2011):

1. *Geocoding* manual, merupakan metode dimana informasi mengenai lokasi ditambahkan secara manual dengan menginputkan koordinat tertentu atau memilih lokasi pada saat melakukan upload suatu media ke dalam internet. Tingkat akurasi dari metode *geotagging* ini tergantung pada tools yang digunakan ataupun GPS receiver untuk mendapatkan koordinat.
2. Kamera digital dengan fitur GPS, pada beberapa kamera digital telah dilengkapi dengan fitur GPS secara langsung. Hasil foto maupun video akan secara otomatis memiliki informasi lokasi pada EXIF header dari data tersebut.
3. Kamera digital dengan GPS terpisah, pada kamera digital yang belum dilengkapi dengan fitur GPS *geotagging* dilakukan dengan menyingkronkan kamera dan GPS melalui sebuah *software*. Koordinat yang terekam oleh GPS pada waktu yang bersamaan dengan waktu pengambilan foto atau video akan ditambahkan sebagai informasi lokasi pada EXIF foto.

Mekanisme *geotagging* pada foto dimulai ketika foto diambil menggunakan kamera yang telah memiliki fitur *geotagging*, kamera tersebut dapat mencatat informasi yang lebih banyak dibandingkan dengan sebuah foto yang diambil dengan kamera biasa. Informasi yang dapat dicatat diantaranya waktu dan lokasi dimana foto diambil, orientasi dari kamera (*portrait* atau *landscape*), apakah menggunakan lampu *flash* dan detil kamera lainnya yang digunakan seperti *Apertur*, *Exposure*, dan *Local Length*. Semua data ini disimpan pada suatu tempat yang disebut *EXIF Headers*.

Format Exchangeable Image File format (EXIF) adalah spesifikasi industri yang diterbitkan untuk format file gambar yang digunakan oleh kamera digital. *Exchangeable Image File Format (EXIF) headers* dapat berisi informasi mengenai foto dengan jenis data yang dapat dibaca oleh perangkat lunak manajemen foto.

2.4 Android

Android merupakan salah satu sistem operasi *mobile* yang berkembang diantara sistem operasi *mobile* lainnya seperti iOS, *Windows Mobile*, dan masih banyak lagi yang menawarkan kekayaan isi dan keoptimalan ketika berjalan diatas perangkat *hardware* yang ada. Beberapa OS yang ada saat ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan dalam distribusi aplikasi pihak ketiga untuk *platform* tersebut (Rillaningrum, 2014).

Android berkembang pesat dari segi teknologi dan jumlah *device* yang ada dikarenakan beberapa faktor seperti kelengkapan aplikasi, kelengkapan *tool* pengembangan, serta dukungan yang sangat tinggi dari komunitas *open source* di dunia. Selain itu, Android juga dipuji sebagai *platform mobile* pertama yang lengkap, terbuka, dan bebas (Mulyadi, 2010)

1. Lengkap (*Complete Platform*) : Para *designer* dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan aplikasi pada platform Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan adanya peluang untuk pengembangan aplikasi.
2. Terbuka (*Open Source Platform*) : *Platform* Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri mengembangkan Linux Kernel 2.6.
3. Free (*Free Platform*) : Android adalah platform atau aplikasi yang bebas untuk develop. Tidak ada lisensi atau biaya royalti untuk pengembangan pada platform Android. Aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apa pun.

2.5 Google Maps API

Google Maps *Application Programming Interface* (API) adalah sebuah bahasa pemrograman yang disediakan oleh google maps. Google Maps API terdiri dari *Javascript* dan HTML. Dengan adanya google maps API, website lain dapat menampilkan google maps didalamnya (Sholeh, et al., 2011).

Google Maps API merupakan fitur yang di sediakan oleh google untuk memberikan fasilitas kepada pengguna yang akan mengintergrasikan Google Maps ke dalam *website* yang ingin di kembangkan dengan menampilkan *data point* milik sendiri dengan menggunakan API *key*.

API *key* adalah kode unik yang digenerasikan oleh Google untuk suatu *website* tertentu. Fungsi dari API *key* adalah agar server dari Google Maps dapat mengenali website tersebut. Pada Google Maps API versi 3 sudah tidak

membutuhkan API key lagi, sehingga mempermudah penggunaan dari Google Maps API (Sholeh, et al., 2011).

2.6 Rumus Haversine

Rumus haversine merupakan metode yang digunakan untuk menghitung jarak diantara dua titik yang ada pada permukaan bumi berdasarkan koordinat *latitude* dan *longitude*. Rumus ini biasanya digunakan untuk melakukan perhitungan untuk memprediksi jarak antara lokasi titik penting untuk mengatur waktu dan mengatur rencana perjalanan. Penggunaan rumus ini mengabaikan efek elipsoidal sehingga mengasumsikan bahwa bumi berbentuk bulat (Essayad, 2011). Rumus haversine dituliskan dalam persamaan 2.1

$$\begin{aligned} R &= 6371 \text{ km} \\ \Delta t &= t_2 - t_1 \\ \Delta l &= l_2 - l_1 \\ a &= \sin^2\left(\frac{\Delta t}{2}\right) + \cos(t_1) \times \cos(t_2) \times \sin^2\left(\frac{\Delta l}{2}\right) \\ c &= 2 \times \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \\ d &= R \cdot c \end{aligned} \tag{2.1}$$

Keterangan :

- R = Radius Bumi
- t_1 = *latitude* lokasi 1
- t_2 = *latitude* lokasi 2
- l_1 = *longitude* lokasi 1
- l_2 = *longitude* lokasi 2
- d = jarak (km)

Rumus perhitungan jarak antara dua lokasi yang berbeda ini dapat diterapkan dengan mengkonversikan nilai *latitude* dan *longitude* dari derajat ke radian dengan mengalikan $\pi/180$. (Essayad, 2011)

2.7 PhoneGap

PhoneGap merupakan *framework open source* yang digunakan untuk membuat aplikasi di berbagai macam platform seperti Blackberry, Android,

Windows Phone, atau Iphone dimana bahan baku dari aplikasi yang akan dibuat dengan PhoneGap adalah sebuah *file web* (EMS, 2013).

PhoneGap memungkinkan pembangun aplikasi untuk membuat suatu aplikasi *native* pada perangkat *mobile*. *User Interface* dalam aplikasi yang dibangun dengan PhoneGap dibuat dengan CSS, HTML dan JavaScripts. *UI layer* pada aplikasi PhoneGap merupakan suatu tampilan web browser yang dapat memenuhi 100% lebar layar dan 100% tinggi layar. Beberapa fitur yang dapat diberikan oleh PhoneGap dijelaskan pada Tabel 2.3

Tabel 2.3 Fitur PhoneGap

	iPhone/ iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 6.0+	WebOS	Windows Phone 7+ 8	Symbian	Bada
Accelerometer	√	√	√	√	√	√	√	√
Camera	√	√	√	√	√	√	√	√
Compass		√	√	X	√	√	X	√
Contact	√	√	√	√	X	√	√	√
File	√	√	√	√	X	√	X	X
Geolocation	√	√	√	√	√	√	√	√
Media	√	√	√	X	X	√	X	X
Network	√	√	√	√	√	√	√	√
Notification (Alert)		√	√	√	√	√	√	√
Notification (Sound)			√	√	√	√	√	√
Notification (Vibration)	√	√	√	√	√	√	√	√
Storage	√	√	√	√	√	√	X	X

Keterangan:

√ - supported feature

X – unsupported feature due to hardware or software restrictions

2.8 CodeIgniter

CodeIgniter merupakan *framework open source* PHP yang mudah dikuasai dan powerful. *Framework* ini dibangun untuk pengguna bahasa pemrograman PHP yang membutuhkan toolkit yang sederhana dan baik untuk membuat *full featured web applications*. CodeIgniter merupakan framework yang menerapkan konsep MVC (*Model View Controller*) yang di desain untuk memudahkan penggunaannya ketika membangun aplikasi berbasis web yang dinamis (Blanco, 2009).

Model terdiri dari *class* yang memiliki fungsi mengubah, menghapus, dan mengambil data dari basis data. *Controller* terdiri dari *class* yang memiliki fungsi sebagai penghubung antara *Model* dan *View*. *Controller* memiliki fungsi untuk memanggil fungsi-fungsi yang terdapat pada Model serta mengambil data yang

telah diambil melalui Model dari basis data. Selanjutnya data ditampilkan melalui *View*. Codeigniter dapat mempercepat pengembangan sebuah aplikasi web karena bersifat cepat, ringan, serta memiliki dokumentasi yang lengkap. Selain itu, Codeigniter juga menyediakan contoh implementasi kode yang dapat digunakan sebagai referensi. Karena Kelebihan-kelebihan tersebut, maka banyak memilih menggunakan Codeigniter dalam mengembangkan sebuah aplikasi web. (Voldemord, 2015).

2.9 Pengembangan Perangkat Lunak

Terdapat beberapa pembahasan mengenai pengembangan perangkat lunak yaitu model pengembangan perangkat lunak dan model *waterfall*.

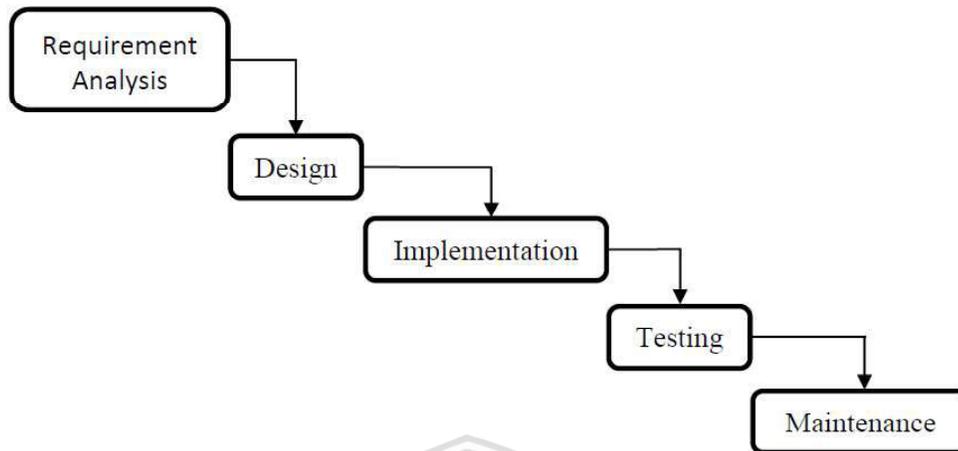
2.9.1 Model Pengembangan Perangkat Lunak

Terdapat banyak sekali jenis dari model proses pengembangan perangkat lunak yang diantaranya mencakup metodologi konvensional dan inovatif seperti *waterfall*, *prototyping*, *spiral development*, *iterative and incremental development*, *extreme programming*, dan *rapid application development*. Proses pengembangan perangkat lunak terbagi ke dalam beberapa tahapan, yaitu (Caytiles,2014):

1. Tahap Analisis
2. Tahap Kebutuhan
3. Tahap Desain
4. Tahap Implementasi (*coding*)
5. Pengujian
6. *Deployment*
7. Pemeliharaan dan perbaikan *bug*

2.9.2 Model *Waterfall*

Model SDLC *waterfall* merupakan salah satu model SDLC klasik yang menyediakan pendekatan alur hidup dari perangkat lunak secara runtut atau sekuensial yang dimulai dari tahap analisis, tahap desain, tahap pengkodean, tahap pengujian serta tahap pendukung (*support*) (S & Shalahuddin, 2015). Dalam model *waterfall* adanya dokumentasi serta perencanaan yang baik akan membuat proses pengembangan sistem berjalan dengan baik. Pada Gambar 2.1 akan digambarkan diagram alir *waterfall model*.



Gambar 2.1 Diagram Alir Model Waterfall

1. Requirement Analysis

Requirement analysis dilakukan sebagai tahap awal pada proses pengembangan suatu perangkat lunak. Tujuan dari tahap ini adalah untuk memahami kebutuhan dari klien dan mendokumentasikannya dengan benar. Penekanan pada tahap *Requirement analysis* adalah untuk mengidentifikasi apa saja yang dibutuhkan dari sebuah sistem. Output yang dihasilkan dari tahap ini adalah *Software Requirement Specification (SRS)* (Kumar, 2013).

2. Design

Untuk berpindah dari domain masalah ke dalam domain solusi, maka dilakukan tahap Design. Output yang dihasilkan dari tahap ini adalah Software Design Document (SDD) (Kumar, 2013). Tahap *design* menjelaskan bagaimana kebutuhan yang sudah diidentifikasi akan diimplementasikan (Bindal, 2015).

3. Implementation

Implementation merupakan tahap untuk mengubah SDD ke dalam bentuk kode pemrograman dengan menggunakan beberapa bahasa pemrograman. Tahap ini merupakan fase logika dari proses pengembangan perangkat lunak. Output yang dihasilkan dari tahap ini adalah kode program (Kumar, 2013).

4. Testing

Testing atau pengujian merupakan tahap untuk menguji semua unit di dalam sistem untuk memvalidasi operasi dan spesifikasi sistem (Bindal, 2015). Pengujian yang efektif dapat memberikan kontribusi dalam pembangunan suatu produk perangkat lunak yang berkualitas tinggi, biaya pemeliharaan yang lebih rendah, kepuasan pengguna, dan menghasilkan produk yang lebih akurat serta dapat diandalkan (Kumar, 2013).

5. Maintenance

Maintenance merupakan tahap yang dilakukan ketika terjadi kesalahan atau adanya modifikasi yang perlu dilakukan pada sistem yang telah diterapkan (Kumar, 2013).

2.10 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan sebuah standarisasi bahasa pemodelan dalam pengembangan perangkat lunak yang menerapkan pemrograman berorientasi objek (S & Shalahuddin, 2016). UML digunakan untuk menggambarkan pemodelan visual dari sebuah perangkat lunak. Terdapat tiga diagram UML yang akan digunakan pada proses pembangunan sistem dalam penelitian ini, yaitu (Rumbaugh, 2005):

1. Use Case Diagram

Use case diagram merupakan diagram UML yang menggambarkan perilaku dari sistem, subsistem, klas, atau komponen yang tampak dari luar dan merupakan bagian dari fungsional sistem. Sebuah use case menggambarkan sebuah interaksi antara aktor dengan sistem.

2. Sequence Diagram

Sequence diagram merupakan diagram UML yang menggambarkan interaksi yang terjadi diantara objek di dalam atau pun di sekitar sistem yang berupa message yang digambarkan dalam urutan waktu. *Sequence diagram* terdiri dari dimensi horizontal (objek-objek yang terkait) dimensi vertikal (waktu). *Sequence diagram* digunakan untuk menggambarkan suatu rangkaian langkah-langkah atau skenario yang dilakukan sebagai respon dari sebuah event yang bertujuan untuk menghasilkan output tertentu.

3. Class Diagram

Class diagram merupakan diagram UML yang menggambarkan struktur serta deskripsi klas, *package* dan objek dan hubungan satu sama lain seperti asosiasi, pewarisan, dan lain-lain. Klas merupakan sebuah spesifikasi yang jika diinstansiasi dapat menghasilkan sebuah objek. Klas merupakan inti dari pengembangan dan perancangan berorientasi objek.

2.11 Pengujian Perangkat Lunak

Pengujian merupakan sebuah proses yang dilakukan terhadap suatu aplikasi atau program untuk menemukan kesalahan dan kemungkinan yang akan menimbulkan kesalahan sesuai dengan spesifikasi perangkat lunak yang telah ditentukan (Simarmata, 2010).

2.11.1 Pengujian *Black-Box*

Pengujian *blackbox* merupakan suatu pendekatan untuk menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian tersebut memiliki tujuan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (S & Shalahuddin, 2016).

2.11.1.1 Pengujian Validasi

Pengujian validasi adalah suatu pengujian yang termasuk ke dalam metode pengujian *blackbox*. Pengujian ini berfokus kepada kesesuaian antara kinerja dari sistem dengan kebutuhan yang telah didefinisikan.

Pengujian Validasi memiliki tujuan untuk memastikan seluruh kebutuhan yang telah di definisikan sebelumnya telah diimplemenasikan di dalam perangkat lunak. Pengujian validasi dapat dikatakan berhasil jika fungsi-fungsi pada perangkat lunak yang telah dibuat berjalan sudah sesuai dengan yang diharapkan (Pressman, 2015).

2.11.1.2 Pengujian *Compatibility*

Pengujian *compatibility* merupakan pengujian yang memiliki tujuan untuk mengetahui apakah sistem yang telah dibangun berjalan dengan baik jika dijalankan pada sistem operasi, komputer, browser, layar, dan koneksi internet yang berbeda (Pressman, 2015). Dalam penelitian ini dilakukan pengujian *compatibility* dengan menjalankan sistem pada sistem operasi Android dengan versi yang berbeda. Versi sistem operasi yang berbeda terkadang menghasilkan perbedaan pada tampilan dan terdapat masalah pada proses yang berjalan. Masalah perbedaan tampilan yang dapat terjadi diantaranya perbedaan gaya *font* atau tata letak halaman yang berubah dan menjadi sulit atau tidak terbaca.

2.11.2 Pengujian *White-Box*

Pengujian *whitebox* merupakan suatu pendekatan dengan menguji suatu perangkat lunak dari segi desain dan kode program untuk mengetahui apakah perangkat lunak yang dibuat mampu menghasilkan fungsi-fungsi dengan masukan dan keluaran yang sudah sesuai dengan spesifikasi kebutuhan. Pengujian *whitebox* dilakukan dengan cara memeriksa logika dalam kode program (S, 2015).

2.11.2.1 Pengujian *Basis Path*

Pengujian *basis path* merupakan pengujian dengan cara membuat suatu *test case* untuk mengukur kompleksitas logika dari suatu desain prosedural yang dapat digunakan sebagai pedoman dalam menetapkan jalur eksekusi program (Ladjamudin, 2006).

Metode ini dilakukan untuk menemukan kesalahan yang terdapat pada kode program dengan cara mengeksekusi semua *path* atau jalur. Jumlah jalur independen atau jalur yang dilalui suatu fungsi untuk mencapai suatu hasil yang diharapkan yang dapat digunakan untuk membuat test case merupakan hasil dari pengujian dengan menggunakan metode ini. Pengukuran kompleksitas program dalam metode ini menggunakan *cyclomatic complexity*. Berikut ini cara untuk menentukan nilai kompleksitas serta penjelasan dari nilai kompleksitas berdasarkan jumlah jalur independent. :

a. $V(G) = E - N + 2$

b. $V(G) = P + 1$

c. $V(G) = R$

Keterangan

E = *Edges*

N = *Nodes*

P = *Predicate Note* atau node yang memiliki kondisi

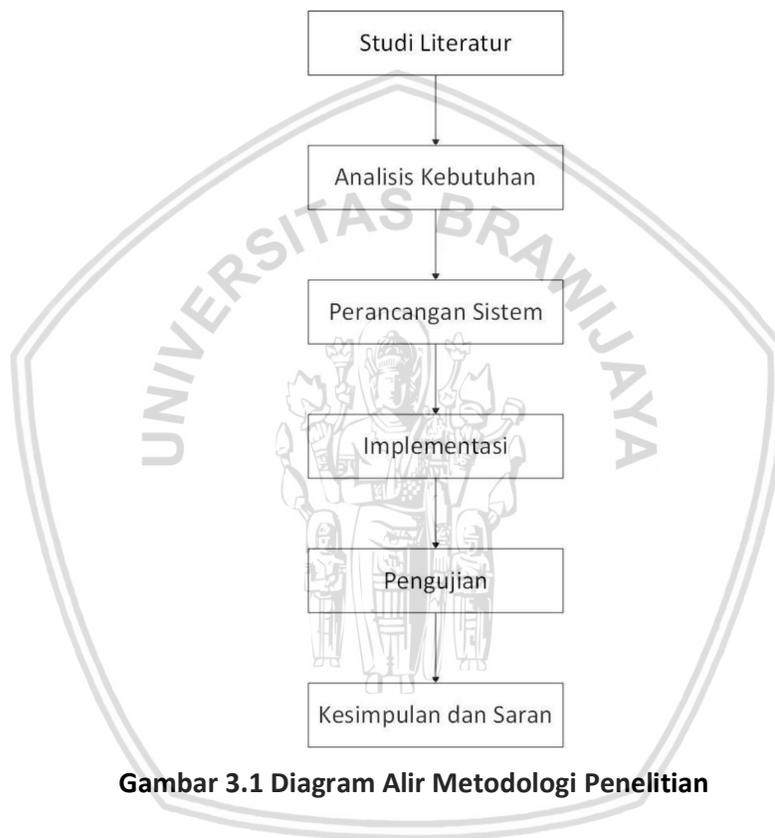
R = *Region*

Tabel 2.4 Tabel Nilai Kompleksitas

Jumlah Kompleksitas	Arti
1-10	Terstruktur dan ditulis dengan baik <i>Testability</i> tinggi Mudah dan biaya terjangkau
10-20	Kode program kompleks <i>Testability</i> sedang Tidak terlalu mudah dan biaya sedang
20-40	Kode program sangat kompleks <i>Testability</i> rendah Susah dan Mahal
>40	Usaha dan biaya sangat tinggi <i>Not testable</i>

BAB 3 METODOLOGI

Pada bab ini akan dijelaskan tahapan dalam proses penelitian. Tahap pertama yang dilakukan adalah studi literatur untuk mendapatkan teori-teori yang mendukung penelitian yang dilakukan. Kemudian melakukan analisis kebutuhan untuk mengidentifikasi aktor dan uraian sistem yang akan dibuat. Selanjutnya melakukan implementasi serta pengujian, dan penarikan kesimpulan. Adapun diagram alur metodologi penelitian tersebut ditunjukkan pada gambar 3.1 berikut:



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Literatur

Studi literatur dilakukan dengan mengumpulkan teori-teori yang terkait dengan penelitian yang dilakukan yang berasal dari paper, jurnal, buku, dan juga skripsi yang memiliki topik terkait dengan penelitian yang dilakukan. Adapun bahan yang menjadi studi literatur dalam penelitian ini adalah sebagai berikut:

1. Kajian Pustaka
2. Wisata
3. *Geotagging*
4. Android

5. Google Maps API
6. Rumus Haversine
7. PhoneGap
8. CodeIgniter
9. Pengembangan Perangkat Lunak
10. *Unified Modeling Language* (UML)
11. Pengujian Perangkat Lunak

3.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk menganalisis semua kebutuhan yang diperlukan pada perancangan perangkat lunak. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan sistem baik fungsional maupun non-fungsional dan siapa saja yang terlibat di dalam sistem. Analisis kebutuhan dilakukan dengan melakukan studi literatur terhadap beberapa penelitian sebelumnya dan observasi aplikasi yang sudah ada yang memiliki fungsi yang sama dengan aplikasi yang akan di bangun.

Analisis kebutuhan terdiri dari 4 tahap yaitu penjabaran gambaran umum aplikasi yang akan dibangun, melakukan identifikasi aktor yang terlibat, mengidentifikasi kebutuhan fungsional dan non fungsional dari aplikasi yang akan dibangun.

Proses analisis kebutuhan fungsional dilakukan dengan mengidentifikasi apa saja kebutuhan yang merupakan fungsi-fungsi yang harus dapat ditangani oleh aplikasi yang akan dibangun. Daftar kebutuhan fungsional nantinya digambarkan menggunakan *Unified Modelling Language* (UML) dengan pemodelan diagram *Use Case diagram* serta digambarkan dalam *Use Case* skenario. Proses analisis kebutuhan non-fungsional dilakukan dengan mengidentifikasi kemampuan yang harus dimiliki oleh sistem yang diharapkan dapat digunakan pada berbagai versi sistem operasi Android.

3.3 Perancangan Sistem

Perancangan sistem dilakukan setelah semua kebutuhan telah di analisis. Pada tahap ini akan dibuat rancangan sistem yang menjelaskan bagaimana sistem bekerja secara keseluruhan agar sistem yang dibuat dapat memenuhi kebutuhan yang telah di definisikan sebelumnya. Proses yang dilakukan dalam tahap perancangan sistem meliputi:

1. Perancangan Arsitektur

Proses perancangan arsitektur merupakan proses yang menjelaskan arsitektur sistem serta pemodelan perancangan menggunakan diagram-diagram UML seperti *sequence diagram* dan *class diagram*.

2. Perancangan Komponen

Proses perancangan komponen dilakukan dengan menuliskan beberapa sampel algoritma yang diambil dari setiap kelas controller yang digambarkan dengan pseudocode.

3. Perancangan Data

Proses perancangan data merupakan proses pemodelan perancangan untuk merepresentasikan model dari penyimpanan data yang ada pada sistem yang digambarkan dengan *Entity Relationship Diagram* (ERD).

4. Perancangan Antarmuka

Proses perancangan antarmuka merupakan proses untuk menggambarkan tata letak dari komponen yang harus disediakan oleh sistem yang dibangun berdasarkan kebutuhan yang telah didefinisikan. Perancangan antarmuka yang digambarkan digunakan sebagai dasar dalam implementasi antarmuka sistem. Dalam proses ini akan digambarkan beberapa sampel antarmuka utama.

3.4 Implementasi

Implementasi merupakan tahap dimana pembangunan sistem dijalankan. Pembangunan sistem tersebut berdasarkan hasil analisis kebutuhan serta perancangan sistem yang sudah dilakukan pada tahap sebelumnya. Implementasi dalam pembangunan aplikasi pada penelitian ini menggunakan bahasa pemrograman seperti *PHP*, *CSS*, *HTML*, dan *JavaScript* dengan memanfaatkan *framework CodeIgniter* dalam pembangunan aplikasi web dan *framework PhoneGap* untuk membangun aplikasi *mobile hybrid*. Pengelolaan data pada penelitian ini dilakukan dengan basis data DBMS MySQL. Implementasi antarmuka berdasarkan perancangan yang telah dilakukan sebelumnya. Pada tahap akhir dalam proses ini dilakukan implementasi simulasi pada *hardware* secara langsung menggunakan Android *mobile device*.

3.5 Pengujian

Pengujian dilakukan setelah proses implementasi telah dilakukan. Pengujian memiliki tujuan untuk mengetahui apakah perangkat lunak yang sudah dibangun telah sesuai dengan kebutuhan yang di definisikan sebelumnya. Terdapat dua pengujian pada penelitian ini yaitu pengujian *whitebox* dan pengujian *blackbox*. Pada pengujian *whitebox* dilakukan dengan metode *basis path testing*. Pada

pengujian *blackbox* dilakukan dengan metode pengujian validasi dan pengujian *compatibility*. Pengujian validasi dilakukan dengan memeriksa hasil yang dikeluarkan oleh sistem ketika dijalankan dengan fungsionalnya guna mengetahui apakah perangkat lunak tersebut telah sesuai dengan kebutuhan fungsional yang telah di definisikan. Pengujian *compatibility* dilakukan dengan menjalankan aplikasi pada perangkat Android dengan versi sistem operasi yang berbeda untuk mengetahui apakah kebutuhan non fungsional sistem telah terpenuhi.

3.6 Kesimpulan Dan Saran

Tahap pengambilan kesimpulan dapat dilakukan setelah seluruh tahap pada penelitian ini telah selesai dilakukan. Kesimpulan didapatkan dari hasil pengujian perangkat lunak yang sudah dilakukan sebelumnya. Saran diberikan untuk pengembangan lebih lanjut dari hasil penelitian ini dan perbaikan kesalahan yang ada.



BAB 4 ANALISIS KEBUTUHAN

Bab ini berisi proses analisis kebutuhan yang merupakan tahap pertama yang dilakukan pada pembangunan sebuah sistem. Tahap ini adalah suatu tahapan untuk menentukan apa saja kebutuhan yang harus ada pada sebuah sistem. Kebutuhan yang didapat pada tahap ini berasal dari permasalahan yang telah di definisikan sebelumnya, yaitu membangun suatu aplikasi berbasis Android dengan memanfaatkan fitur *geotagging* yang dapat membantu pengguna dalam berbagi informasi seputar tempat wisata, membantu pengguna mencari tempat terdekat, memberikan rekomendasi berdasarkan harga, serta membantu pengguna membuat agenda wisata. Pada tahap analisis kebutuhan ini juga dijelaskan siapa saja aktor yang dapat mengakses sistem. Selain itu dalam tahap ini terdapat pemodelan kebutuhan seperti *use case diagram* dan *use case* skenario yang dapat berguna dalam memahami kebutuhan-kebutuhan yang ada.

4.1 Gambaran Umum Aplikasi

Aplikasi *social geotagging* destinasi wisata berbasis Android ini adalah sebuah aplikasi yang merupakan wadah bagi para penikmat wisata untuk berbagi maupun mencari informasi seputar tempat wisata. Pengguna dapat berbagi pengalaman berwisata dengan memanfaatkan fitur *geotagging* untuk mendapatkan data tempat wisata yang dikunjungi. Aplikasi ini menerapkan sistem komunitas dimana data yang menjadi keluaran dalam aplikasi ini merupakan data dari pengguna dan untuk pengguna, selain itu setiap pengguna dapat memberikan komentar terhadap aktivitas yang di bagikan oleh pengguna lainya. Fitur yang dimiliki aplikasi ini mempermudah penggunanya untuk mencari tempat wisata terdekat dari tempat dimana pengguna itu berada. Selain itu pengguna juga dapat melakukan pencarian tempat wisata termurah dan termahal di sekitar tempatnya berada untuk mempermudah pengguna menentukan tempat wisata sesuai dengan kebutuhannya. Pengguna juga dapat menyimpan lokasi yang ingin dikunjungi dan menentukan tanggal yang di rencanakan hal ini dapat mempermudah pengguna untuk mengatur agenda perjalanan wisata yang akan di lakukan.

4.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mendapatkan daftar kebutuhan-kebutuhan aplikasi yang nantinya harus ditangani oleh sistem yang dibangun. Kebutuhan terdiri dari dua jenis yaitu kebutuhan fungsional dan non fungsional. Tahapan yang dilakukan dalam analisis kebutuhan diantaranya identifikasi aktor, analisis kebutuhan fungsional, analisis kebutuhan non-fungsional, serta pemodelan *Use Case* diagram dan *Use Case* skenario.

4.2.1 Identifikasi Aktor

Sistem yang akan dibangun memiliki pengguna yang dapat berbagi informasi foto wisata secara bebas. Kebebasan dalam berbagi informasi menimbulkan

adanya informasi yang tidak sesuai di dalam sistem, sehingga dibutuhkan peran admin yang dapat menjaga informasi yang dibagikan dari konten yang tidak sesuai.

Tahap identifikasi aktor dilakukan dengan mengidentifikasi siapa saja aktor yang akan berinteraksi dengan aplikasi *social geotagging* destinasi wisata. Tabel 4.1 menjelaskan aktor apa saja yang berperan dan deskripsinya.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Tamu	Tamu adalah orang yang belum memiliki akun untuk mengakses aplikasi. Tamu dapat melakukan registrasi pada aplikasi.
Pengguna	Pengguna adalah orang yang telah melakukan registrasi pada aplikasi. Pengguna dapat menambahkan post berupa foto wisata, mengikuti akun pengguna lainnya, menambahkan komentar pada post, membuat agenda wisata, melakukan pencarian tempat wisata, melaporkan post yang tidak sesuai.
Admin	Admin adalah orang yang memiliki tugas mengelola data dalam aplikasi. Admin memiliki hak akses untuk menghapus post yang dilaporkan oleh pengguna.

4.2.2 Analisis Kebutuhan Fungsional

Tahap analisis kebutuhan fungsional menjabarkan fungsi-fungsi yang akan ditangani oleh aplikasi *social geotagging* destinasi wisata untuk memenuhi kebutuhan pengguna. Kebutuhan fungsional didapatkan dengan melihat referensi yang berasal dari penelitian terkait. Berdasarkan penelitian tersebut diambil fitur atau fungsi yang pada umumnya dimiliki dan membantu pengguna aplikasi sejenis sehingga diterapkan di dalam aplikasi yang akan dibangun. Tabel 4.2 menjelaskan judul penelitian yang menjadi referensi dalam menentukan kebutuhan fungsional aplikasi serta fitur yang akan diterapkan dalam aplikasi yang akan dibangun.

Tabel 4.2 Judul Penelitian dan Fitur yang Diterapkan

No	Judul Penelitian	Fitur yang Digunakan
1	Rancang Bangun Aplikasi Sosial <i>Geotagging</i> Keindahan Alam Indonesia pada Sistem Operasi Android.	a. Mengunggah foto b. Melihat daftar foto yang diunggah
2	Rancang Bangun Sistem Informasi Geografis Rekomendasi	a. Mencari tempat terdekat

	Pariwisata Malang dengan Mengimplementasikan Metode Analisis Kebutuhan XREAP	
3	Rancang Bangun Sistem Komunitas Berbagi Informasi Resep dan Lokasi Kuliner Berbasis Mobile	<ul style="list-style-type: none"> a. Mengikuti pengguna lain b. Memberi komentar c. Pencarian pengguna lain

Selain mengacu pada penelitian sebelumnya, penentuan kebutuhan fungsional juga didapatkan dengan mengacu pada aplikasi yang menjadi referensi diantaranya aplikasi foodeo dan TripAdvisor. Aplikasi foodeo memberikan rekomendasi harga termahal dan termurah untuk membantu pengguna dalam mendapatkan tambahan informasi. Aplikasi TripAdvisor menyediakan fasilitas agenda untuk menyimpan tempat-tempat menarik disertakan dengan tanggal untuk membantu pengguna menyusun waktu perkiraan akan melaksanakan kegiatan wisata.

Aplikasi yang akan dibangun merupakan aplikasi dalam bentuk komunitas atau sosial media berbasis foto dimana penggunanya dapat saling mengunggah dan membagikan foto terkait dengan wisata. Untuk memudahkan pengguna dalam melihat foto-foto yang telah dibagikan maka ditampilkan daftar post pengguna. Pengguna juga dapat mencari dan mengikuti pengguna lainya agar memudahkan pengguna untuk melihat post yang dibagikan oleh pengguna yang diikuti. Pengguna dapat memberikan komentar untuk menambah informasi dengan pendapat pribadi pengguna.

Berdasarkan referensi yang digunakan, maka dijabarkan spesifikasi kebutuhan fungsional dari aplikasi *social geotagging* destinasi wisata yang akan di jelaskan pada tabel 4.3 dan tabel 4.4. Setiap Kebutuhan akan diberikan kode SRS-01X-YY-ZZ. SRS merupakan singkatan dari *Software Requirements Specification*, 01 merupakan kode untuk kebutuhan fungsional, X menunjukkan jenis aplikasi, YY menunjukkan nomor dari definisi kebutuhan utama dan ZZ menunjukkan nomor spesifikasi dari definisi kebutuhan utama.

Tabel 4.3 Spesifikasi Kebutuhan Fungsional Aplikasi Pengguna

No	Kebutuhan Sistem	Kode	Use Case	Aktor
1	Aplikasi harus mampu menambah Pengguna baru sehingga user baru dapat mengakses aplikasi.	SRS-01P-01-00	<i>Register</i>	Tamu
1.1	Aplikasi menyediakan <i>form</i> untuk mengisi <i>username, password, nama, alamat, kota, dan</i>	SRS-01P-01-01		



No	Kebutuhan Sistem	Kode	Use Case	Aktor
	telepon.			
2	Aplikasi harus mampu menyediakan fungsi untuk mengizinkan pengguna mengakses sistem sesuai dengan otoritasnya dengan cara memasukkan <i>username</i> dan <i>password</i> .	SRS-01P-02-00	<i>Login</i>	Pengguna
3	Aplikasi harus mampu menampilkan daftar post terbaru dari pengguna yang diikuti.	SRS-01P-03-00	Lihat Post Terbaru	Pengguna
3.1	Aplikasi menyediakan informasi username pengguna yang membagikan post, foto post, lokasi post, harga tiket pada post.	SRS-01P-03-01		
4	Aplikasi harus mampu menampilkan profil pengguna.	SRS-01P-04-00	Lihat Profil	Pengguna
4.1	Aplikasi menyediakan informasi nama pengguna, username pengguna, bio pengguna, jumlah <i>followers</i> dan <i>following</i> pengguna, lokasi pengguna, serta daftar post yang telah dibagikan pengguna.	SRS-01P-04-01		
5	Aplikasi harus mampu menambah <i>post</i> ke dalam sistem.	SRS-01P-05-00	Tambah Post	Pengguna
5.1	Aplikasi menyediakan <i>form</i> untuk mengisi keterangan dan harga tiket.	SRS-01P-05-01		

No	Kebutuhan Sistem	Kode	Use Case	Aktor
5.2	Aplikasi menyediakan tombol untuk <i>browse</i> foto dari galeri.	SRS-01P-05-02		
5.3	Aplikasi menyediakan tombol untuk mengambil foto dengan kamera.	SRS-01P-05-03		
5.4	Aplikasi menyediakan tombol untuk melakukan pemilihan lokasi terdekat dari lokasi foto.	SRS-01P-05-04		
6	Aplikasi harus mampu mengambil foto menggunakan kamera	SRS-01P-06-00	Ambil Foto Dari Kamera	Pengguna
7	Aplikasi harus mampu mengambil foto dari galeri	SRS-01P-07-00	Ambil Foto Dari Galeri	Pengguna
8	Aplikasi harus mampu menambah lokasi ke dalam sistem.	SRS-01P-08-00	Tambah Lokasi	Pengguna
8.1	Aplikasi harus menyediakan form untuk mengisi nama dan alamat lokasi.	SRS-01P-08-01		
9	Aplikasi harus mampu menampilkan daftar nama lokasi wisata terdekat berdasarkan radius yang ditentukan pengguna.	SRS-01P-09-00	Lihat Lokasi Terdekat	Pengguna
9.1	Aplikasi harus menyediakan form untuk mengisi radius.	SRS-01P-09-01		
10	Aplikasi harus mampu menampilkan daftar wisata dengan harga termurah.	SRS-01P-10-00	Lihat Wisata Termurah	Pengguna
11	Aplikasi harus mampu menampilkan daftar wisata dengan harga	SRS-01P-11-00	Lihat Wisata Termahal	Pengguna

No	Kebutuhan Sistem	Kode	Use Case	Aktor
	termahal.			
12	Aplikasi harus mampu menampilkan daftar agenda wisata pengguna.	SRS-01P-12-00	Lihat Agenda	Pengguna
12.1	Aplikasi menyediakan informasi nama lokasi, tanggal rencana wisata, dan keterangan	SRS-01P-12-01		
13	Aplikasi harus mampu menambahkan agenda wisata pengguna ke dalam sistem.	SRS-01P-13-00	Tambah Agenda	Pengguna
13.1	Aplikasi menyediakan <i>form</i> untuk mengisi lokasi, keterangan, dan tanggal.	SRS-01P-13-01		
14	Aplikasi harus mampu mengubah data agenda pengguna.	SRS-01P-14-00	Edit Agenda	Pengguna
14.1	Aplikasi harus menyediakan <i>form</i> untuk mengubah data agenda.	SRS-01P-14-01		
15	Aplikasi harus mampu menghapus data agenda pengguna.	SRS-01P-15-00	Hapus Agenda	Pengguna
15.1	Aplikasi menyediakan tombol hapus untuk melakukan penghapusan data agenda pengguna.	SRS-01P-15-01		
16	Aplikasi harus mampu menampilkan daftar post berdasarkan lokasi.	SRS-01P-16-00	Lihat Post Berdasarkan Lokasi	Pengguna
16.1	Aplikasi menampilkan nama lokasi, lokasi di dalam peta dan menampilkan daftar post sesuai lokasi.	SRS-01P-16-01		

No	Kebutuhan Sistem	Kode	Use Case	Aktor
17	Aplikasi harus mampu menambahkan komentar pengguna ke dalam sistem.	SRS-01P-17-00	Tambah Komentar	Pengguna
17.1	Aplikasi harus menyediakan <i>form</i> untuk mengisi komentar.	SRS-01P-17-01		
18	Aplikasi harus mampu menampilkan komentar Pengguna.	SRS-01P-18-00	Lihat Komentar	Pengguna
18.1	Aplikasi harus menampilkan post yang di komentari, nama pengguna yang mengomentari, dan isi komentar	SRS-01P-18-01		
19	Aplikasi harus mampu menampilkan daftar pengguna lain berdasarkan <i>username</i> yang dicari.	SRS-01P-19-00	Cari Pengguna	Pengguna
19.1	Aplikasi harus menyediakan <i>form</i> untuk mengisi <i>username</i> pengguna yang dicari.	SRS-01P-19-01		
20	Aplikasi harus mampu menampilkan profil pengguna lain.	SRS-01P-20-00	Lihat Profil Pengguna Lain	Pengguna
20.1	Aplikasi menyediakan informasi nama pengguna, <i>username</i> pengguna, bio pengguna, jumlah <i>followers</i> dan <i>following</i> pengguna, lokasi pengguna, serta daftar post yang telah di bagikan pengguna.	SRS-01P-20-01		
20.2	Aplikasi menyediakan tombol untuk mengikuti pengguna	SRS-01P-20-02		

No	Kebutuhan Sistem	Kode	Use Case	Aktor
20.3	Aplikasi menyediakan tombol untuk berhenti mengikuti pengguna	SRS-01P-20-03		
21	Aplikasi harus mampu menyimpan data <i>follow</i> kedalam sistem	SRS-01P-21-00	Ikuti Pengguna	Pengguna
22	Aplikasi harus mampu menghapus data <i>follow</i> kedalam sistem	SRS-01P-22-00	Berhenti Mengikuti Pengguna	Pengguna
23	Aplikasi harus mampu menampilkan daftar nama lokasi berdasarkan kata kunci lokasi yang di cari.	SRS-01P-23-00	Cari Lokasi	Pengguna
23.1	Aplikasi harus menyediakan <i>form</i> untuk mengisi nama lokasi yang dicari.	SRS-01P-23-01		
24	Aplikasi harus mampu menghapus data post pengguna.	SRS-01P-24-00	Hapus Post	Pengguna
24.1	Aplikasi menyediakan tombol hapus untuk melakukan penghapusan data post pengguna.	SRS-01P-24-01		
25	Aplikasi harus mampu mengubah data profil pengguna.	SRS-01P-25-00	Edit Profil	Pengguna
25.1	Aplikasi harus menyediakan <i>form</i> untuk mengisi nama, bio, foto profil, dan lokasi pengguna.	SRS-01P-25-01		
26	Aplikasi harus mampu menyediakan fungsi untuk melaporkan post pengguna yang tidak sesuai dengan aplikasi.	SRS-01P-26-00	Laporkan Post	Pengguna
26.1	Aplikasi menyediakan	SRS-01P-26-		

No	Kebutuhan Sistem	Kode	Use Case	Aktor
	tombol lapor untuk melaporkan post.	01		
27	Aplikasi harus mampu menampilkan daftar notifikasi pengguna.	SRS-01P-27-00	Lihat Notifikasi	Pengguna
27.1	Aplikasi menyediakan informasi post pengguna yang di komentari dan nama pengguna yang mengomentari.	SRS-01P-27-01		
27.2	Aplikasi menyediakan informasi nama pengguna yang mengikuti.	SRS-01P-27-02		
27.3	Aplikasi menyediakan informasi post pengguna yang dihapus karena di laporkan.	SRS-01P-27-03		
28	Aplikasi harus mampu menyediakan fungsi agar pengguna dapat keluar dari sistem.	SRS-01P-28-00	Logout	Pengguna

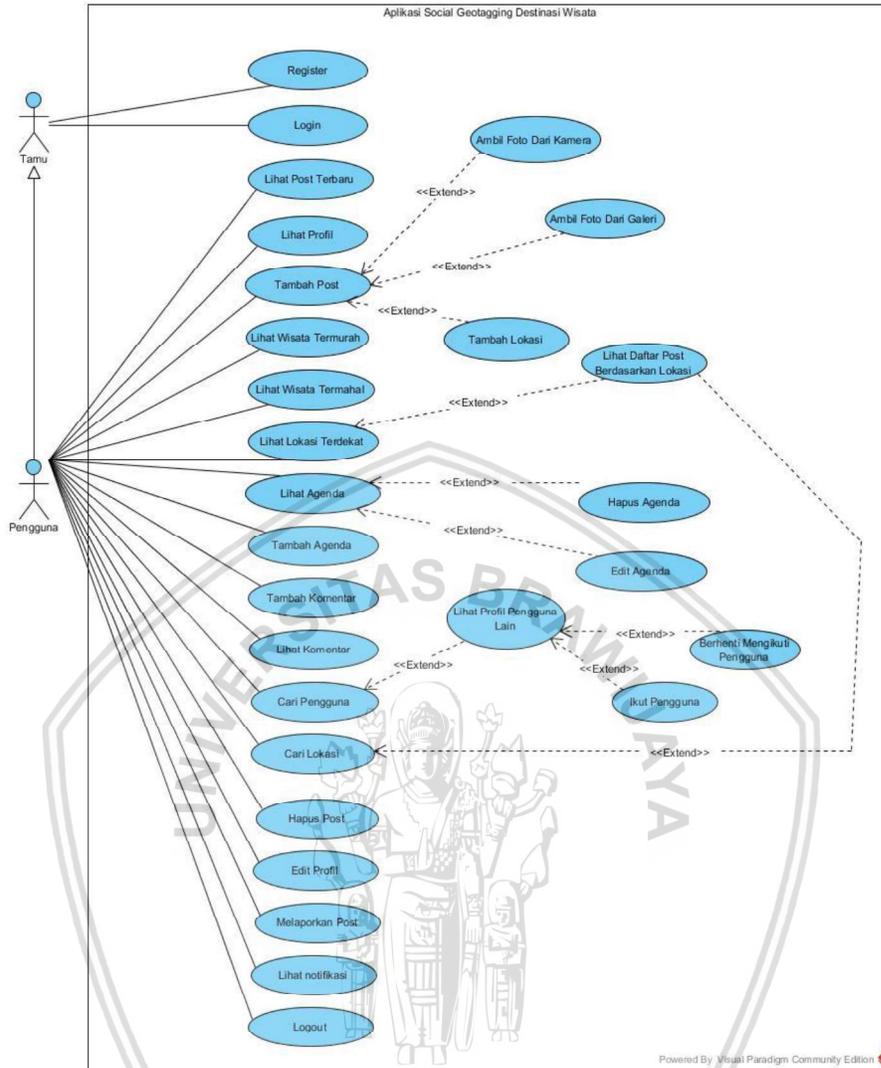
Tabel 4.4 Spesifikasi Kebutuhan Fungsional Aplikasi Admin

No	Kebutuhan Sistem	Kode	Use Case	Aktor
1	Aplikasi harus mampu menyediakan fungsi untuk mengizinkan pengguna mengakses sistem sesuai dengan otoritasnya dengan cara memasukkan <i>username</i> dan <i>password</i> .	SRS-01A-01-00	Login	Admin
2	Aplikasi harus mampu menampilkan daftar post yang dilaporkan.	SRS-01A-02-00	Lihat Daftar Laporan Post	Admin
2.1	Aplikasi menyediakan informasi username			

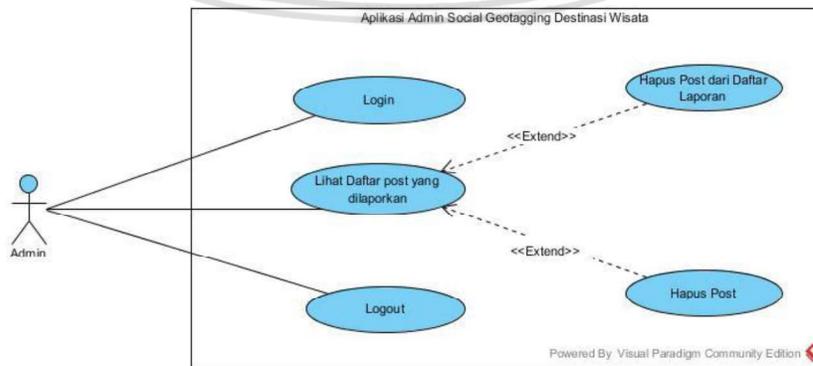
No	Kebutuhan Sistem	Kode	Use Case	Aktor
	pengguna yang membagikan post dan post yang laporkan			
3	Aplikasi harus mampu menghapus data post dari daftar laporan.	SRS-01A-03-00	Hapus Post dari Daftar Laporan	Admin
3.1	Aplikasi menyediakan tombol hapus untuk melakukan penghapusan data post dari daftar laporan.	SRS-01A-03-01		
4	Aplikasi harus mampu menghapus data post dari sistem.	SRS-01A-04-00	Hapus Post	Admin
4.1	Aplikasi menyediakan tombol hapus untuk melakukan penghapusan data post dari sistem.	SRS-01A-04-01		
5	Aplikasi harus mampu menyediakan fungsi agar pengguna dapat keluar dari sistem.	SRS-01A-08-00	Logout	Admin

A. Use Case Diagram

Use case diagram merupakan diagram yang digunakan dalam memodelkan perilaku suatu sistem. *Use case* diagram menggambarkan seluruh kebutuhan fungsional sistem yang akan dibangun. Tujuan dari pembuatan *use case* diagram adalah agar pembaca dapat memahami dan mengerti sistem yang akan dibangun dengan mudah. Pemodelan *use case diagram* terbagi dalam 2 bagian. *Use case diagram* pertama menggambarkan *use case diagram* dari aplikasi yang akan digunakan oleh sisi pengguna, yang ditunjukkan pada Gambar 4.1 dan *use case diagram* kedua menggambarkan *use case diagram* dari aplikasi yang akan digunakan oleh sisi Admin yang ditunjukkan pada Gambar 4.2.



Gambar 4.1 Use Case Diagram Pengguna



Gambar 4.2 Use Case Diagram Admin

B. Use Case Skenario

Use Case yang telah digambarkan dalam diagram use case akan lebih dijelaskan secara terperinci dalam use case skenario. Penggunaan use case skenario ini bertujuan untuk mendapatkan deskripsi secara global mengenai use case, kondisi awal dan akhir yang harus dipenuhi oleh use case setelah fungsionalitas selesai dijalankan. Dalam skenario ini akan diulas bagaimana tanggapan sistem terhadap aksi yang dilakukan oleh aktor. Tabel 4.5 sampai dengan Tabel 4.36 menjelaskan use case skenario dari aplikasi yang akan dibangun.

Tabel 4.5 Use Case Skenario Register

<i>Objective</i>	Aktor dapat mendaftarkan akun baru untuk mengakses aplikasi	
Aktor	Tamu	
<i>Pre-Condition</i>	Aktor telah mengakses halaman login aplikasi <i>social geotagging</i> destinasi wisata	
<i>Main Flow</i>	Aktor	Sistem
	<ol style="list-style-type: none"> 1. Aktor menekan tombol Daftar 3. Aktor mengisi form pendaftaran dengan data diri (<i>Username, password, Nama, Alamat, Kota, Telepon</i>) 4. Aktor menekan tombol daftar 	<ol style="list-style-type: none"> 2. Sistem menampilkan form pendaftaran 5. Sistem menyimpan data pengguna baru ke dalam <i>database</i>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Apabila <i>form</i> registrasi tidak diisi dengan lengkap, maka sistem akan menampilkan pesan kesalahan "<i>The field is required</i>" 2. Apabila <i>username</i> yang diisi sudah ada di dalam sistem maka akan menampilkan pesan kesalahan "<i>Username tidak tersedia</i>" 	
<i>Post Condition</i>	Aktor berhasil membuat akun baru untuk mengakses aplikasi	

Tabel 4.6 Use Case Skenario Login

<i>Objective</i>	Aktor dapat masuk ke dalam akun aplikasi sesuai dengan otoritasnya
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah mengakses halaman login

<i>Main Flow</i>	Aktor	Sistem
	<ol style="list-style-type: none"> 1. Aktor mengisi <i>username</i> dan <i>password</i> pada form login 2. Aktor menekan tombol login 	<ol style="list-style-type: none"> 3. Sistem menampilkan halaman pos terbaru
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Apabila kombinasi dari <i>username</i> dan <i>password</i> tidak terdaftar di dalam <i>database</i> sistem, maka sistem akan menampilkan sebuah pesan kesalahan “<i>username dan password salah</i>” 2. Apabila <i>form</i> login tidak diisi dengan lengkap maka sistem akan menampilkan pesan kesalahan “<i>The field is required</i>” 	
<i>Post Condition</i>	Aktor berhasil masuk ke dalam akun dengan otoritasnya	

Tabel 4.7 Use Case Skenario Lihat Post Terbaru

<i>Objective</i>	Aktor dapat melihat daftar post terbaru aplikasi	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman <i>home</i> aplikasi <i>social geotagging</i> destinasi wisata	
<i>Main Flow</i>	Aktor	Sistem
	<ol style="list-style-type: none"> 1. Aktor menekan tombol wisata terbaru 	<ol style="list-style-type: none"> 2. Sistem menampilkan daftar post terbaru
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Daftar post terbaru berhasil ditampilkan oleh aplikasi	

Tabel 4.8 Use Case Skenario Lihat Profil

<i>Objective</i>	Aktor dapat melihat profil pengguna	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah berhasil melakukan <i>login</i>	
<i>Main Flow</i>	Aktor	Sistem
	<ol style="list-style-type: none"> 1. Aktor menekan tombol profile 	<ol style="list-style-type: none"> 2. Sistem menampilkan profile pengguna beserta daftar post pengguna
<i>Alternative Flow</i>	-	

<i>Post Condition</i>	Profil pengguna berhasil ditampilkan oleh aplikasi
-----------------------	--

Tabel 4.9 Use Case Skenario Tambah Post

<i>Objective</i>	Aktor dapat menambahkan post berupa foto dengan lokasi	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah berhasil melakukan <i>login</i>	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol tambah post	2. Sistem menampilkan <i>form</i> untuk menambah post (foto, lokasi, harga, keterangan)
	3. Aktor mengisi <i>form</i> tambah post	
	4. Aktor menekan tombol post baru	5. Sistem menampilkan halaman post terbaru
<i>Alternative Flow</i>	1. Apabila <i>form</i> tidak diisi dengan lengkap, maka sistem akan menampilkan pesan kesalahan " <i>The field is required</i> "	
<i>Post Condition</i>	Post baru berhasil ditambahkan dan di tampilkan di halaman post terbaru	

Tabel 4.10 Use Case Skenario Ambil Foto Dari Kamera

<i>Objective</i>	Aktor dapat mengambil foto menggunakan kamera	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman tambah foto	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol foto kamera	2. Sistem mengakses kamera pada perangkat
	3. Aktor mengambil foto menggunakan kamera	4. Sistem menampilkan <i>form</i> tambah post
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Foto berhasil diambil menggunakan kamera	

Tabel 4.11 Use Case Skenario Ambil Foto Dari Galeri

<i>Objective</i>	Aktor dapat mengambil foto dari galeri perangkat	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman tambah foto	
<i>Main Flow</i>	Aktor	Sistem

	1. Aktor menekan tombol foto album 3. Aktor memilih foto dari galeri perangkat	2. Sistem mengakses galeri pada perangkat 4. Sistem menampilkan <i>form</i> tambah post
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Foto berhasil diambil dari galeri perangkat	

Tabel 4.12 Use Case Skenario Tambah Lokasi

<i>Objective</i>	Aktor dapat menambahkan lokasi ke dalam sistem	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman tambah foto	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol tambah lokasi 3. Aktor mengisi <i>form</i> nama dan alamat lokasi dan menekan tombol lokasi baru	2. Sistem menampilkan <i>form</i> untuk mengisi nama dan alamat lokasi 4. Sistem menampilkan <i>form</i> tambah post
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Lokasi berhasil ditambahkan dan form tambah post ditampilkan	

Tabel 4.13 Use Case Skenario Lihat Lokasi Terdekat

<i>Objective</i>	Aktor dapat melihat daftar lokasi terdekat dari posisi user berdasarkan radius	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman <i>home</i> aplikasi <i>social geotagging</i> destinasi wisata	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol wisata terdekat 3. User mengisi form radius dan menekan tombol pencarian	2. Sistem menampilkan form untuk mengisi radius 4. Sistem menampilkan nama lokasi terdekat berdasarkan radius
<i>Alternative Flow</i>	1. Apabila <i>form</i> tidak diisi dengan lengkap, maka sistem akan menampilkan pesan kesalahan " <i>The field is required</i> "	

<i>Post Condition</i>	Daftar lokasi terdekat berdasarkan radius berhasil di tampilkan
-----------------------	---

Tabel 4.14 Use Case Skenario Lihat Wisata Termurah

<i>Objective</i>	Aktor dapat melihat daftar post wisata dengan harga termurah	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman <i>home</i> aplikasi <i>social geotagging</i> destinasi wisata	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor mennekan tombol wisata termurah	2. Sistem menampilkan daftar post yang diurutkan berdasarkan harga terkecil
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Daftar post wisata dengan harga termurah berhasil ditampilkan	

Tabel 4.15 Use Case Skenario Lihat Wisata Termahal

<i>Objective</i>	Aktor dapat melihat daftar post wisata dengan harga termahal	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman <i>home</i> aplikasi <i>social geotagging</i> destinasi wisata	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor mennekan tombol wisata termahal	2. Sistem menampilkan daftar post yang diurutkan berdasarkan harga terbesar
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Daftar post wisata dengan harga termahal berhasil ditampilkan	

Tabel 4.16 Use Case Skenario Lihat Agenda

<i>Objective</i>	Aktor dapat melihat daftar agenda yang dimiliki
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah mengakses halaman <i>home</i> aplikasi <i>social</i>



	<i>geotagging</i> destinasi wisata	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol agenda	2. Sistem menampilkan daftar agenda yang dimiliki oleh aktor
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Daftar agenda berhasil ditampilkan	

Tabel 4.17 Use Case Skenario Tambah Agenda

<i>Objective</i>	Aktor dapat menambahkan agenda wisata ke dalam daftar	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah berhasil melakukan <i>login</i>	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol tambah agenda 3. Aktor mengisi <i>form</i> agenda dan menekan tombol simpan agenda	2. Sistem menampilkan <i>form</i> untuk mengisi lokasi, tanggal, dan keterangan agenda 4. Sistem menampilkan halaman daftar agenda
<i>Alternative Flow</i>	1. Apabila <i>form</i> tidak diisi dengan lengkap, maka sistem akan menampilkan pesan kesalahan " <i>The field is required</i> "	
<i>Post Condition</i>	Agenda berhasil ditambahkan dan daftar agenda berhasil ditampilkan	

Tabel 4.18 Use Case Skenario Edit Agenda

<i>Objective</i>	Aktor dapat merubah data agenda	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman daftar agenda	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol edit pada agenda 3. Aktor mengisi <i>form</i> edit agenda dan menekan tombol simpan agenda	2. Sistem menampilkan <i>form</i> untuk tanggal, dan keterangan agenda. 4. Sistem menampilkan halaman daftar agenda
<i>Alternative Flow</i>	-	

<i>Post Condition</i>	Data agenda pengguna berhasil di ubah
-----------------------	---------------------------------------

Tabel 4.19 Use Case Skenario Hapus Agenda

<i>Objective</i>	Aktor dapat menghapus data agenda	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman daftar agenda	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol hapus	2. Sistem menghapus agenda dan menampilkan halaman daftar agenda
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Data agenda pengguna berhasil di hapus	

Tabel 4.20 Use Case Skenario Cari Pengguna

<i>Objective</i>	Aktor dapat melihat daftar user sesuai dengan pencarian	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman <i>home</i> aplikasi <i>social geotagging</i> destinasi wisata	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol cari user	2. Sistem menampilkan <i>form</i> untuk mengisi <i>username</i> pengguna
	3. Aktor mengisi <i>form</i> dengan <i>username</i> pengguna	
	4. Aktor menekan tombol pencarian user	5. Menampilkan daftar pengguna yang terkait dengan pencarian
<i>Alternative Flow</i>	1. Apabila <i>form</i> pencarian tidak diisi, maka sistem akan menampilkan pesan kesalahan " <i>The field is required</i> " 2. Apabila pengguna tidak ditemukan maka form akan ditampilkan kembali	
<i>Post Condition</i>	Sistem berhasil menampilkan daftar pengguna yang dicari	

Tabel 4.21 Use Case Skenario Lihat Profil Pengguna Lain

<i>Objective</i>	Aktor dapat melihat profil pengguna lain
Aktor	Pengguna

<i>Pre-Condition</i>	Aktor telah mengakses halaman hasil pencarian pengguna	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor memilih nama pengguna	3. Sistem menampilkan halaman profil pengguna lain
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Sistem berhasil menampilkan profil pengguna lain	

Tabel 4.22 Use Case Skenario Ikuti Pengguna

<i>Objective</i>	Aktor dapat mengikuti pengguna lain	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman profil pengguna lain	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol <i>follow</i>	2. Sistem menyimpan data <i>follow</i> , menambahkan jumlah followers pengguna yang diikuti, dan menampilkan halaman profil pengguna lain 3. Sistem mengirim notifikasi kepada pengguna yang diikuti
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Sistem berhasil menyimpan data <i>follow</i> dan menambahkan jumlah <i>followers</i> pengguna yang diikuti	

Tabel 4.23 Use Case Skenario Berhenti Mengikuti Pengguna

<i>Objective</i>	Aktor dapat berhenti mengikuti <i>update</i> post pengguna lain	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman profil pengguna lain	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol <i>unfollow</i>	2. Sistem menghapus data <i>follow</i> dan menampilkan halaman profil pengguna lain
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Sistem berhasil menyimpan data <i>follow</i> dan mengurangi jumlah <i>followers</i> pengguna yang berhenti diikuti	

Tabel 4.24 Use Case Skenario Cari Lokasi

<i>Objective</i>	Aktor dapat melihat daftar nama lokasi yang dicari	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman <i>home</i> aplikasi <i>social geotagging</i> destinasi wisata	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol cari lokasi	2. Sistem menampilkan <i>form</i> untuk mengisi nama lokasi
	3. Aktor mengisi nama lokasi yang di cari	
	4. Aktor menekan tombol pencarian	5. Sistem menampilkan daftar nama lokasi terkait
<i>Alternative Flow</i>	1. Apabila <i>form</i> pencarian tidak diisi, maka sistem akan menampilkan pesan kesalahan " <i>The field is required</i> " 2. Apabila lokasi tidak ditemukan maka sistem akan menampilkan pesan kesalahan "Lokasi tidak ditemukan"	
<i>Post Condition</i>	Sistem berhasil menampilkan daftar nama lokasi yang dicari	

Tabel 4.25 Use Case Skenario Lihat Post Berdasarkan Lokasi

<i>Objective</i>	Aktor dapat melihat daftar post sesuai dengan lokasi yang dicari	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman hasil pencarian lokasi	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan nama lokasi	2. Sistem menampilkan daftar post dengan lokasi terkait dan posisi lokasi di dalam peta
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Sistem berhasil menampilkan daftar post sesuai dengan lokasi yang dicari	

Tabel 4.26 Use Case Skenario Tambah Komentar

<i>Objective</i>	Aktor dapat menambahkan komentar pada post
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah mengakses halaman post terbaru

<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol komentar pada post 3. Aktor mengisi form komentar dan menekan tombol kirim	2. Sistem menampilkan detail post dan <i>form</i> untuk mengisi komentar 4. Sistem menampilkan komentar dan mengirim notifikasi kepada penerima komentar
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Komentar berhasil di tambahkan dan sistem menampilkan komentar pada post	

Tabel 4.27 Use Case Skenario Lihat Komentar

<i>Objective</i>	Aktor dapat melihat komentar pada post	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman post terbaru	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol komentar pada post	2. Sistem menampilkan detail post dan komentar pada post
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Komentar berhasil ditampilkan	

Tabel 4.28 Use Case Skenario Edit Profil

<i>Objective</i>	Aktor dapat merubah data profil	
Aktor	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman profil	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor mengisi form edit profil (foto profil, nama, bio, dan lokasi) 2. Aktor menekan tombol update	3. Sistem menyimpan data profil yang telah diedit ke dalam sistem dan menampilkan halaman profil
<i>Alternative Flow</i>	-	



<i>Post Condition</i>	Data profil yang sudah diubah berhasil di simpan ke dalam sistem dan data profil baru berhasil di tampilkan
-----------------------	---

Tabel 4.29 Use Case Skenario Hapus Post

<i>Objective</i>	Aktor dapat menghapus post dari sistem	
<i>Aktor</i>	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman profil pengguna	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol hapus pada post yang dipilih	2. Sistem menghapus post yang di pilih dari sistem dan menampilkan daftar post pengguna
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Post yang dipilih berhasil di hapus dari sistem	

Tabel 4.30 Use Case Skenario Laporkan Post

<i>Objective</i>	Aktor dapat melaporkan post yang tidak sesuai	
<i>Aktor</i>	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman post terbaru	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol lapor pada post 2. Aktor memilih alasan melaporkan post	3. Sistem menyimpan laporan ke dalam sistem
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Laporan post yang tidak sesuai berhasil di simpan ke dalam database	

Tabel 4.31 Use Case Skenario Lihat Notifikasi

<i>Objective</i>	Aktor dapat melihat daftar notifikasi	
<i>Aktor</i>	Pengguna	
<i>Pre-Condition</i>	Aktor telah mengakses halaman <i>home</i> aplikasi <i>social geotagging</i> destinasi wisata	
<i>Main Flow</i>	Aktor	Sistem

	1. Aktor menekan tombol notifikasi	2. Sistem menampilkan daftar notifikasi
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Daftar notifikasi berhasil ditampilkan	

Tabel 4.32 Use Case Skenario Logout

<i>Objective</i>	Aktor dapat keluar dari akun aplikasi <i>social geotagging</i> destinasi wisata	
Aktor	User, Admin	
<i>Pre-Condition</i>	Aktor telah masuk kedalam akun sistem	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor mengklik tombol sign out	2. Sistem mengeluarkan aktor dari akun sistem dan menampilkan halaman login
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Aktor berhasil keluar dari akun dan session dihapus	

Tabel 4.33 Use Case Skenario Login

<i>Objective</i>	Aktor dapat masuk ke dalam akun aplikasi sesuai dengan otoritasnya	
Aktor	Admin	
<i>Pre-Condition</i>	Aktor telah mengakses halaman login	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor memasukkan <i>username</i> dan <i>password</i> pada form login 2. Aktor menekan tombol login	3. Sistem menampilkan halaman daftar laporan
<i>Alternative Flow</i>	1. Apabila kombinasi dari <i>username</i> dan <i>password</i> tidak terdaftar di dalam <i>database</i> sistem, maka sistem akan menampilkan pesan kesalahan "username dan password salah" 2. Apabila <i>form</i> login tidak diisi dengan lengkap maka sistem akan menampilkan suatu pesan kesalahan " <i>The field is</i>	

	<i>required</i>
<i>Post Condition</i>	Aktor berhasil masuk ke dalam akun dengan otoritasnya

Tabel 4.34 Use Case Skenario Lihat Daftar Laporan Post

<i>Objective</i>	Aktor dapat melihat daftar post yang dilaporkan	
<i>Aktor</i>	Admin	
<i>Pre-Condition</i>	Aktor telah berhasil melakukan <i>login</i>	
<i>Main Flow</i>	Aktor	Sistem
		1. Sistem menampilkan daftar post yang dilaporkan
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Daftar post yang dilaporkan berhasil ditampilkan	

Tabel 4.35 Use Case Skenario Hapus Post Dari Daftar Laporan

<i>Objective</i>	Aktor dapat menghapus post yang telah dilaporkan dari daftar laporan	
<i>Aktor</i>	Admin	
<i>Pre-Condition</i>	Aktor telah berada pada halaman daftar laporan post	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol hapus laporan	2. Sistem menghapus post yang di pilih dari daftar laporan dan menampilkan daftar post yang dilaporkan
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Post yang dilaporkan berhasil di hapus dari daftar laporan	

Tabel 4.36 Use Case Skenario Hapus Post

<i>Objective</i>	Aktor dapat menghapus post yang telah dilaporkan dari sistem	
<i>Aktor</i>	Admin	
<i>Pre-Condition</i>	Aktor telah mengakses halaman daftar laporan post	
<i>Main Flow</i>	Aktor	Sistem
	1. Aktor menekan tombol hapus total	2. Sistem menghapus post yang di pilih dari sistem dan menampilkan daftar post yang dilaporkan



		3. Sistem mengirim notifikasi kepada pengguna yang memiliki post yang di hapus
<i>Alternative Flow</i>	-	
<i>Post Condition</i>	Post yang dilaporkan berhasil di hapus dari sistem	

4.2.3 Analisis Kebutuhan Non Fungsional

Tahap Analisis kebutuhan non-fungsional merupakan tahap yang dilakukan untuk menentukan kemampuan dari keseluruhan perangkat lunak yang dibangun sehingga memiliki kinerja yang baik saat digunakan.

Analisis kebutuhan non-fungsional dilakukan dengan menganalisis kemampuan yang dimiliki oleh aplikasi yang menjadi referensi. Aplikasi yang menjadi referensi adalah aplikasi foodeo dan TripAdvisor. Kedua aplikasi tersebut memiliki kemampuan *compatibility* yang baik sehingga fungsi-fungsi yang dimiliki tetap berjalan dengan baik ketika digunakan pada perangkat Android dengan versi sistem operasi yang berbeda. Sehingga pengguna dapat menjalankan aplikasi dengan baik di berbagai jenis perangkat Android.

Tabel 4.37 menjelaskan kebutuhan non-fungsional yang harus dimiliki oleh aplikasi yang akan dibangun beserta deskripsinya.

Tabel 4.37 Kebutuhan Non-Fungsional

Nomor Kebutuhan	Parameter Kebutuhan	Deskripsi
SRS-02-01	<i>Compatibility</i>	Aplikasi harus dapat dijalankan di berbagai versi sistem operasi Android minimal versi 4.2 (<i>Jelly Bean</i>)

BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM

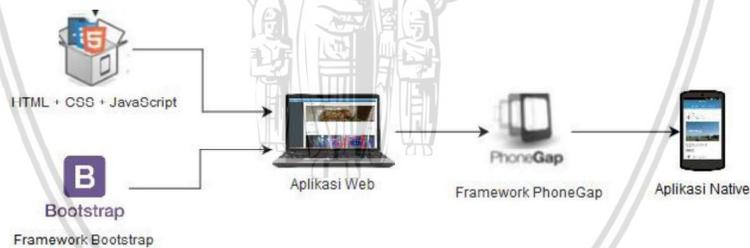
Bab ini berisi pembahasan mengenai perancangan dan implementasi sistem dari aplikasi *social geotagging* destinasi wisata.

5.1 Perancangan Sistem

Proses perancangan sistem dilakukan setelah proses analisis kebutuhan telah selesai dilaksanakan. Perancangan dilakukan dengan berdasarkan hasil proses analisis kebutuhan. Proses dari perancangan aplikasi *social geotagging* destinasi wisata terdiri dari beberapa tahapan diantaranya pemodelan *sequence diagram*, pemodelan *class diagram*, perancangan komponen, perancangan data, serta perancangan antarmuka.

5.1.1 Perancangan Arsitektur Sistem

Aplikasi *social geotagging* destinasi wisata dibangun dengan menerapkan konsep pengembangan *hybrid mobile application* pada perangkat Android. Dengan menerapkan konsep pengembangan *hybrid* maka dimungkinkan untuk membuat suatu aplikasi *native* yang dibangun dengan bahasa pemrograman web seperti HTML, CSS, dan JavaScript. Untuk membuat antarmuka aplikasi digunakan framework Bootstrap sehingga aplikasi dapat menyesuaikan tampilan ketika diaplikasikan dalam perangkat *mobile*. Framework PhoneGap digunakan untuk membangun aplikasi web yang telah dibuat menjadi aplikasi *native*. Gambar 5.1 menggambarkan perancangan arsitektur sistem yang dibangun.



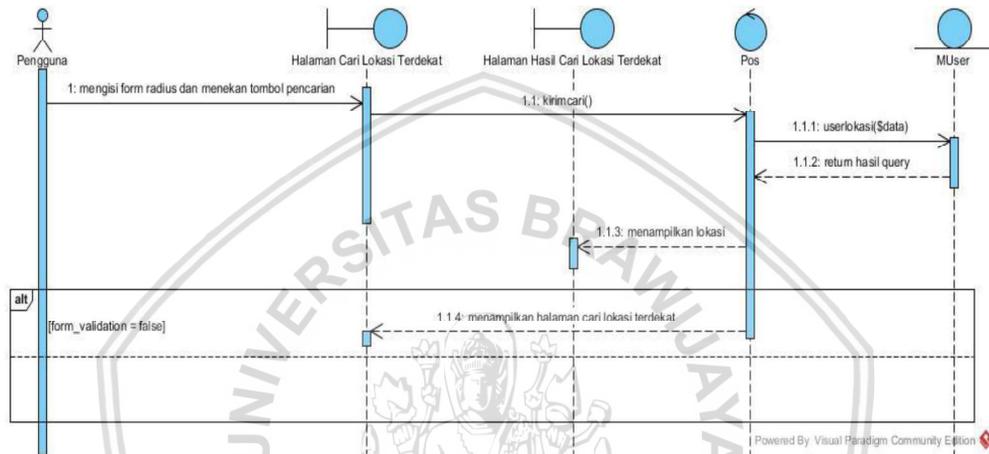
Gambar 5.1 Perancangan Arsitektur Sistem

5.1.2 Pemodelan *Sequence Diagram*

Pemodelan *sequence diagram* berfungsi untuk menjelaskan urutan dari proses-proses yang terjadi dalam mencapai sebuah kebutuhan dalam sistem aplikasi *social geotagging* destinasi wisata. *Sequence diagram* terdiri dari objek-objek yang merupakan hasil dari identifikasi spesifikasi kebutuhan dan *use case* skenario. Terdapat 3 sampel dalam pemodelan *sequence diagram* yang akan ditampilkan. Setiap *sequence diagram* menggambarkan sebuah *use case*.

5.1.2.1 Sequence Diagram Lihat Lokasi Terdekat

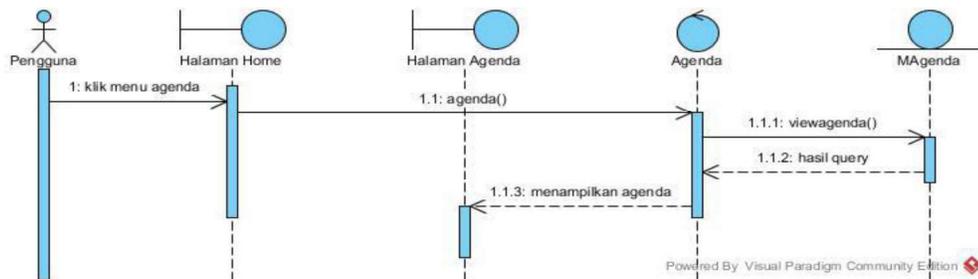
Pada *sequence diagram* lihat lokasi terdekat, pengguna mengakses halaman cari lokasi terdekat kemudian mengisi form dengan radius yang diinginkan dan menekan tombol pencarian. Setelah pengguna menekan tombol pencarian maka sistem akan menjalankan fungsi `kirimcari()` pada controller yang akan meminta data ke model melalui fungsi `userlokasi()`. Kemudian data dikirimkan kembali ke controller dan sistem akan diredirect ke halaman hasil cari lokasi terdekat yang akan menampilkan hasil pencarian lokasi terdekat.



Gambar 5.2 Sequence Diagram Lihat Lokasi Terdekat

5.1.2.2 Sequence Diagram Lihat Agenda

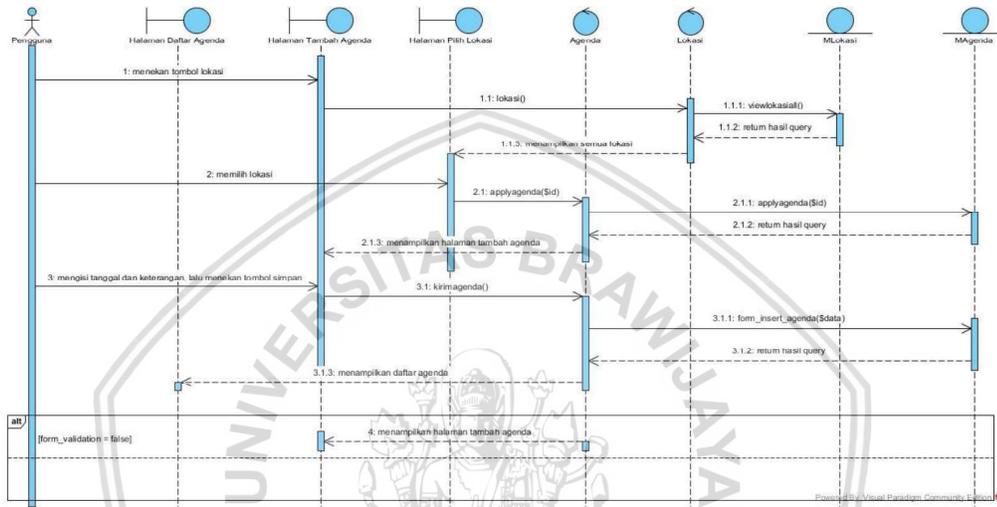
Pada *sequence diagram* lihat agenda, pengguna mengakses halaman home kemudian memilih menu agenda. Setelah pengguna menekan menu agenda maka sistem akan menjalankan fungsi `agenda()` pada controller yang akan meminta data ke model melalui fungsi `viewagenda()`. Kemudian data dikirimkan kembali ke controller dan sistem akan diredirect ke halaman agenda yang akan menampilkan daftar agenda pengguna.



Gambar 5.3 Sequence Diagram Lihat Agenda

5.1.2.3 Sequence Diagram Tambah Agenda

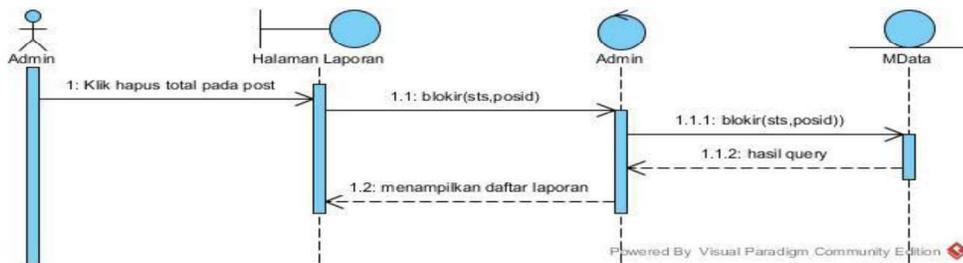
Pada *sequence diagram* tambah agenda, pengguna mengakses halaman tambah agenda kemudian memilih lokasi dan mengisi form yang disediakan. Setelah pengguna menekan tombol simpan maka sistem akan menjalankan fungsi `kirimagenda()` pada controller yang akan melakukan insert data melalui model. Kemudian data dikirimkan kembali ke controller dan sistem akan diredirect ke halaman daftar agenda.



Gambar 5.4 Sequence Diagram Tambah Agenda

5.1.2.4 Sequence Diagram Hapus Laporan

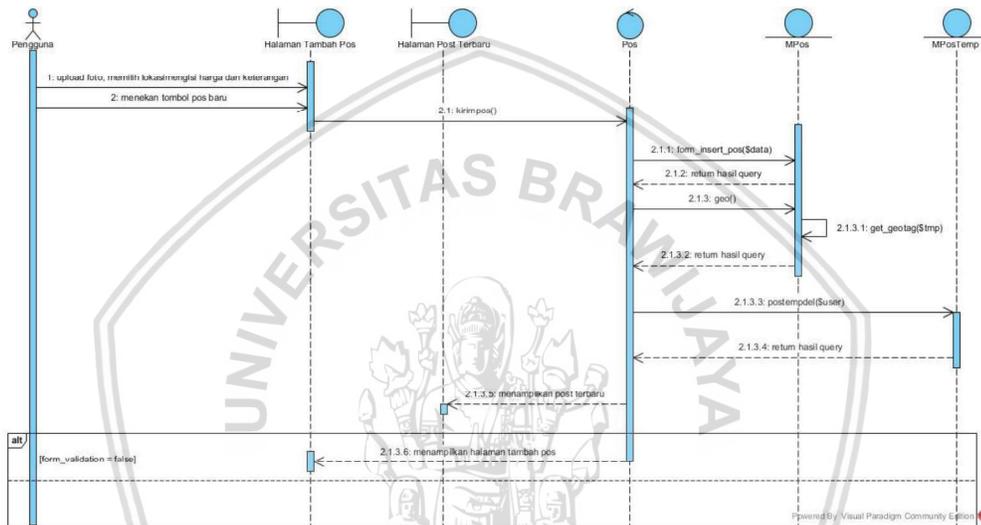
Pada *sequence diagram* hapus laporan, admin mengakses halaman laporan kemudian menekan menu hapus total pada post yang akan dihapus dari sistem. Setelah admin menekan menu hapus total maka sistem akan menjalankan fungsi `blokir()` pada controller yang akan menghapus data melalui fungsi `blokir()` pada model. Kemudian data dikirimkan kembali ke controller dan sistem akan diredirect ke halaman laporan yang akan menampilkan daftar laporan.



Gambar 5.5 Sequence Diagram Hapus Laporan

5.1.2.5 Sequence Diagram Tambah Post

Pada *sequence diagram* tambah post, pengguna mengakses halaman tambah post kemudian melakukan upload foto, memilih lokasi post, dan mengisi form harga tiket dan keterangan. Setelah pengguna menekan tombol pos baru maka sistem akan menjalankan fungsi kirimpos() pada controller yang akan melakukan insert data melalui model, lalu mengambil geotag pada foto dengan menjalankan fungsi geo() pada MPos dan menghapus data post sementara melalui model MPosTemp. Kemudian data dikirimkan kembali ke controller dan sistem akan redirect ke halaman post terbaru.

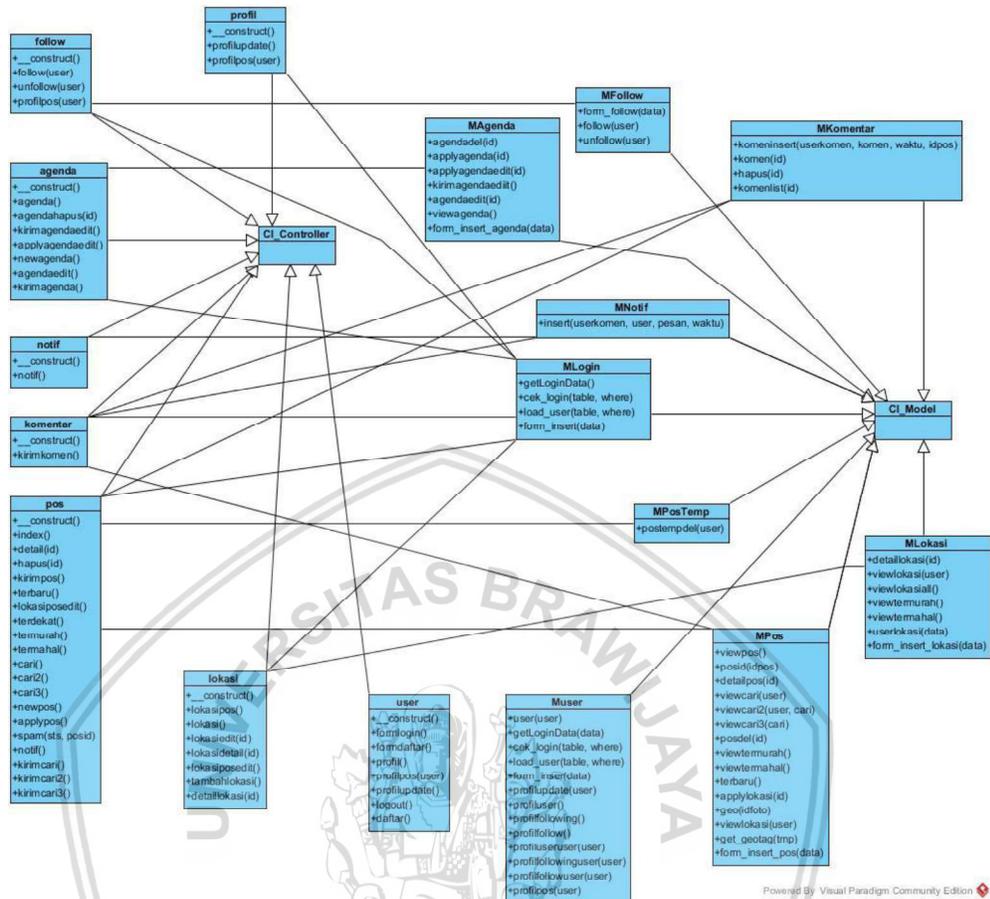


Gambar 5.6 Sequence Diagram Tambah Post

5.1.3 Pemodelan Class Diagram

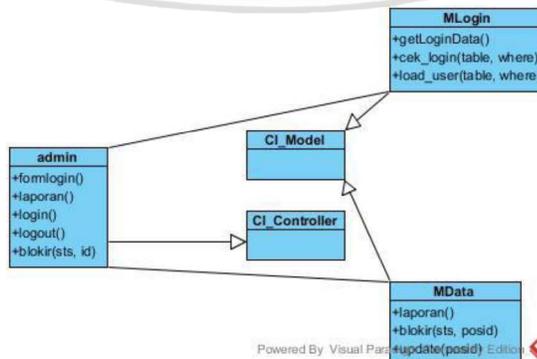
Pemodelan *class diagram* berfungsi untuk menggambarkan objek-objek yang terbentuk serta bagaimana relasi yang terjadi diantara objek-objek tersebut. Pemodelan *class diagram* aplikasi pengguna digambarkan pada Gambar 5.7, dan pemodelan *class diagram* aplikasi admin digambarkan pada Gambar 5.8

Pada *class diagram* yang digambarkan dalam Gambar 5.7, terdapat 2 *class* induk, diantaranya *class* CI_Controller dan CI_Model. Setiap *class* induk memiliki *class* turunan yang masing-masing berjumlah 8 pada *class* CI_Controller dan 9 pada *class* CI_Model. *Class* yang merupakan turunan dari CI_Controller mempunyai hubungan asosiasi dengan *class* yang merupakan turunan dari CI_Model.



Gambar 5.7 Class Diagram Aplikasi Pengguna

Pada *class diagram* yang digambarkan dalam Gambar 5.8, terdapat 2 *class* induk, diantaranya *class* CI_Controller dan CI_Model. Setiap *class* induk memiliki *class* turunan yang masing-masing berjumlah 1 pada *class* CI_Controller dan 2 pada *class* CI_Model. *Class* yang merupakan turunan dari CI_Controller mempunyai hubungan asosiasi dengan kelas yang merupakan turunan dari CI_Model.



Gambar 5.8 Class Diagram Aplikasi Admin

5.1.4 Perancangan Komponen

Perancangan komponen berfungsi untuk menggambarkan proses yang terjadi pada suatu algoritma dalam mencapai tujuan dari suatu fungsional tertentu yang digambarkan dengan pseudocode. Pada tahap perancangan komponen ini dipilih satu algoritma dari setiap controller yang ada.

5.1.4.1 Perancangan Komponen Klas Agenda

Nama operasi : kirimagenda()

Algoritma :

```

set form validation Lokasi
set form validation Wisata
set form validation Tgl
set form validation Keterangan
if(form_validation = false) then
    menampilkan halaman tambah agenda
else
    inialisasi $time = waktu
    inialisasi $data = array (
        Lokasi = input->post('Wisata'),
        Keterangan = input->post('Keterangan'),
        Tgl = input->post('Tgl')
        Waktu = $time
        User = session->userdata('Username'))
    panggil fungsi form_insert_agenda()
    menampilkan halaman agenda
end if

```

5.1.4.2 Perancangan Komponen Klas Follow

Nama operasi : follow()

Algoritma :

```

inialisasi $data = array(
    IDUser = session->userdata('Username'),
    IDFollowing = $user )

```

```

inisialisasi $q = panggil fungsi follow()
inisialisasi $cek = jumlah row $q
if ($cek = 0) then
    panggil fungsi form_follow()
end if
inisialisasi $pesan = "User : "username pengguna" telah follow Anda"
query insert notif
menampilkan halaman profil pengguna

```

5.1.4.3 Perancangan Komponen Klas Komentar

Nama operasi : kirimkomen()

Algoritma :

```

inisialisasi $komen = input->post('Komen')
inisialisasi $IDPos = input->post('IDPos')
inisialisasi $waktu = waktu
inisialisasi $UserKomen = session->userdata('Username');
panggil fungsi komeninsert()
inisialisasi $q = query select * from pos where id ='$idpos
inisialisasi $rowpos = nilai $q
inisialisasi $pesan = wisata, lokasi, dan keterangan rowpos
panggil fungsi insert()
inisialisasi nilai $q = MKomentar->komen($IDPos)
data['totkomen'] = jumlah row $q
data['rows'] = hasil query $q
panggil fungsi detailpos()
menampilkan detail pos

```

5.1.4.4 Perancangan Komponen Klas Lokasi

Nama operasi : lokasipos()

Algoritma :

```

set form validation Lokasi
set form validation Alamat

```

```

inisialisasi $time = waktu
    inisialisasi $data = array (
        Lokasi = input->post('Lokasi'),
        Alamat = input->post('Alamat'),
        Lat = input->post('Lat')
        Lang = input->post('Lang')
        Stamp = $_SESSION['Stamp'].jpg )
panggil fungsi form_insert_lokasi()
menampilkan halaman tambah post

```

5.1.4.5 Perancangan Komponen Klas Notif

Nama operasi : notif()

Algoritma :

```

inisialisasi $q = query select notif
data['query'] = hasil query $q
menampilkan halaman notifikasi

```

5.1.4.6 Perancangan Komponen Klas Pos

Nama operasi : kirimpos()

Algoritma :

```

set form validation Keterangan
set form validation Harga
if(form_validation = false) then
    data['listwisata'] = panggil fungsi viewlokasi()
    inisialisasi $pstamp = query * from pos_temp where user = session-
    >userdata('Username')
    data['Foto'] = hasil query $pstamp
    menampilkan halaman tambah post
else
    inisialisasi $q = query select * from lokasi where id = input-
    >post('listwisata')
    inisialisasi $rop = hasil query $q

```

```

inialisasi $postmp = query select * from pos_temp where user = session-
>userdata('username') and lat = 0
inialisasi $fotoo = hasil query postmp
if(jumlah row $pstmp > 0)
    menampilkan pemberitahuan lokasi tidak terdeteksi. Silahkan upload
    lagi.
else
    inialisasi $time = waktu
    inialisasi $data = array (
        Wisata = lokasi $rop,
        Lokasi = alamat $rop,
        Keterangan = input->post('Keterangan')
        Foto = input->post('Foto')
        Harga = input->post('Harga')
        Lat = input->post('Lat')
        Long = input->post('Long')
        Waktu = $time
        User = session->userdata('Username') )
    panggil fungsi form_insert_post()
    panggil fungsi geo()
    panggil fungsi postempdel()
    $_SESSION['Stamp']==""
    menampilkan halaman post terbaru
end if
end if

```

5.1.4.7 Perancangan Komponen Klas Profil

Nama operasi : profilupdate()

Algoritma :

```

data['query'] = panggil fungsi profilupdate()
menampilkan profil pengguna

```

5.1.4.8 Perancangan Komponen Klas User

Nama operasi : formdaftar()

Algoritma :

```
set form validation Username
set form validation Password
set form validation Nama
set form validation Alamat
set form validation Kota
set form validation Telepon
if(form_validation = false) then
    menampilkan halaman daftar
else
    inialisasi $data = array (
        Username = input->post(' Username'),
        Password = input->post('Password'),
        Nama = input->post('Nama')
        Alamat = input->post('Alamat')
        Kota = input->post('Kota')
        Telepon = input->post('Telepon')
    panggil fungsi form_insert()
end if
```

5.1.4.9 Perancangan Komponen Klas Admin

Nama operasi : blokir()

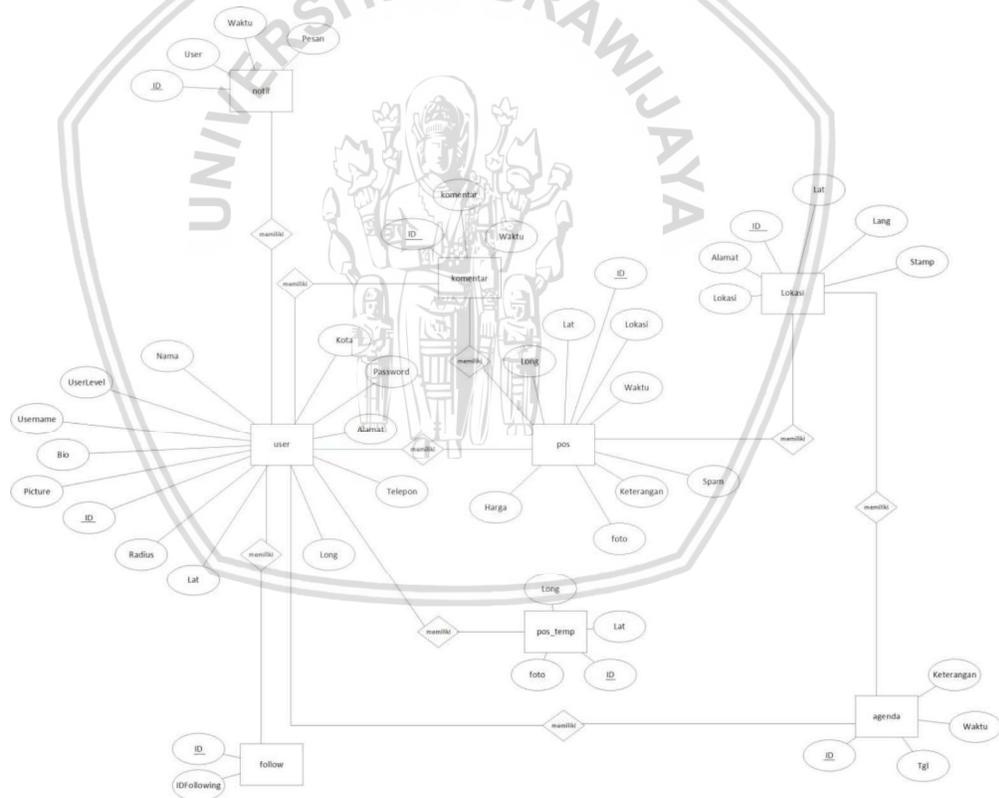
Algoritma :

```
If (sts = 1) then
    panggil fungsi blokir()
else
    panggil fungsi update()
end if
menampilkan halaman laporan
```

5.1.5 Perancangan Data

Perancangan data digambarkan dengan *Entity Relationship Diagram (ERD)*. Penentuan entitas, atribut, dan relasi disesuaikan dengan kebutuhan sistem yang akan di bangun. Terdapat 8 entitas yang menjadi komponen ERD aplikasi *social geotagging* destinasi wisata yaitu user, komentar, lokasi, pos, follow, agenda, pos_temp, dan notifikasi.

Entitas user menyimpan data pengguna yang sudah melakukan registrasi ke dalam aplikasi. Entitas komentar menyimpan data komentar dari post. Entitas lokasi menyimpan data lokasi yang ditambahkan. Entitas pos menyimpan data post yang dibagikan oleh pengguna. Entitas follow menyimpan data pengguna yang mengikuti dan diikuti. Entitas pos_temp menyimpan foto dan lokasinya yang akan di tambahkan oleh pengguna. Entitas agenda menyimpan data agenda yang tambahkan oleh pengguna. Gambar 5.9 menjelaskan representasi ERD dari aplikasi yang akan di bangun.



Gambar 5.9 Entity Relationship Diagram

Berdasarkan perancangan data yang dibuat dengan menggunakan *Entity Relationship Diagram (ERD)* yang digambarkan pada Gambar 5.9, maka dapat dijelaskan struktur dari tabel-tabel yang akan diterapkan. Tabel-tabel tersebut diantaranya adalah user, komentar, lokasi, pos, follow, agenda, pos_temp, dan notif.

a. Struktur Tabel User

Entitas user merupakan representasi dari tabel user di dalam *database* aplikasi. Tabel user menyimpan data pengguna yang telah melakukan registrasi pada aplikasi dan juga menyimpan perubahan data diri pengguna ketika pengguna melakukan edit profil.

Tabel 5.1 Struktur Tabel User

Nama Field	Tipe Data	Keterangan
ID	Int	Id pengguna
Picture	Varchar	Foto profil pengguna
Bio	Text	Biodata pengguna
Username	Varchar	Username pengguna
Password	Varchar	Password pengguna
UserLevel	Int	Hak akses pengguna
Nama	Varchar	Nama pengguna
Alamat	Varchar	Alamat pengguna
Kota	Varchar	Kota tempat tinggal pengguna
Telepon	Varchar	Nomor telepon pengguna
Lat	Varchar	Koordinat garis bujur dari lokasi pengguna
Long	Varchar	Koordinat garis lintang dari lokasi pengguna
Radius	Int	Radius untuk pencarian tempat disekitar pengguna

b. Struktur Tabel Komentar

Entitas komentar merupakan representasi dari tabel komentar di dalam *database* aplikasi. Tabel komentar menyimpan data komentar yang diberikan user kepada suatu post.

Tabel 5.2 Struktur Tabel Komentar

Nama Field	Tipe Data	Keterangan
ID	Int	Id komentar
Komentar	Text	Isi komentar
Waktu	Datetime	Waktu komentar dikirim

c. Struktur Tabel Lokasi

Entitas lokasi merupakan representasi dari tabel lokasi di dalam *database* aplikasi. Tabel lokasi menyimpan data lokasi baru yang ditambahkan oleh pengguna ketika melakukan tambah post.

Tabel 5.3 Struktur Tabel Lokasi

Nama Field	Tipe Data	Keterangan
ID	Int	Id lokasi
Lokasi	Varchar	Nama lokasi
Alamat	Varchar	Alamat Lokasi
Lat	Varchar	Koordinat garis bujur dari lokasi
Lang	Varchar	Koordinat garis lintang dari lokasi
Stamp	Varchar	Nama foto yang menyimpan lokasi ketika ditambahkan

d. Struktur Tabel Pos

Entitas pos merupakan representasi dari tabel pos di dalam *database* aplikasi. Tabel pos menyimpan data post yang ditambahkan oleh pengguna ketika melakukan tambah post.

Tabel 5.4 Struktur Tabel Pos

Nama Field	Tipe Data	Keterangan
ID	Varchar	Id post
Lokasi	Varchar	Alamat lokasi post
Lat	Datetime	Koordinat garis bujur dari lokasi post
Long	Varchar	Koordinat garis lintang dari lokasi post
Keterangan	Varchar	Keterangan dari post
Foto	Varchar	Nama foto post
Harga	Varchar	Harga tiket tempat yang ada pada post
Waktu	Datetime	Waktu post dikirim
Spam	Varchar	Status laporan post

e. Struktur Tabel Follow

Entitas follow merupakan representasi dari tabel follow di dalam *database* aplikasi. Tabel follow menyimpan data ketika pengguna mengikuti akun pengguna lainnya.

Tabel 5.5 Struktur Tabel Follow

Nama Field	Tipe Data	Keterangan
ID	Int	Id follow
IDFollowing	Varchar	Username pengguna yang diikuti

f. Struktur Tabel Agenda

Entitas agenda merupakan representasi dari tabel agenda di dalam *database* aplikasi. Tabel agenda menyimpan data agenda yang ditambahkan oleh pengguna.

Tabel 5.6 Struktur Tabel Agenda

Nama Field	Tipe Data	Keterangan
ID	Int	Id agenda
Keterangan	Text	Keterangan agenda
Tgl	Date	Tanggal agenda di simpan
Waktu	Datetime	Waktu agenda di simpan

g. Struktur Tabel Pos_temp

Entitas pos_temp merupakan representasi dari tabel pos_temp di dalam *database* aplikasi. Tabel pos_temp menyimpan data lokasi foto yang akan ditambahkan oleh pengguna secara sementara.

Tabel 5.7 Struktur Tabel Pos_temp

Nama Field	Tipe Data	Keterangan
ID	Int	Id post sementara
Lat	Varchar	Koordinat garis bujur dari lokasi post sementara
Long	Varchar	Koordinat garis lintang dari lokasi post sementara
Foto	Varchar	Nama foto post sementara



h. Struktur Tabel Notif

Entitas notif merupakan representasi dari tabel notif di dalam *database* aplikasi. Tabel notif menyimpan data notifikasi ketika ada post yang di berikan komentar, post dihapus oleh admin, dan jika diikuti oleh pengguna lain.

Tabel 5.8 Struktur Tabel Notif

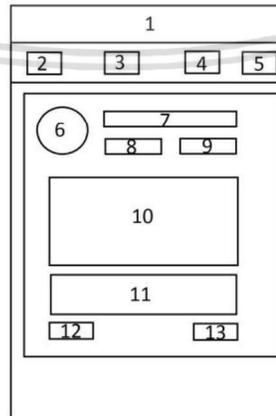
Nama Field	Tipe Data	Keterangan
ID	Int	Id notifikasi
User	Varchar	Username pengguna yang melakukan kegiatan sehingga ada notifikasi
Pesan	Text	Pesan notifikasi
Waktu	Datetime	Waktu notifikasi dibuat

5.1.6 Perancangan Antarmuka

Perancangan antarmuka dilakukan untuk menggambarkan rancangan tampilan antarmuka yang akan diterapkan ke dalam sistem yang akan dibuat. Dalam perancangan antarmuka akan ditampilkan beberapa sampel antarmuka yaitu post terbaru, lokasi terdekat, daftar agenda, daftar post berdasarkan lokasi, dan daftar post yang di laporkan.

5.1.6.1 Perancangan Antarmuka Post Terbaru

Penjelasan perancangan antarmuka lihat post terbaru terdapat pada Gambar 5.10, dan Penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.9. Antarmuka ini menampilkan post terbaru yang dibagikan oleh pengguna.



Gambar 5.10 Perancangan Antarmuka Post Terbaru

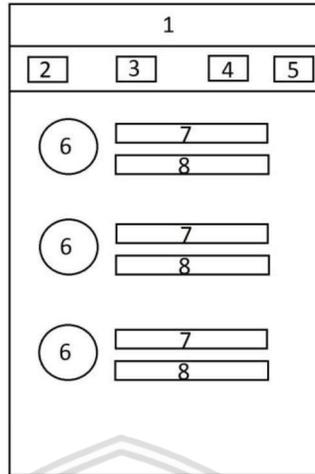


Tabel 5.9 Penjelasan Antarmuka Post Terbaru

NO	Nama Objek	Keterangan
1	Nama aplikasi	Menampilkan nama aplikasi
2	Tombol home	Tombol untuk menuju halaman home yang berisi menu utama
3	Tombol tambah post	Tombol untuk menuju halaman tambah post
4	Tombol tambah agenda	Tombol untuk menuju halaman tambah agenda
5	Foto profil pengguna	Menampilkan foto profil pengguna yang sedang log in
6	Icon marker	Menampilkan icon marker
7	Nama lokasi	Menampilkan lokasi post
8	<i>Username</i> pengguna	Menampilkan <i>username</i> pengguna yang membagikan post
9	Tanggal	Menampilkan tanggal post dibagikan
10	Foto	Menampilkan foto post
11	Keterangan post	Menampilkan alamat, keterangan, dan harga tiket dalam post
12	Tombol Komentar	Tombol untuk melihat komentar
13	Tombol lapor	Tombol untuk melaporkan post

5.1.6.2 Perancangan Antarmuka Lokasi Terdekat

Penjelasan perancangan antarmuka lihat lokasi terdekat terdapat pada Gambar 5.11, dan Penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.10. Antarmuka ini menampilkan daftar lokasi terdekat dari posisi pengguna.



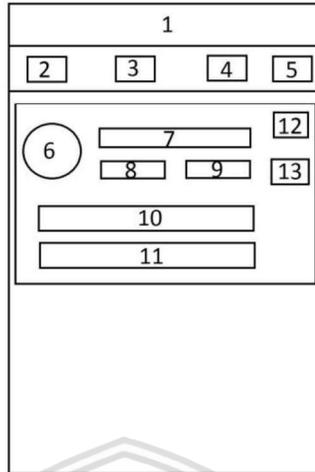
Gambar 5.11 Perancangan Antarmuka Lokasi Terdekat

Tabel 5.10 Penjelasan Antarmuka Lokasi Terdekat

NO	Nama Objek	Keterangan
1	Nama aplikasi	Menampilkan nama aplikasi
2	Tombol home	Tombol untuk menuju halaman home yang berisi menu utama
3	Tombol tambah post	Tombol untuk menuju halaman tambah post
4	Tombol tambah agenda	Tombol untuk menuju halaman tambah agenda
5	Foto profil pengguna	Menampilkan foto profil pengguna yang sedang log in
6	Icon marker	Menampilkan icon marker
7	Nama lokasi	Menampilkan nama lokasi
8	Jarak lokasi	Menampilkan Jarak lokasi dari jarak pengguna

5.1.6.3 Perancangan Antarmuka Daftar Agenda

Penjelasan perancangan antarmuka daftar agenda terdapat pada Gambar 5.12, dan Penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.11. Antarmuka ini menampilkan daftar agenda yang dimiliki oleh pengguna.



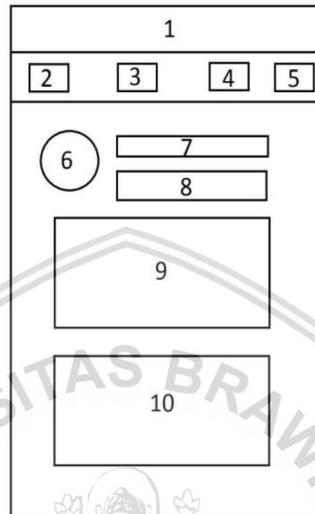
Gambar 5.12 Perancangan Antarmuka Daftar Agenda

Tabel 5.11 Penjelasan Antarmuka Daftar Agenda

NO	Nama Objek	Keterangan
1	Nama aplikasi	Menampilkan nama aplikasi
2	Tombol home	Tombol untuk menuju halaman home yang berisi menu utama
3	Tombol tambah post	Tombol untuk menuju halaman tambah post
4	Tombol tambah agenda	Tombol untuk menuju halaman tambah agenda
5	Foto profil pengguna	Menampilkan foto profil pengguna yang sedang log in
6	Icon marker	Menampilkan icon marker
7	Nama lokasi	Menampilkan lokasi agenda
8	<i>Username</i> pengguna	Menampilkan <i>username</i> pengguna yang membuat agenda
9	Tanggal agenda dibuat	Menampilkan tanggal agenda dibuat
10	Tanggal rencana	Menampilkan tanggal yang direncanakan dalam agenda
11	Keterangan	Menampilkan keterangan agenda
12	Tombol edit	Tombol untuk menuju halaman edit agenda
13	Tombol hapus	Tombol untuk menghapus agenda

5.1.6.4 Perancangan Antarmuka Daftar Post Berdasarkan Lokasi

Penjelasan perancangan antarmuka daftar post berdasarkan lokasi terdapat pada Gambar 5.13, dan Penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.12. Antarmuka ini menampilkan daftar post dengan lokasi yang sama.



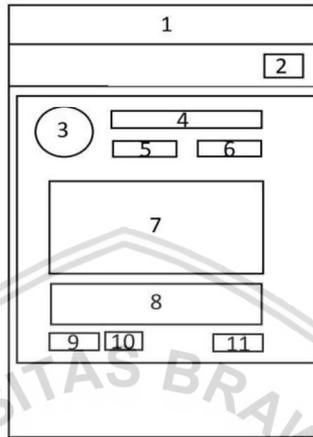
Gambar 5.13 Perancangan Antarmuka Daftar Post Berdasarkan Lokasi

Tabel 5.12 Penjelasan Antarmuka Daftar Post Berdasarkan Lokasi

NO	Nama Objek	Keterangan
1	Nama aplikasi	Menampilkan nama aplikasi
2	Tombol home	Tombol untuk menuju halaman home yang berisi menu utama
3	Tombol tambah post	Tombol untuk menuju halaman tambah post
4	Tombol tambah agenda	Tombol untuk menuju halaman tambah agenda
5	Foto profil pengguna	Menampilkan foto profil pengguna yang sedang log in
6	Icon marker	Menampilkan icon marker
7	Nama lokasi	Menampilkan lokasi
8	Alamat	Menampilkan alamat lokasi
9	Peta	Menampilkan posisi lokasi di dalam peta
10	Daftar post	Menampilkan daftar post pada lokasi terkait

5.1.6.5 Perancangan Antarmuka Daftar Laporan

Penjelasan perancangan antarmuka daftar laporan terdapat pada Gambar 5.14, dan Penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada Tabel 5.13. Antarmuka ini menampilkan daftar post yang dilaporkan oleh pengguna.



Gambar 5.14 Perancangan Antarmuka Daftar Laporan

Tabel 5.13 Penjelasan Antarmuka Daftar Laporan

NO	Nama Objek	Keterangan
1	Nama aplikasi	Menampilkan nama aplikasi
2	Tombol log out	Tombol untuk melakukan log out
3	Icon marker	Menampilkan icon marker
4	Nama lokasi	Menampilkan lokasi post
5	Username pengguna	Menampilkan <i>username</i> pengguna yang membagikan post
6	Icon marker	Menampilkan icon marker
7	Foto	Menampilkan foto post
8	Keterangan post	Menampilkan alamat, keterangan, dan harga tiket dalam post
9	Alasan laporan	Menampilkan alasan laporan dilaporkan
10	Tombol hapus	Tombol untuk menghapus laporan dengan pilihan hapus total dan hapus laporan
11	Tombo Komentar	Tombol Tombol untuk melihat komentar

5.2 Implementasi Sistem

Implementasi sistem dilaksanakan setelah tahap perancangan telah selesai. Implementasi dilakukan berdasarkan hasil yang telah didapatkan dalam tahap analisis kebutuhan serta tahap perancangan sistem. Setiap kebutuhan yang telah dideskripsikan pada tahap sebelumnya akan diimplementasikan ke dalam sistem yang dibangun. Implementasi harus membuat struktur serta kode-kode program yang didasarkan oleh *class* dan algoritma yang telah dibuat pada tahap perancangan. Dalam Implementasi sistem dijelaskan spesifikasi sistem, batasan implementasi, implementasi *database*, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Spesifikasi dari sistem yang digunakan dalam membangun sistem ini terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.2.1.1 Spesifikasi Perangkat Keras

Pembangunan aplikasi *social geotagging* destinasi wisata menggunakan perangkat keras berupa komputer yang memiliki spesifikasi seperti dijelaskan pada Tabel 5.14 berikut.

Tabel 5.14 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
System Model	Asus X450J
Processor	Intel® Core(TM) i7-4710HQ CPU @ 2,50 GHz
Memory	4.00 GB RAM
Display	Intel® HD Graphics

Perangkat yang digunakan dalam proses instalasi dan pengujian sistem adalah perangkat *smartphone* Android yang memiliki spesifikasi perangkat keras seperti yang ditunjukkan pada Tabel 5.15 berikut.

Tabel 5.15 Spesifikasi Perangkat Keras Smartphone Android

Nama Komponen	Spesifikasi
System model	Asus Zenfone 5
Processor	Intel Z2560
Memory	8 GB
Display	5.0" Super IPS, 1280x720 HD

5.2.1.2 Spesifikasi Perangkat Lunak

Pembuatan aplikasi *social geotagging* destinasi wisata ini menggunakan sebuah unit komputer dengan spesifikasi perangkat lunak seperti pada Tabel 5.16 berikut.

Tabel 5.16 Spesifikasi Perangkat Lunak Komputer

Nama Komponen	Spesifikasi
Operating System	Windows 8.1 Professional 64 Bit
Bahasa Pemrograman	HTML, CSS, JavaScript, PHP dan MySQL
IDE (Integrated Development Environment)	Eclipse Juno dengan ADT (<i>Android Development Tools</i>) plugin

Perangkat yang digunakan dalam proses instalasi dan pengujian sistem adalah perangkat *smartphone* Android dengan spesifikasi perangkat lunak seperti yang ditunjukkan pada Tabel 5.17 berikut.

Tabel 5.17 Spesifikasi Perangkat Lunak *Smartphone* Android

Nama Komponen	Spesifikasi
<i>Platform</i>	Android versi 4.3 Android Browser HSDPA, 42.2 Mbps/ HSUPA, 5.76 Mbps

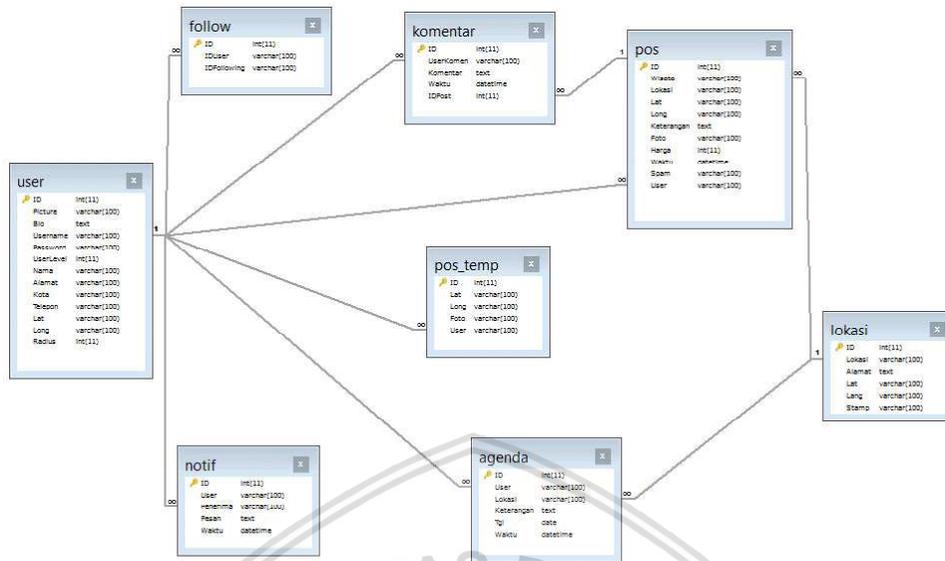
5.2.2 Batasan Implementasi

Pada implementasi sistem terdapat batasan-batasan dalam proses yaitu sebagai berikut :

1. Aplikasi *social geotagging* destinasi wisata dirancang untuk dapat dijalankan pada *smartphone* dengan sistem operasi *mobile* Android dengan menggunakan konsep *hybrid*.
2. Aplikasi dijalankan dengan menggunakan koneksi internet.
3. Aplikasi menggunakan *framework* PhoneGap dalam pembuatan aplikasi *mobile hybrid*.
4. Penyimpanan data menggunakan *database* MySQL.
5. Pembuatan *layout* pada *user interface* aplikasi menggunakan bahasa pemrograman HTML, CSS, JavaScript, dan menggunakan *framework* Bootstrap.

5.2.3 Implementasi *Database*

Implementasi *database* dibuat berdasarkan hasil dari perancangan data yang telah dilakukan sebelumnya. Implementasi *database* digambarkan pada Gambar 5.14



Gambar 5.15 Implementasi Database

Berdasarkan hasil implementasi *database* yang digambarkan pada Gambar 5.15, maka didapatkan tabel-tabel dari *database* yang digunakan dalam pembangunan sistem. Tabel-tabel pada *database* tersebut dibuat berdasarkan hasil dari perancangan data yang telah dilakukan sebelumnya. Dalam tabel-tabel tersebut terlihat bahwa ada beberapa tabel yang memiliki hubungan satu sama lain. Tabel-tabel yang memiliki hubungan tersebut menghasilkan *foreign key*. Contoh dari tabel yang memiliki hubungan yaitu tabel *pos* yang memiliki hubungan dengan tabel *komentar* dan menghasilkan suatu *foreign key* "IDPost" yang terdapat dalam tabel *komentar*. Dari hubungan tersebut dapat dijelaskan bahwa satu field dari tabel *pos* bisa dimiliki oleh banyak field dari tabel *komentar*.

5.2.4 Implementasi Antarmuka

Implementasi antarmuka ini dilakukan berdasarkan hasil dari perancangan antarmuka yang dibuat dalam tahap perancangan. Berikut ini adalah contoh halaman antarmuka pada aplikasi *social geotagging* destinasi wisata.

5.2.4.1 Implementasi Antarmuka Post Terbaru

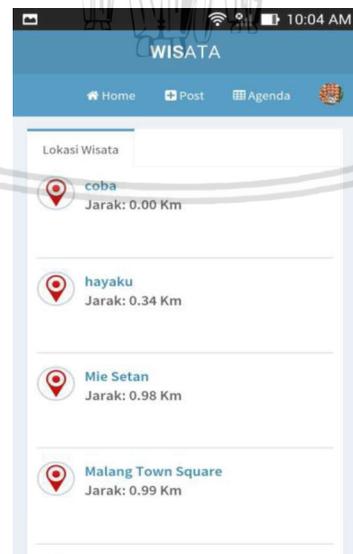
Pada Implementasi antarmuka post terbaru yang digambarkan pada Gambar 5.16 menjelaskan bahwa dalam halaman post terbaru ditampilkan daftar post terbaru berupa foto, nama lokasi post, *username* pengguna yang membagikan post, alamat, keterangan, dan harga tiket dari wisata yang dibagikan dalam post. Selain itu disetiap post terdapat tombol lapor dan komentar. Tombol lapor digunakan untuk melaporkan post yang tidak sesuai dengan aplikasi. Tombol komentar digunakan untuk melihat dan menambahkan komentar.



Gambar 5.16 Implementasi Antarmuka Post Terbaru

5.2.4.2 Implementasi Antarmuka Lokasi Terdekat

Pada Implementasi antarmuka lokasi terdekat yang digambarkan pada Gambar 5.17 menjelaskan bahwa dalam halaman lokasi terdekat ditampilkan daftar lokasi terdekat dari posisi user berada dalam bentuk nama lokasi dan jarak lokasi dari posisi pengguna



Gambar 5.17 Implementasi Antarmuka Lokasi Terdekat

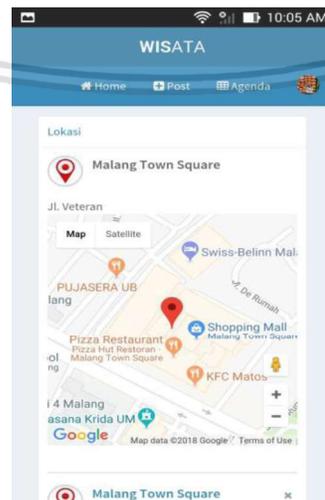
5.2.4.3 Implementasi Antarmuka Daftar Agenda

Pada Implementasi antarmuka daftar agenda yang digambarkan pada Gambar 5.18 menjelaskan bahwa dalam daftar agenda ditampilkan daftar agenda yang dimiliki pengguna berupa nama lokasi agenda, *username* pengguna, tanggal, dan keterangan agenda. Selain itu, pada setiap agenda terdapat tombol edit dan hapus. Tombol edit digunakan untuk melakukan edit agenda. Tombol hapus digunakan untuk menghapus agenda.



Gambar 5.18 Implementasi Antarmuka Daftar Agenda

5.2.4.4 Implementasi Antarmuka Daftar Post Berdasarkan Lokasi



Gambar 5.19 Implementasi Antarmuka Daftar Post Berdasarkan Lokasi

Pada Implementasi antarmuka daftar post berdasarkan lokasi yang digambarkan pada Gambar 5.19 menjelaskan bahwa dalam halaman daftar post berdasarkan lokasi ditampilkan nama dan alamat lokasi, posisi lokasi di dalam peta, dan daftar pos berdasarkan lokasi terkait.

5.2.4.5 Implementasi Antarmuka Daftar Laporan

Pada Implementasi antarmuka daftar laporan yang digambarkan pada Gambar 5.20 dijelaskan bahwa pada halaman daftar laporan ditampilkan daftar post yang dilaporkan. Laporan yang ditampilkan adalah *username* pengguna yang membagikan post, lokasi post, alamat, keterangan, dan harga tiket wisata yang dibagikan dalam post. Pada setiap post yang di laporkan terdapat tombol hapus untuk menghapus post dari sistem atau menghapus post dari daftar laporan.



Gambar 5.20 Implementasi Antarmuka Daftar Laporan

BAB 6 PENGUJIAN SISTEM

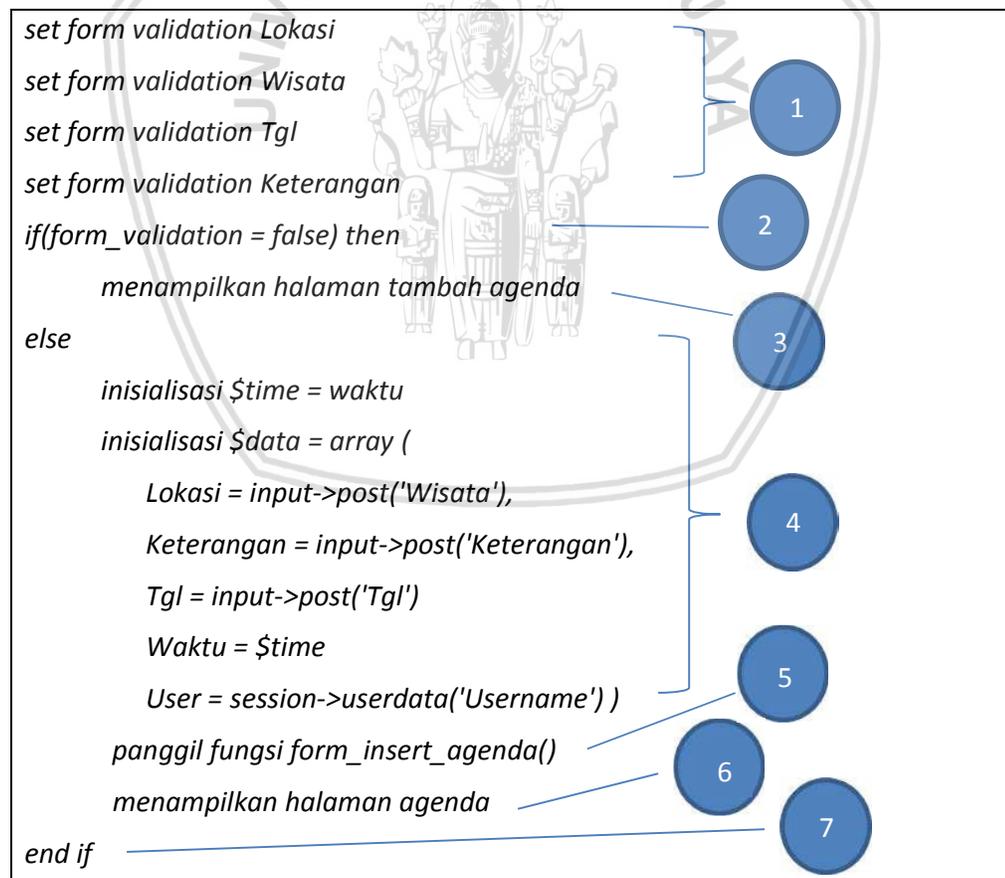
Tahap pengujian sistem dilaksanakan setelah tahap implementasi sudah selesai dilakukan. Tahap Pengujian memiliki tujuan untuk memeriksa apakah hasil dari implementasi sistem telah sesuai dengan hasil dari analisis kebutuhan serta perancangan sistem atau tidak. Tahap dalam proses pengujian yang dilakukan dalam penelitian ini adalah pengujian unit, pengujian validasi, serta pengujian *compatibility*.

6.1 Pengujian Unit

Pengujian unit dilakukan untuk menguji suatu unit dari *class* yang telah diimplementasikan dalam sistem. Pengujian unit dilakukan dengan metode *whitebox* testing dengan menerapkan jenis pengujian basis path. Pengujian unit dilakukan terhadap tiga sampel uji, diantaranya operasi *kirimagenda()* dari kelas *Agenda*, operasi *kirimpos()* dari kelas *Pos*, dan operasi *blokir()* dari kelas *Admin*.

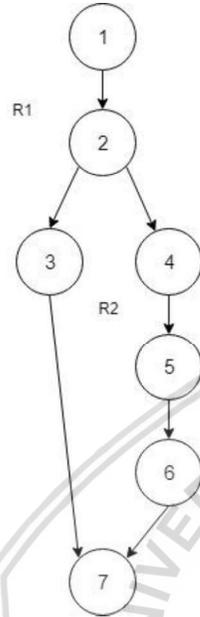
6.1.1 Pengujian Unit Klas *Agenda* Untuk Operasi *kirimagenda()*

1. Pseudocode



2. Basis Path Testing

a. Flowgraph



b. Cyclomatic Complexity

$V(G) = 2$ region (R1 dan R2)
 $V(G) = 7$ edges $- 7$ nodes $+ 2 = 2$
 $V(G) = 1$ predicate nodes $+ 1 = 2$

c. Independent Path

Jalur 1 = 1-2-3-7
 Jalur 2 = 1-2-4-5-6-7

Test case serta hasil akan dijelaskan pada Tabel 6.1

Tabel 6.1 Pengujian Unit Klas Agenda Operasi kirimagenda()

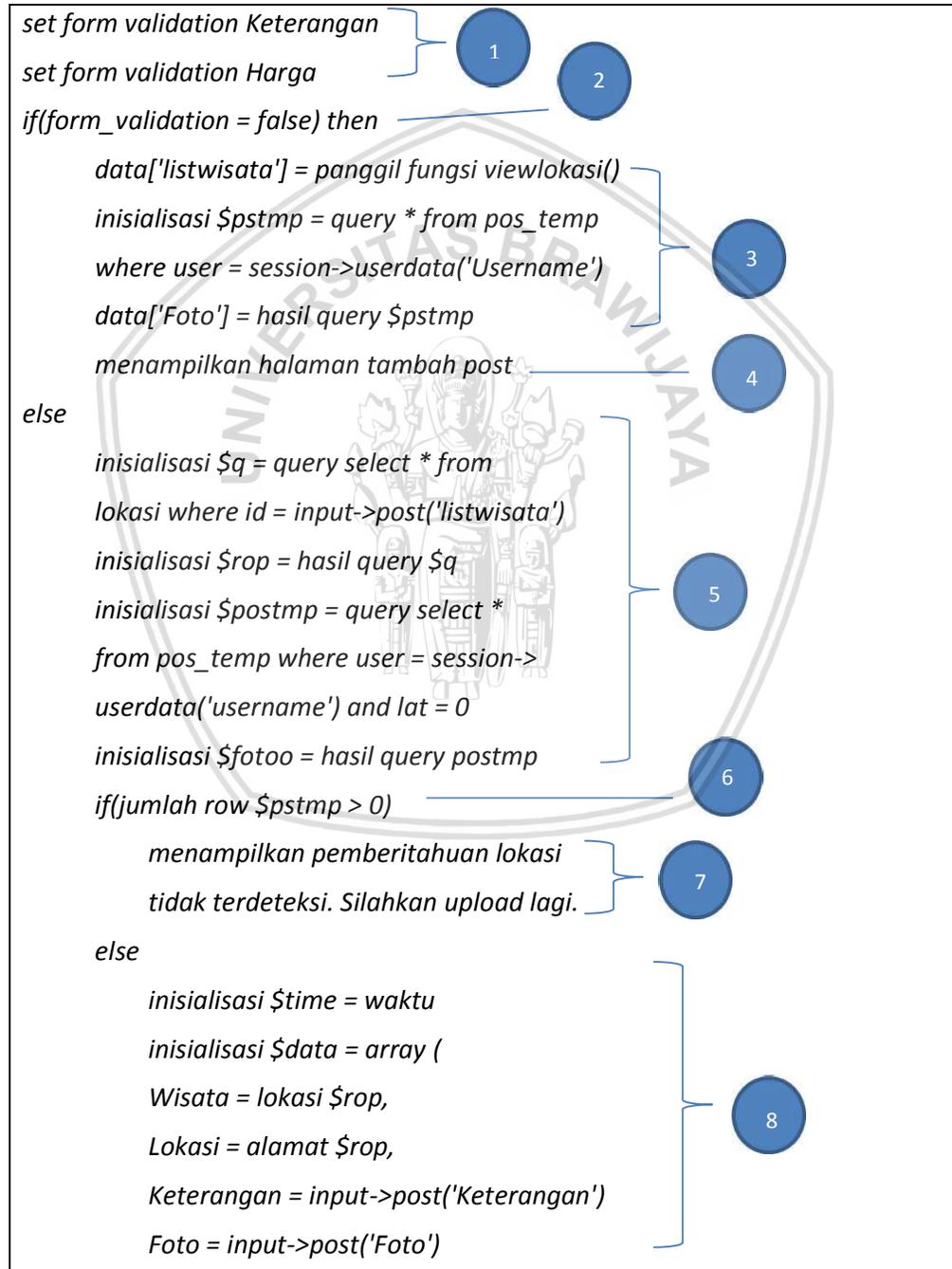
NO	Nomor Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil	Status
1	1	Menjalankan operasi kirimagenda() dengan status <i>form_validation = false</i>	Menampilkan halaman tambah agenda	Menampilkan halaman tambah agenda	Valid
2	2	Menjalankan operasi	Agenda tersimpan,	Agenda tersimpan,	Valid

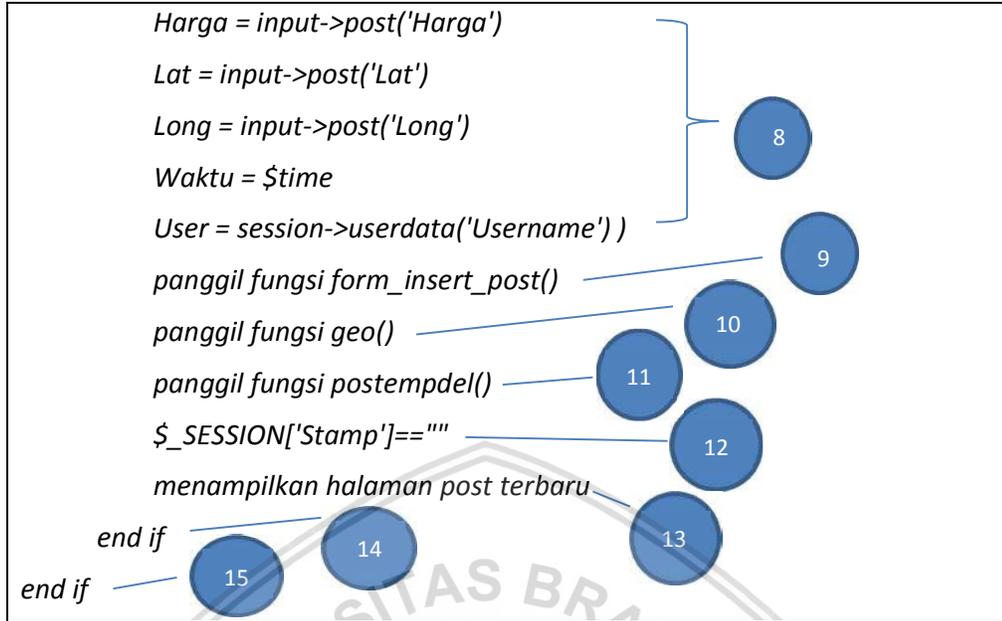


		kirimagenda() dengan status <i>form_validation</i> = true	menampilkan daftar agenda	menampilkan daftar agenda	
--	--	--	---------------------------	---------------------------	--

6.1.2 Pengujian Unit Klas Pos Untuk Operasi kirimpos()

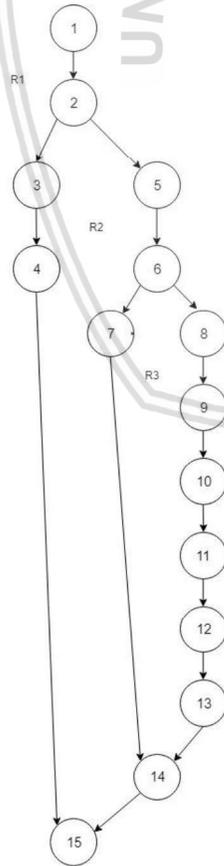
1. Pseudocode





2. Basis Path Testing

a. Flowgraph



b. Cyclomatic Complexity

$$V(G) = 3 \text{ region (R1, R2, dan R3)}$$

$$V(G) = 16 \text{ edges} - 15 \text{ nodes} + 2 = 3$$

$$V(G) = 2 \text{ predicate nodes} + 1 = 3$$

c. Independent Path

Jalur 1 = 1-2-3-4-15

Jalur 2 = 1-2-5-6-7-14-15

Jalur 3 = 1-2-5-6-8-9-10-11-12-13-14-15

Test case serta hasil akan dijelaskan pada Tabel 6.2

Tabel 6.2 Pengujian Unit Klas Pos Operasi kirimpos()

NO	Nomor Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil	Status
1	1	Menjalankan operasi kirimpos() dengan status <i>form_validation = false</i>	Menampilkan halaman tambah post	Menampilkan halaman tambah post	Valid
2	2	Menjalankan operasi kirimpos() dengan status <i>form_validation = true</i> dan nilai <i>row pstmp = 1</i>	Menampilkan pemberitahuan lokasi tidak terdeteksi. Silahkan upload lagi.	Menampilkan pemberitahuan lokasi tidak terdeteksi. Silahkan upload lagi.	Valid
3	3	Menjalankan operasi kirimpos() dengan status <i>form_validation = true</i> dan nilai <i>row pstmp = 0</i>	Post tersimpan, menampilkan halaman post terbaru	Post tersimpan, menampilkan halaman post terbaru	Valid

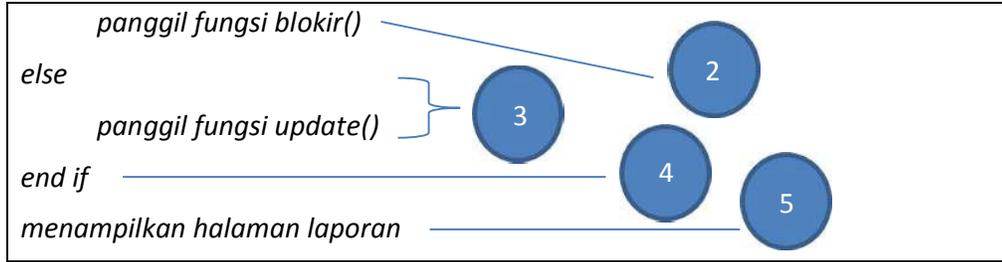
6.1.3 Pengujian Unit Klas Admin Untuk Operasi blokir()

1. Pseudocode

```

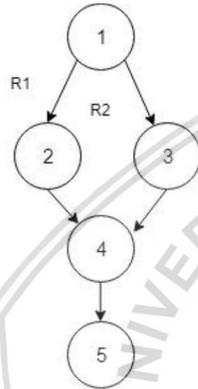
If (sts = 1) then
    _____
    
```





2. Basis Path Testing

a. Flowgraph



b. Cyclomatic Complexity

$V(G) = 2$ region (R1 dan R2)

$V(G) = 5$ edges - 5 nodes + 2 = 2

$V(G) = 1$ predicate nodes + 1 = 2

c. Independent Path

Jalur 1 = 1-2-4-5

Jalur 2 = 1-3-4-5

Test case serta hasil akan dijelaskan pada Tabel 6.3

Tabel 6.3 Pengujian Unit Klas Admin Operasi blokir()

NO	Nomor Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil	Status
1	1	Menjalankan operasi blokir() dengan nilai sts = 1	Post dihapus dari sistem, menampilkan daftar post	Post dihapus dari sistem, menampilkan daftar post	Valid
2	2	Menjalankan operasi blokir() dengan nilai sts	Post dihapus dari daftar laporan,	Post dihapus dari daftar laporan,	Valid



		= 2	menampilkan daftar post	menampilkan daftar post	
--	--	-----	-------------------------	-------------------------	--

6.2 Pengujian Validasi

Pengujian validasi merupakan tahap yang dilakukan untuk memastikan seluruh kebutuhan yang sudah di definisikan sebelumnya pada tahap analisis kebutuhan telah diimplemenasikan ke dalam sistem yang sudah dibangun. Pengujian validasi dilakukan dengan cara memeriksa fungsi-fungsi pada sistem apakah sudah berjalan dengan baik dan tidak terjadi kesalahan. Pengujian validasi menggunakan metode pengujian *Black-Box*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan.

6.2.1 Pengujian Validasi Register

- a. Kasus uji berhasil register

Tabel 6.4 Kasus Uji Berhasil Register

Nama Kasus Uji	Kasus uji berhasil register
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman login aplikasi pengguna 2. Memilih menu daftar 3. Mengisi kolom username = uji, password = uji, nama = coba, alamat = sigura-gura, kota = malang, telepon = 08123456789 4. Menekan tombol daftar
Hasil yang diharapkan	Data user baru tersimpan dan menampilkan halaman login
Hasil	Data user baru tersimpan dan menampilkan halaman login
Status	Valid

- b. Kasus uji gagal register 1

Tabel 6.5 Kasus Uji Gagal Register 1

Nama Kasus Uji	Kasus uji gagal register 1
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman login aplikasi pengguna 2. Memilih menu daftar 3. Mengisi kolom username = uji, password = uji, alamat = sigura-gura, kota = malang, telepon = 081123456789, nama tidak diisi



	4. Menekan tombol daftar
Hasil yang diharapkan	Menampilkan pemberitahuan “ <i>The field is required</i> ”
Hasil	Menampilkan pemberitahuan “ <i>The field is required</i> ”
Status	Valid

c. Kasus uji gagal register 2

Tabel 6.6 Kasus Uji Gagal Register 2

Nama Kasus Uji	Kasus uji gagal register 2
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman login aplikasi pengguna 2. Memilih menu daftar 3. Mengisi kolom username = uji, password = uji, nama = uji, alamat = sigura-gura, kota = malang, telepon = 081123456789, nama tidak diisi 4. Menekan tombol daftar
Hasil yang diharapkan	Menampilkan pemberitahuan “Username tidak tersedia”
Hasil	Menampilkan pemberitahuan “Username tidak tersedia”
Status	Valid

6.2.2 Pengujian Validasi Login

a. Kasus uji berhasil login

Tabel 6.7 Kasus Uji Berhasil Login

Nama Kasus Uji	Kasus uji berhasil login
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman login aplikasi pengguna 2. Mengisi kolom username = uji, password = uji 3. Menekan tombol Login
Hasil yang diharapkan	Sistem akan menampilkan halaman post terbaru
Hasil	Sistem menampilkan halaman post terbaru
Status	Valid

b. Kasus uji gagal login 1

Tabel 6.8 Kasus Uji Gagal Login 1

Nama Kasus Uji	Kasus uji gagal login 1
-----------------------	-------------------------



Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman login aplikasi pengguna 2. Mengisi kolom username = uji, password = 123 3. Menekan tombol <i>Login</i>
Hasil yang diharapkan	Sistem menampilkan pemberitahuan "Username dan password salah"
Hasil	Sistem menampilkan pemberitahuan "Username dan password salah"
Status	Valid

c. Kasus uji gagal *login* 2

Tabel 6.9 Kasus Uji Gagal *Login* 2

Nama Kasus Uji	Kasus uji gagal <i>login</i> 2
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman login aplikasi pengguna 2. Mengisi kolom username = uji, password tidak diisi 3. Menekan tombol <i>Login</i>
Hasil yang diharapkan	Menampilkan pemberitahuan " <i>The field is required</i> "
Hasil	Menampilkan pemberitahuan " <i>The field is required</i> "
Status	Valid

6.2.3 Pengujian Validasi Lihat Post Terbaru

a. Kasus uji lihat post terbaru

Tabel 6.10 Kasus Uji Lihat Post Terbaru

Nama Kasus Uji	Kasus uji lihat post terbaru
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol wisata terbaru
Hasil yang diharapkan	Sistem menampilkan daftar post terbaru
Hasil	Sistem menampilkan daftar post terbaru
Status	Valid

6.2.4 Pengujian Validasi Lihat Profil

a. Kasus uji lihat profil

Tabel 6.11 Kasus Uji Lihat Profil

Nama Kasus Uji	Kasus uji lihat profil
Prosedur	1. Sudah berada dalam kondisi <i>login</i> 2. Menekan tombol profile
Hasil yang diharapkan	Sistem menampilkan profil pengguna
Hasil	Sistem menampilkan profil pengguna
Status	Valid

6.2.5 Pengujian Validasi Tambah Post

a. Kasus uji berhasil tambah post

Tabel 6.12 Kasus Uji Berhasil Tambah Post

Nama Kasus Uji	Kasus uji tambah post
Prosedur	1. Upload foto 2. Memilih lokasi wisata = lapangan rampal 3. Mengisi harga tiket = 0 dan keterangan = seru 4. Menekan tombol post baru
Hasil yang diharapkan	Menampilkan halaman post terbaru dan post ditampilkan
Hasil	Menampilkan halaman post terbaru dan post ditampilkan
Status	Valid

b. Kasus uji gagal tambah post

Tabel 6.13 Kasus Uji Gagal Tambah Post

Nama Kasus Uji	Kasus uji tambah post
Prosedur	1. Upload foto 2. Memilih lokasi wisata = lapangan rampal 3. Menekan tombol post baru
Hasil yang diharapkan	Menampilkan pemberitahuan " <i>The field is required</i> "
Hasil	Menampilkan pemberitahuan " <i>The field is required</i> "

Status	Valid
---------------	-------

6.2.6 Pengujian Validasi Ambil Foto Dari Kamera

- a. Kasus uji ambil foto dari kamera

Tabel 6.14 Kasus Uji Ambil Foto dari Kamera

Nama Kasus Uji	Kasus uji ambil foto dari kamera
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman tambah post 2. Menekan tombol foto kamera 3. Mengambil foto dengan kamera
Hasil yang diharapkan	Menampilkan halaman tambah post dan pemberitahuan sukses upload
Hasil	Menampilkan halaman tambah post dan pemberitahuan sukses upload
Status	Valid

6.2.7 Pengujian Validasi Ambil Foto Dari Galeri

- a. Kasus uji ambil foto dari galeri

Tabel 6.15 Kasus Uji Ambil Foto dari Galeri

Nama Kasus Uji	Kasus uji ambil foto dari galeri
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman tambah post 2. Menekan tombol foto album 3. Memilih foto
Hasil yang diharapkan	Menampilkan halaman tambah post dan pemberitahuan sukses upload
Hasil	Menampilkan halaman tambah post dan pemberitahuan sukses upload
Status	Valid

6.2.8 Pengujian Validasi Tambah Lokasi

- a. Kasus uji tambah lokasi

Tabel 6.16 Kasus Uji Tambah Lokasi

Nama Kasus Uji	Kasus uji tambah lokasi
-----------------------	-------------------------

Prosedur	<ol style="list-style-type: none"> 1. Sudah melakukan upload foto dengan lokasi 2. Menekan tombol tambah lokasi 3. Mengisi nama lokasi = coba dan alamat = coba 4. Menekan tombol lokasi baru
Hasil yang diharapkan	Lokasi disimpan dan menampilkan halaman tambah post
Hasil	Lokasi disimpan dan menampilkan halaman tambah post
Status	Valid

6.2.9 Pengujian Validasi Lihat Lokasi Terdekat

a. Kasus uji berhasil lihat lokasi terdekat

Tabel 6.17 Kasus Uji Berhasil Lihat Lokasi Terdekat

Nama Kasus Uji	Kasus uji berhasil lihat lokasi terdekat
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol wisata terdekat 3. Mengisi radius = 10 4. Menekan tombol pencarian
Hasil yang diharapkan	Sistem menampilkan nama lokasi terdekat dalam jarak 10 km dari posisi pengguna
Hasil	Sistem menampilkan nama lokasi terdekat dalam jarak 10 km dari posisi pengguna
Status	Valid

b. Kasus uji gagal lihat lokasi terdekat

Tabel 6.18 Kasus Uji Gagal Lihat Lokasi Terdekat

Nama Kasus Uji	Kasus uji gagal lihat lokasi terdekat
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol wisata terdekat 3. Mengisi radius tidak diisi 4. Menekan tombol pencarian
Hasil yang diharapkan	Menampilkan pemberitahuan " <i>The field is required</i> "
Hasil	Menampilkan pemberitahuan " <i>The field is required</i> "

Status	Valid
---------------	-------

6.2.10 Pengujian Validasi Lihat Wisata Termurah

- a. Kasus uji lihat wisata termurah

Tabel 6.19 Kasus Uji Lihat Wisata Termurah

Nama Kasus Uji	Kasus uji lihat wisata termurah
Prosedur	1. Sudah masuk ke halaman home 2. Menekan tombol wisata termurah
Hasil yang diharapkan	Sistem menampilkan daftar post yang diurutkan berdasarkan harga terkecil
Hasil	Sistem menampilkan daftar post yang diurutkan berdasarkan harga terkecil
Status	Valid

6.2.11 Pengujian Validasi Lihat Wisata Termahal

- a. Kasus uji lihat wisata termahal

Tabel 6.20 Kasus Uji Lihat Wisata Termahal

Nama Kasus Uji	Kasus uji lihat wisata termahal
Prosedur	1. Sudah masuk ke halaman home 2. Menekan tombol wisata termahal
Hasil yang diharapkan	Sistem menampilkan daftar post yang diurutkan berdasarkan harga terbesar
Hasil	Sistem menampilkan daftar post yang diurutkan berdasarkan harga terbesar
Status	Valid

6.2.12 Pengujian Validasi Lihat Agenda

- a. Kasus uji lihat agenda

Tabel 6.21 Kasus Uji Lihat Agenda

Nama Kasus Uji	Kasus uji lihat agenda
Prosedur	1. Sudah masuk ke halaman home 2. Menekan tombol agenda

Hasil yang diharapkan	Sistem menampilkan daftar agenda yang dimiliki pengguna
Hasil	Sistem menampilkan daftar agenda yang dimiliki pengguna
Status	Valid

6.2.13 Pengujian Validasi Tambah Agenda

a. Kasus uji berhasil tambah agenda

Tabel 6.22 Kasus Uji Berhasil Tambah Agenda

Nama Kasus Uji	Kasus uji berhasil tambah agenda
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi <i>login</i> 2. Menekan tombol agenda 3. Memilih lokasi = fun and fit 4. Mengisi tanggal = 25-06-2018, keterangan = bermain trampoline 5. Menekan tombol simpan agenda
Hasil yang diharapkan	Data agenda tersimpan dan menampilkan halaman daftar agenda
Hasil	Data agenda tersimpan dan menampilkan halaman daftar agenda
Status	Valid

b. Kasus uji gagal tambah agenda

Tabel 6.23 Kasus Uji Gagal Tambah Agenda

Nama Kasus Uji	Kasus uji gagal tambah agenda
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi <i>login</i> 2. Menekan tombol agenda 3. Memilih lokasi = fun and fit 4. Mengisi keterangan = bermain trampoline, tanggal tidak diisi 5. Menekan tombol simpan agenda
Hasil yang diharapkan	Menampilkan pemberitahuan " <i>The field is required</i> "
Hasil	Menampilkan pemberitahuan " <i>The field is required</i> "
Status	Valid

6.2.14 Pengujian Validasi Edit Agenda

a. Kasus uji edit agenda

Tabel 6.24 Kasus Uji Edit Agenda

Nama Kasus Uji	Kasus uji edit agenda
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman daftar agenda 2. Menekan tombol edit pada agenda lokasi = fun and fit, tanggal = 25-06-2018, keterangan = bermain trampolin 3. Mengisi tanggal = 26-07-2018, keterangan = bermain bola 4. Menekan tombol simpan agenda
Hasil yang diharapkan	Data agenda tersimpan dan menampilkan daftar agenda yang dimiliki pengguna
Hasil	Data agenda tersimpan dan menampilkan daftar agenda yang dimiliki pengguna
Status	Valid

6.2.15 Pengujian Validasi Hapus Agenda

a. Kasus uji hapus agenda

Tabel 6.25 Kasus Uji Hapus Agenda

Nama Kasus Uji	Kasus uji hapus agenda
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman daftar agenda 2. Menekan tombol hapus pada agenda lokasi = fun and fit, tanggal = 26-07-2018, keterangan = bermain bola
Hasil yang diharapkan	Data agenda dihapus dan menampilkan daftar agenda yang dimiliki pengguna
Hasil	Data agenda dihapus dan menampilkan daftar agenda yang dimiliki pengguna
Status	Valid

6.2.16 Pengujian Validasi Cari Pengguna

a. Kasus uji berhasil cari pengguna

Tabel 6.26 Kasus Uji Berhasil Cari Pengguna

Nama Kasus Uji	Kasus uji berhasil cari pengguna
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol cari user 3. Mengisi <i>username</i> pengguna = www 4. Menekan tombol pencarian user
Hasil yang diharapkan	Sistem menampilkan pengguna dengan <i>username</i> = www
Hasil	Sistem menampilkan pengguna dengan <i>username</i> = www Sistem menampilkan pengguna dengan <i>username</i> = www
Status	Valid

b. Kasus uji gagal cari pengguna 1

Tabel 6.27 Kasus Uji Gagal Cari Pengguna 1

Nama Kasus Uji	Kasus uji gagal cari pengguna 1
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol cari user 3. <i>Username</i> pengguna tidak diisi 4. Menekan tombol pencarian user
Hasil yang diharapkan	Menampilkan pemberitahuan " <i>The field is required</i> "
Hasil	Menampilkan pemberitahuan " <i>The field is required</i> "
Status	Valid

c. Kasus uji gagal cari pengguna 2

Tabel 6.28 Kasus Uji Gagal Cari Pengguna 2

Nama Kasus Uji	Kasus uji gagal cari pengguna 2
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol cari user 3. Mengisi <i>Username</i> pengguna = abcdefg 4. Menekan tombol pencarian user
Hasil yang diharapkan	Menampilkan halaman cari user

Hasil	Menampilkan halaman cari user
Status	Valid

6.2.17 Pengujian Validasi Lihat Profil Pengguna Lain

- a. Kasus uji lihat profil pengguna lain

Tabel 6.29 Kasus Uji Lihat Profil Pengguna Lain

Nama Kasus Uji	Kasus uji lihat profil pengguna lain
Prosedur	1. Sudah masuk ke halaman hasil pencarian pengguna 2. Menekan <i>username</i> = www
Hasil yang diharapkan	Sistem menampilkan profil pengguna dengan <i>username</i> = www
Hasil	Sistem menampilkan profil pengguna dengan <i>username</i> = www
Status	Valid

6.2.18 Pengujian Validasi Ikuti Pengguna

- a. Kasus uji ikuti pengguna

Tabel 6.30 Kasus Uji Ikuti Pengguna

Nama Kasus Uji	Kasus uji ikuti pengguna
Prosedur	1. Sudah masuk ke halaman profil pengguna lain 2. Menekan tombol <i>follow</i>
Hasil yang diharapkan	Jumlah followers pengguna yang diikuti bertambah
Hasil	Jumlah followers pengguna yang diikuti bertambah
Status	Valid

6.2.19 Pengujian Validasi Berhenti Mengikuti Pengguna

- a. Kasus uji berhenti mengikuti pengguna

Tabel 6.31 Kasus Uji Berhenti Mengikuti Pengguna

Nama Kasus Uji	Kasus uji berhenti mengikuti pengguna
Prosedur	1. Sudah masuk ke halaman profil pengguna lain yang sudah diikuti

	2. Menekan tombol <i>unfollow</i>
Hasil yang diharapkan	Jumlah followers pengguna yang diikuti berkurang
Hasil	Jumlah followers pengguna yang diikuti berkurang
Status	Valid

6.2.20 Pengujian Validasi Cari Lokasi

a. Kasus uji berhasil cari lokasi

Tabel 6.32 Kasus Uji Berhasil Cari Lokasi

Nama Kasus Uji	Kasus uji berhasil cari Lokasi
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol cari lokasi 3. Mengisi lokasi = fun and fit 4. Menekan tombol pencarian
Hasil yang diharapkan	Sistem menampilkan lokasi dengan nama = fun and fit
Hasil	Sistem menampilkan lokasi dengan nama = fun and fit
Status	Valid

b. Kasus uji gagal cari lokasi 1

Tabel 6.33 Kasus Uji Gagal Cari Lokasi 1

Nama Kasus Uji	Kasus uji gagal cari lokasi 1
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol cari lokasi 3. Lokasi tidak diisi 4. Menekan tombol pencarian
Hasil yang diharapkan	Menampilkan pemberitahuan " <i>The field is required</i> "
Hasil	Menampilkan pemberitahuan " <i>The field is required</i> "
Status	Valid

c. Kasus uji gagal cari lokasi 2

Tabel 6.34 Kasus Uji Gagal Cari Lokasi 2

Nama Kasus Uji	Kasus uji gagal cari lokasi 2
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol cari lokasi



	3. Mengisi lokasi = aaaaa 4. Menekan tombol pencarian
Hasil yang diharapkan	Menampilkan pemberitahuan “Lokasi tidak ditemukan”
Hasil	Menampilkan pemberitahuan “Lokasi tidak ditemukan”
Status	Valid

6.2.21 Pengujian Validasi Lihat Post Berdasarkan Lokasi

- a. Kasus uji lihat post berdasarkan lokasi

Tabel 6.35 Kasus Uji Lihat Post Berdasarkan Lokasi

Nama Kasus Uji	Kasus uji lihat profil post berdasarkan lokasi
Prosedur	1. Sudah masuk ke halaman hasil pencarian lokasi 2. Menekan lokasi = fun and fit
Hasil yang diharapkan	Sistem menampilkan daftar post dengan lokasi = fun and fit
Hasil	Sistem menampilkan daftar post dengan lokasi = fun and fit
Status	Valid

6.2.22 Pengujian Validasi Tambah Komentar

- a. Kasus uji tambah komentar

Tabel 6.36 Kasus Uji Tambah Komentar

Nama Kasus Uji	Kasus uji tambah komentar
Prosedur	1. Sudah masuk ke halaman post terbaru 2. Menekan tombol komentar pada post 3. Mengisi komentar = bagus
Hasil yang diharapkan	Sistem menampilkan komentar yang ditambahkan
Hasil	Sistem menampilkan komentar yang ditambahkan
Status	Valid

6.2.23 Pengujian Validasi Lihat Komentar

- a. Kasus uji lihat Komentar

Tabel 6.37 Kasus Uji Lihat Komentar

Nama Kasus Uji	Kasus uji lihat komentar
Prosedur	1. Sudah masuk ke halaman post terbaru 2. Menekan tombol komentar pada post
Hasil yang diharapkan	Sistem menampilkan komentar
Hasil	Sistem menampilkan komentar
Status	Valid

6.2.24 Pengujian Validasi Edit Profil

- a. Kasus uji edit profil

Tabel 6.38 Kasus Uji Edit Profil

Nama Kasus Uji	Kasus uji edit profil
Prosedur	1. Sudah masuk ke halaman profil pengguna 2. Mengisi nama = cobaa, bio = traveler, location = malang 3. Menekan tombol update
Hasil yang diharapkan	Data profil tersimpan dan menampilkan profil pengguna
Hasil	Data profil tersimpan dan menampilkan profil pengguna
Status	Valid

6.2.25 Pengujian Validasi Hapus Post

- a. Kasus uji hapus post

Tabel 6.39 Kasus Uji Hapus Post

Nama Kasus Uji	Kasus uji hapus post
Prosedur	1. Sudah masuk ke halaman profil pengguna 2. Menekan tombol hapus pada post
Hasil yang diharapkan	Post dihapus dan menampilkan daftar post pengguna
Hasil	Post dihapus dan menampilkan daftar post pengguna

Status	Valid
---------------	-------

6.2.26 Pengujian Validasi Laporkan Post

- a. Kasus uji laporkan post

Tabel 6.40 Kasus Uji Laporkan Post

Nama Kasus Uji	Kasus uji laporkan post
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman post terbaru 2. Menekan tombol lapor pada post 3. Memilih alasan melaporkan post
Hasil yang diharapkan	Sistem akan menampilkan pemberitahuan laporan sukses terkirim
Hasil	Menampilkan pemberitahuan laporan sukses terkirim
Status	Valid

6.2.27 Pengujian Validasi Lihat Notifikasi

- a. Kasus uji lihat notifikasi

Tabel 6.41 Kasus Uji Lihat Notifikasi

Nama Kasus Uji	Kasus uji lihat notifikasi
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan tombol notifikasi
Hasil yang diharapkan	Sistem menampilkan daftar notifikasi
Hasil	Sistem menampilkan daftar notifikasi
Status	Valid

6.2.28 Pengujian Validasi Logout

- a. Kasus uji logout pengguna

Tabel 6.42 Kasus Uji Logout Pengguna

Nama Kasus Uji	Kasus uji logout pengguna
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman home 2. Menekan icon foto pengguna 3. Menekan tombol sign out

Hasil yang diharapkan	Sistem menampilkan halaman login
Hasil	Sistem menampilkan halaman login
Status	Valid

b. Kasus uji logout admin

Tabel 6.43 Kasus Uji Logout Admin

Nama Kasus Uji	Kasus uji logout admin
Prosedur	1. Sudah masuk ke halaman daftar laporan 2. Menekan tombol Logout
Hasil yang diharapkan	Sistem menampilkan halaman login
Hasil	Sistem menampilkan halaman login
Status	Valid

6.2.29 Pengujian Validasi Login Admin

a. Kasus uji berhasil *login* admin

Tabel 6.44 Kasus Uji Berhasil *Login* admin

Nama Kasus Uji	Kasus uji berhasil <i>login</i> admin
Prosedur	1. Sudah masuk ke halaman login aplikasi admin 2. mengisi kolom username = admin, password = admin 3. menekan tombol <i>Login</i>
Hasil yang diharapkan	Sistem akan menampilkan halaman laporan
Hasil	Sistem menampilkan halaman laporan
Status	Valid

b. Kasus uji gagal *login* admin 1

Tabel 6.45 Kasus Uji Gagal *Login* Admin 1

Nama Kasus Uji	Kasus uji gagal <i>login</i> admin
Prosedur	1. Sudah masuk ke halaman login aplikasi admin 2. mengisi kolom username = admin, password = 123 3. menekan tombol <i>Login</i>
Hasil yang diharapkan	Sistem menampilkan pemberitahuan "Username dan password salah"



Hasil	Sistem menampilkan pemberitahuan "Username dan password salah"
Status	Valid

c. Kasus uji gagal *login* admin 2

Tabel 6.46 Kasus Uji Gagal *Login* Admin 2

Nama Kasus Uji	Kasus uji gagal <i>login</i> admin 2
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman login aplikasi admin 2. Mengisi kolom username = admin, password tidak diisi 3. Menekan tombol <i>Login</i>
Hasil yang diharapkan	Menampilkan pemberitahuan " <i>The field is required</i> "
Hasil	Menampilkan pemberitahuan " <i>The field is required</i> "
Status	Valid

6.2.30 Pengujian Validasi Lihat Daftar Laporan Post

a. Kasus uji lihat daftar laporan post

Tabel 6.47 Kasus Uji Lihat Daftar Laporan Post

Nama Kasus Uji	Kasus uji lihat daftar laporan post
Prosedur	1. Sudah berhasil melakukan login
Hasil yang diharapkan	Sistem menampilkan daftar laporan
Hasil	Sistem menampilkan daftar laporan
Status	Valid

6.2.31 Pengujian Validasi Hapus Post Dari Daftar Laporan

a. Kasus uji hapus post dari daftar laporan

Tabel 6.48 Kasus Uji Hapus Post dari Daftar Laporan

Nama Kasus Uji	Kasus uji hapus post dari daftar laporan
Prosedur	<ol style="list-style-type: none"> 1. Sudah masuk ke halaman daftar laporan 2. Menekan tombol hapus laporan pada post
Hasil yang diharapkan	Laporan di hapus dari daftar laporan
Hasil	Laporan di hapus dari daftar laporan

Status	Valid
--------	-------

6.2.32 Pengujian Validasi Hapus Post

a. Kasus uji Hapus Post

Tabel 6.49 Kasus Uji Hapus Post

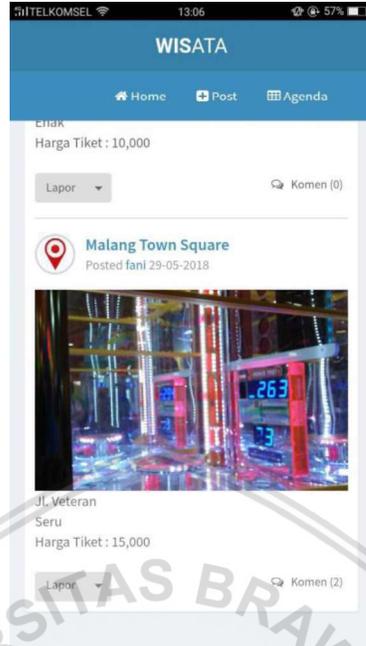
Nama Kasus Uji	Kasus uji hapus post
Prosedur	1. Sudah masuk ke halaman daftar laporan 2. Menekan tombol hapus total pada post
Hasil yang diharapkan	Laporan di hapus dan menampilkan halaman daftar laporan
Hasil	Laporan di hapus dan menampilkan halaman daftar laporan
Status	Valid

6.3 Pengujian *Compatibility*

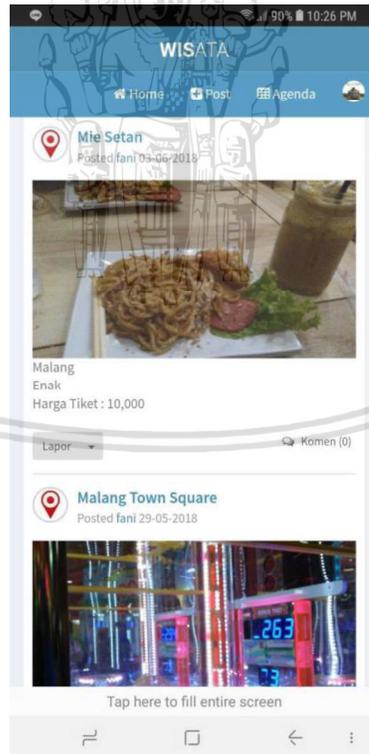
Pengujian *compatibility* merupakan tahap yang dilakukan untuk mengetahui kompatibilitas dari sistem yang dibangun untuk memastikan bahwa kebutuhan non fungsional sistem telah terpenuhi. Pengujian *compatibility* dilakukan dengan menjalankan sistem pada beberapa versi sistem operasi Android diantaranya Android 4.2 (*Jelly Bean*), Android 7.1 (*Nougat*), Android 8.0 (*Oreo*)



Gambar 6.1 Tampilan Aplikasi pada Android Versi 4.2 (*Jelly Bean*)



Gambar 6.2 Tampilan Aplikasi pada Android Versi 7.1 (Nougat)



Gambar 6.3 Tampilan Aplikasi pada Android Versi 8.0 (Oreo)

Tabel 6.50 Karakteristik Pengujian *Compatibility*

Karakteristik	Android 4.2 (<i>Jelly Bean</i>)	Android 7.1 (<i>Nougat</i>)	Android 8.0 (<i>Oreo</i>)
Font, menu, dan gambar serupa, dapat terbaca dan terlihat dengan jelas	Valid	Valid	Valid
Tata letak halaman tidak berubah	Valid	Valid	Valid
Dapat menjalankan fungsional yang ada	Valid	Valid	Valid

6.4 Analisis Hasil Pengujian

Analisis dari hasil pengujian didapatkan berdasarkan hasil pengujian yang telah dilakukan. Analisis hasil pengujian ini dilakukan guna mengetahui apakah sistem yang dibangun sudah berjalan dengan baik dan benar tanpa terjadi gangguan.

6.4.1 Analisis Hasil Pengujian Unit

Pengujian unit dalam aplikasi *social geotagging* destinasi wisata dilakukan dengan menerapkan metode basis path. Pada tahap ini, diuji 3 operasi sampel yaitu operasi kirimagenda(), operasi kirimpos(), dan operasi blokir(). Masing-masing operasi tersebut memiliki nilai cyclomatic complexity sebanyak 2, 3, dan 2 yang memiliki arti bahwa kompleksitas dari tiga unit yang telah diuji tersebut adalah rendah. Sehingga dapat disimpulkan bahwa sistem yang dibangun mudah untuk dilakukan maintenance.

6.4.2 Analisis Hasil Pengujian Validasi

Pengujian validasi dilakukan dengan menjalankan semua fungsi yang ada pada sistem. Hasil dari pengujian ini menyatakan bahwa dari 33 fungsionalitas yang ada pada sistem diperoleh hasil yang valid. Sehingga dapat disimpulkan bahwa sistem yang dibangun telah berjalan sesuai dengan fungsinya baik dalam kondisi normal maupun kondisi alternatifnya.

6.4.3 Analisis Hasil Pengujian *Compatibility*

Pengujian *compatibility* dilakukan dengan menjalankan sistem pada versi sistem operasi Android yang berbeda. Hasil dari pengujian *compatibility* menyatakan bahwa sistem dapat dijalankan pada ketiga versi Android yang berbeda yaitu Android versi 4.2 (*Jelly Bean*), Android 7.1 (*Nougat*), dan Android 8.0 (*Oreo*), fungsional yang ada dapat berjalan dengan baik pada ketiga versi Android yang digunakan, serta tata letak tampilan pada ketiga versi Android serupa dan terlihat dengan jelas.



BAB 7 PENUTUP

11.1 Kesimpulan

Berdasarkan hasil dari tahap analisis kebutuhan, tahap perancangan, tahap implementasi, serta tahap pengujian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan hasil analisis kebutuhan, Aplikasi *social geotagging* destinasi wisata menghasilkan 33 kebutuhan fungsional. Dimana kebutuhan fungsional yang telah didapat tersebut sudah mencakup permasalahan yang ada seperti kebutuhan fungsional dalam mencari wisata terdekat dari lokasi pengguna, membuat agenda wisata pengguna, dan menampilkan referensi wisata berdasarkan harga, dan beberapa kebutuhan fungsional lainnya.
2. Berdasarkan hasil perancangan sistem menghasilkan rancangan berupa kelas *Model*, *View*, dan *Controller*. Selain itu proses perancangan juga menghasilkan sebuah *Entity Relationship Diagram* (ERD), rancangan komponen, serta gambaran dari antarmuka sistem yang telah digunakan sebagai referensi atau acuan dalam pembangunan sistem.
3. Pengujian fungsional dan non fungsional yang sudah dilakukan pada aplikasi *social geotagging* destinasi wisata adalah *white box testing* dengan menerapkan metode *basis path*, *black box testing* yang terdiri dari pengujian validasi dan *compatibility testing*. Dari hasil pengujian unit dan pengujian validasi diperoleh hasil bahwa kompleksitas dari program adalah rendah sehingga program dapat di *maintenance* dengan mudah dan 100% valid atau tidak terjadi kesalahan fungsional sistem. Sedangkan berdasarkan pengujian *compatibility* diperoleh hasil bahwa sistem dapat dijalankan pada 3 versi sistem operasi Android yang berbeda yaitu Android 4.2 (*Jelly Bean*), Android 7.1 (*Nougat*), Android 8.0 (*Oreo*) tanpa adanya hambatan.

11.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi *social geotagging* destinasi wisata ini selanjutnya antara lain:

1. Aplikasi *social geotagging* destinasi wisata dapat dikembangkan dengan menambahkan fitur untuk memesan tiket wisata sehingga mempermudah pengguna untuk mendapatkan tiket wisata.
2. Dapat dilakukan pengembangan lebih lanjut untuk beberapa platform *mobile* lainnya, seperti *Windows Phone* dan *iOS*, sehingga selain pengguna perangkat Android dapat menggunakan aplikasi ini pada perangkatnya.

DAFTAR PUSTAKA

- Antopani, T., Studi, P., & Kelola, T. 2015. *Fotografi , Pariwisata , Dan Media Aktualisasi Diri*, 11(1), 31–40.
- Baskara, K.A., Sukarno, B., & Riyanto, B. 2017. Motif Penggunaan Instagram Sebagai Referensi Wisata (Studi Khusus pada Mahasiswa Fisip Unisri Surakarta). *Transformasi*. Vol I, No. 32.
- Bindal, Nancy and Mehta, Aanchal., 2015. *Survey On Software Development Processing Models. International Journal of Exploring Emerging Trends in Engineering (IJEETE)*.
- Blanco, J., Upton D., 2009. *Codeigniter 1.7*, s.l.: Packt Publishing.
- Caytiles, Ronnie D. dan Lee, Sunguk., 2014. *A Review of an MVC Framework based Software Development*.
- Efendy, A. A., 2016. Rancang Bangun Aplikasi Sosial *Geotagging* Keindahan Alam Indonesia pada Sistem Operasi Android.
- EMS, Tim. 2013. *Pemrograman Mobile dengan PhoneGap*. Jakarta: PT. Elex Media Komputindo.
- Eryani, A. (n.d.). Faktor-Faktor Pengambilan Keputusan Wisatawan Untuk Berkunjung Ke Objek Wisata Istana Siak Sri Indrapura Kabupaten Siak Provinsi Riau, 1–15.
- Essayad, A., 2011. Design and implementation of a platform for location-based services: a case study of GIS of archaeological and handicraft of Fez Medina. *IJCSI International Journal of Computer Science Issues*.
- Guru99, 2014. Learn McCabe’s Cyclomatic Complexity with Example. [Online] Tersedia di: <https://www.guru99.com/cyclomatic-complexity.html> [Diakses 12 Juli 2018].
- KOMINFO, P. 2013. Kominfo : *Pengguna Internet di Indonesia 63 Juta Orang*. [Online] Website Resmi Kementerian Komunikasi dan Informatika RI. Tersedia di: https://kominfo.go.id/index.php/content/detail/3415/Kominfo+%3A+Pengguna+Internet+di+Indonesia+63+Juta+Orang/0/berita_satker [Diakses 2 Januari 2018]
- Kumar, Naresh et. al., 2013. *Evolving a New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction. International Journal of Soft Computing and Engineering (IJSCE)*, Vol. 3
- Ladjamudin, A. B. B. 2006. *Rekayasa Perangkat Lunak*. Yogyakarta: Graha Ilmu
- Made, N., Rukmiyati, S., & Suastini, N. M. 2016. Dampak media sosial terhadap perilaku wisatawan, 339–347.



- Mubiin, M. U., 2016. Rancang Bangun Aplikasi Mobile Social Geotagging Transportasi Publik pada Platform Android.
- Mulyadi, Adi. 2010. *Membangun Aplikasi Android*. Yogyakarta: Multimedia Center Publishing.
- Mulyatiningsih, Endang., dkk. 2014. "Perintisan Bank Resep Elektronik Menggunakan Data Base Mysql (Structured Query Language)". Fakultas Teknik Universitas Negeri Yogyakarta.
- Nandipati, A., 2011. Assesment of metadata associated, s.l.: Institute for Geoinformatics University of Muenster.
- Parikesit, A. G. 2016. Fakta Menarik Pengguna Instagram di Indonesia. [Online] Tersedia di : <https://www.cnnindonesia.com/teknologi/20160118150454-188-105071/fakta-menarik-pengguna-instagram-di-indonesia> [Diakses 17 Maret 2018]
- Pressman, R. S., Maxim B. R., 2015. *Software Engineering: A Practitioner's Approach, Eighth Edition*. New York: McGraw-Hill.
- Republik Indonesia. 1999. Undang-Undang No. 10 Tahun 2009 tentang Kepariwisata. Lembaran Negara RI Tahun 2009, No. 11. Sekretariat Negara. Jakarta.
- Rillaningrum, Y. 2014. Rancang Bangun Aplikasi Android untuk Pencarian Lokasi ATM Terdekat di Kota Malang.
- Rumbaugh, J., Jacobson, I. & Booch, G., 2005. *The Unified Modeling Language reference manual*. 2nd ed. Boston: Addison-Wesley.
- S, Rosa A., Shalahuddin, M. 2015. *Rekayasa Perangkat Lunak*. Bandung: Informatika
- Sholeh, M., Widyastuti, N. & Mashuri, A., 2011. Aplikasi Google Maps API untuk Sistem Informasi Geografis. Aplikasi Google Maps API, pp. 97-103.
- Simarmata, J. 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Statista. 2016. *Number of social media users worldwide 2010-2020 | Statista*. [Online] Tersedia di: <https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/> [Diakses 2 Januari 2018].
- Sulaiman, F. 2016. Perancangan Aplikasi Media Sosial Traveler Berbasis Gamification Menggunakan Google Map API V3.
- Trihatmaja, R. 2015. Rancang Bangun Sistem Komunitas Berbagi Informasi Resep dan Lokasi Kuliner Berbasis Mobile.
- Voldemord, L., 2015. www.codepolitan.com. [Online] Tersedia di: <https://www.codepolitan.com/codeigniter-3-0-akhirnya-dirilis> [Diakses 13 Juli 2018].