

RANCANG BANGUN SISTEM INFORMASI PARIWISATA PULAU LOMBOK BERBASIS PERANGKAT BERGERAK

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Insan Bagus Perkasa
NIM: 105060806111003



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

RANCANG BANGUN SISTEM INFORMASI PARIWISATA PULAU LOMBOK BERBASIS
PERANGKAT BERGERAK

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Insan Bagus Perkasa
NIM: 105060806111003

Skripsi ini telah diuji dan dinyatakan lulus pada
18 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Diah Priharsari, S.T., M.T.
NIK: 201304 810422 2 000

Agi Putra Kharisma, S.T., M.T.
NIK: 201304 860430 1 001

Mengetahui
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T.
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

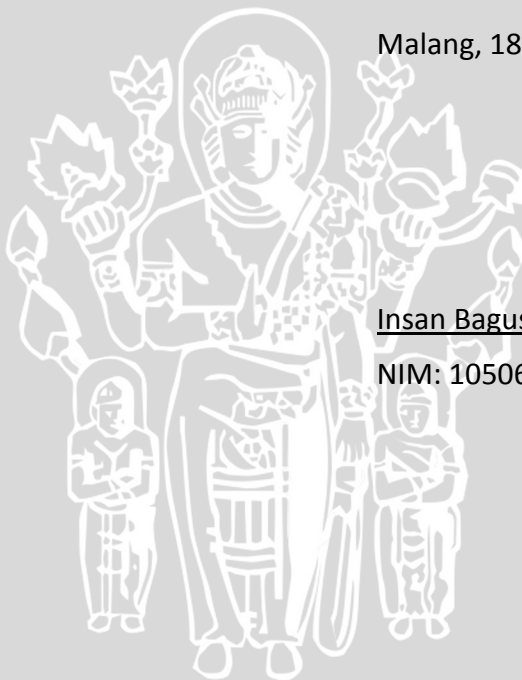
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan 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, 18 Januari 2016

Insan Bagus Perkasa

NIM: 105060806111003



KATA PENGANTAR

Puji dan syukur Penulis panjatkan kehadirat Allah SWT, karena hanya dengan rahmat dan karunia-Nya Penulis dapat menyelesaikan skripsi dengan judul **“RANCANG BANGUN SISTEM INFORMASI PARIWISATA PULAU LOMBOK BERBASIS PERANGKAT BERGERAK”**. Melalui kesempatan ini, Penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama pengerjaan skripsi, diantaranya:

1. Kedua orang tua Suselo dan Syara'lyah yang telah memberikan motivasi, kasih sayang serta dukungan moril dan materil. Adik Billy Sukma Dwi Prasetyo dan Cyntia Aura Tri Pangestu yang telah memberikan semangat dari awal sampai akhir pengerjaan skripsi ini.
2. Ibu Diah Priharsari, S.T., M.T. selaku dosen pembimbing I yang telah banyak memberikan ilmu dan saran selama penyusunan skripsi ini.
3. Bapak Agi Putra Kharisma, S.T, M.T. selaku dosen pembimbing II yang juga telah banyak memberikan ilmu dan saran selama penyusunan skripsi ini.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Semua teman-teman PTIIK, khususnya Informatika/Illmu Komupter angkatan 2010 terima kasih atas segala bantuan dan dukungannya selama ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Dengan segala kerendahan hati, penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Akhirnya, semoga skripsi ini dapat bermanfaat dan berguna bagi pembaca terutama mahasiswa PTIIK Universitas Brawijaya.

Malang, 18 Januari 2016

Penulis

insanbagus@gmail.com

ABSTRAK

Pulau Lombok merupakan salah satu destinasi pariwisata yang diminati di Indonesia saat ini. Seiring dengan banyaknya wisatawan yang datang ke Pulau Lombok, mengakibatkan berkembangnya tempat wisata dan fasilitas pendukung lainnya. Saat ini terdapat banyak lokasi wisata baru yang ada di pulau Lombok. Namun dalam proses pencarian informasi wisata maupun kuliner kurang efisien karena pencarian informasi tidak terpusat pada satu sumber saja. Sehingga dalam proses pencariannya memerlukan waktu yang cukup lama, dan hasil yang diberikan tidak langsung sesuai dengan keinginan. Hal ini dikarenakan kurangnya sarana informasi (terutama pada aplikasi perangkat bergerak) yang memberikan informasi terbaru mengenai pariwisata di Pulau Lombok. Untuk itu perlu adanya suatu sistem informasi berbasis perangkat bergerak sebagai wadah wisatawan dalam mencari informasi. Aplikasi perangkat bergerak yang dibutuhkan untuk menunjang penyampaian informasi wisata di Pulau Lombok adalah dengan penggunaan layanan dari *Google Map API* pada sistem informasi. Penerapan layanan *Google Map API* tersebut memungkinkan sistem informasi yang dibuat dapat diakses secara luas. Adanya kemudahan dalam mengakses semakin memudahkan pengguna khususnya para wisatawan dalam mencari informasi.

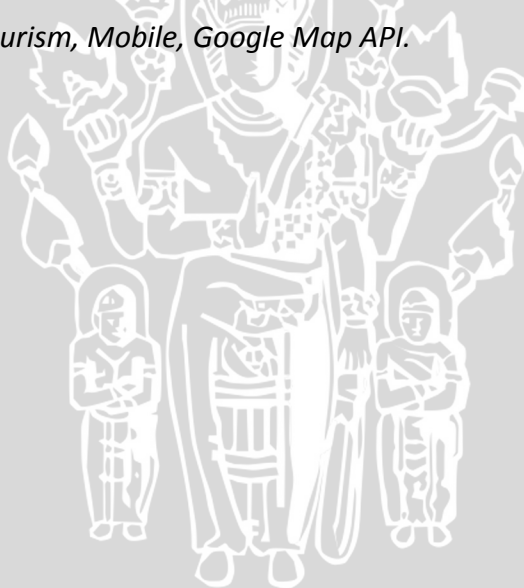
Kata kunci: Pulau Lombok, Pariwisata, Perangkat Bergerak, Google Map API.



ABSTRACT

Lombok Island is one of the attractive tourism destinations in Indonesia at this time. Because of the many tourists coming to Lombok island, resulting in the development of tourist attractions and other supporting facilities. Today there are many new tourist sites on Lombok island. But in the process of finding tour and culinary information less efficiently, because the search information is not centralized on a single source. So the search process requires considerable time, and the results given indirectly accordance with the wishes. This is because the lack of means information (especially mobile application), which provides the latest information on tourism in Lombok Island. For that we need an information system based on mobile devices as a forum for tourists in search of information. Mobile device applications are required to support the delivery of travel information on the island of Lombok is to use the services of Google Map API in the information system. Google Maps API service implementation that allows system information is made widely accessible. The ease in accessing increasingly easier for users, especially tourists in search of information.

Keywords: *Lombok, Tourism, Mobile, Google Map API.*

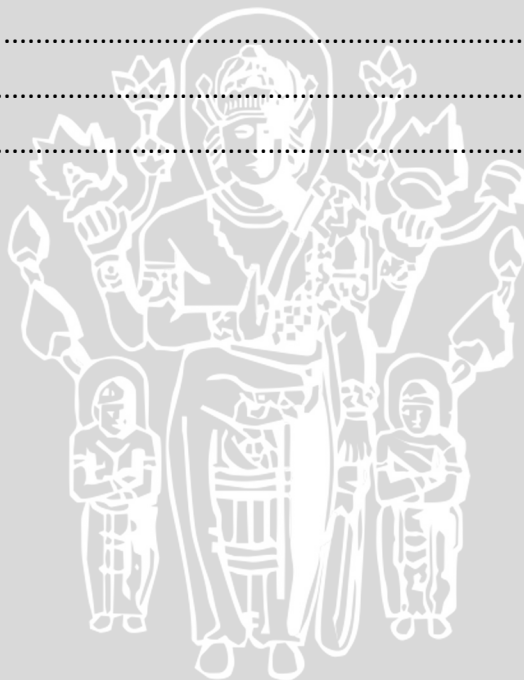


DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI.....	4
2.1 Kajian Pustaka	4
2.1.1 Aplikasi Wisata	4
2.1.2 Penggunaan Google Map API	4
2.1.3 Penggunaan Platform Android Pada Perangkat Bergerak.....	5
2.2 Dasar Teori.....	5
2.2.1 Pariwisata Pulau Lombok.....	5
2.2.2 Android.....	5
2.2.3 Aplikasi Hybrid	6
2.2.4 JQuery Mobile	7
2.2.5 PhoneGap.....	8
2.2.6 Google Map API	8
2.2.7 Pengujian Usability.....	9
BAB 3 METODOLOGI PENELITIAN	10
3.1 Studi Literatur	10

3.2 Analisis Kebutuhan	11
3.3 Perancangan Perangkat Lunak	12
3.3.1 Perancangan Arsitektur Sistem.....	12
3.3.2 Perancangan Arsitektur Diagram	12
3.3.3 Perancangan Basis Data	12
3.3.4 Perancangan Antarmuka.....	12
3.3.5 Perancangan Algoritma.....	12
3.4 Implementasi	12
3.5 Pengujian dan Analisis	13
3.6 Pengambilan Kesimpulan dan Saran	14
BAB 4 ANALISIS DAN PERANCANGAN	15
4.1 Analisis Kebutuhan	15
4.1.1 Analisis Fitur Pilihan	16
4.1.2 Gambaran Umum Aplikasi	16
4.1.3 Identifikasi Aktor	17
4.1.4 Analisis Kebutuhan Fungsional	17
4.1.5 Analisis Kebutuhan Non-Fungsional	21
4.2 Perancangan Perangkat Lunak	22
4.2.1 Perancangan Arsitektur Sistem.....	22
4.2.2 Perancangan Arsitektur Diagram	23
4.2.3 Perancangan Basis Data	24
4.2.4 Perancangan Antarmuka.....	25
4.2.5 Perancangan Algoritma.....	29
BAB 5 IMPLEMENTASI	32
5.1 Implementasi	32
5.1.1 Spesifikasi Lingkungan Pengembangan Sistem.....	33
5.1.2 Batasan-Batasan Implementasi	34
5.1.3 Implementasi Basis Data	34
5.1.4 Implementasi <i>Class</i> dan <i>Assets</i> pada <i>File</i> Program.....	36
5.1.5 Implementasi Kode Program	38
5.1.6 Implementasi Antarmuka	41
5.1.7 Implementasi Geolocation.....	44

BAB 6 PENGUJIAN DAN ANALISIS	50
6.1 Pengujian Validasi	50
6.1.1 Kasus Uji	50
6.1.2 Hasil Pengujian Validasi	53
6.2 Pengujian <i>Usability</i>	55
6.2.1 Perhitungan Kuisioner SUS	56
6.2.2 Hasil Kuisioner SUS	58
6.3 Analisis Hasil Pengujian	58
6.3.1 Analisis Pengujian Hasil Validasi	58
6.3.2 Analisis Pengujian Hasil <i>Usability</i>	58
BAB 7 KESIMPULAN DAN SARAN	59
7.1 Kesimpulan	59
7.2 Saran	59
DAFTAR PUSTAKA	60



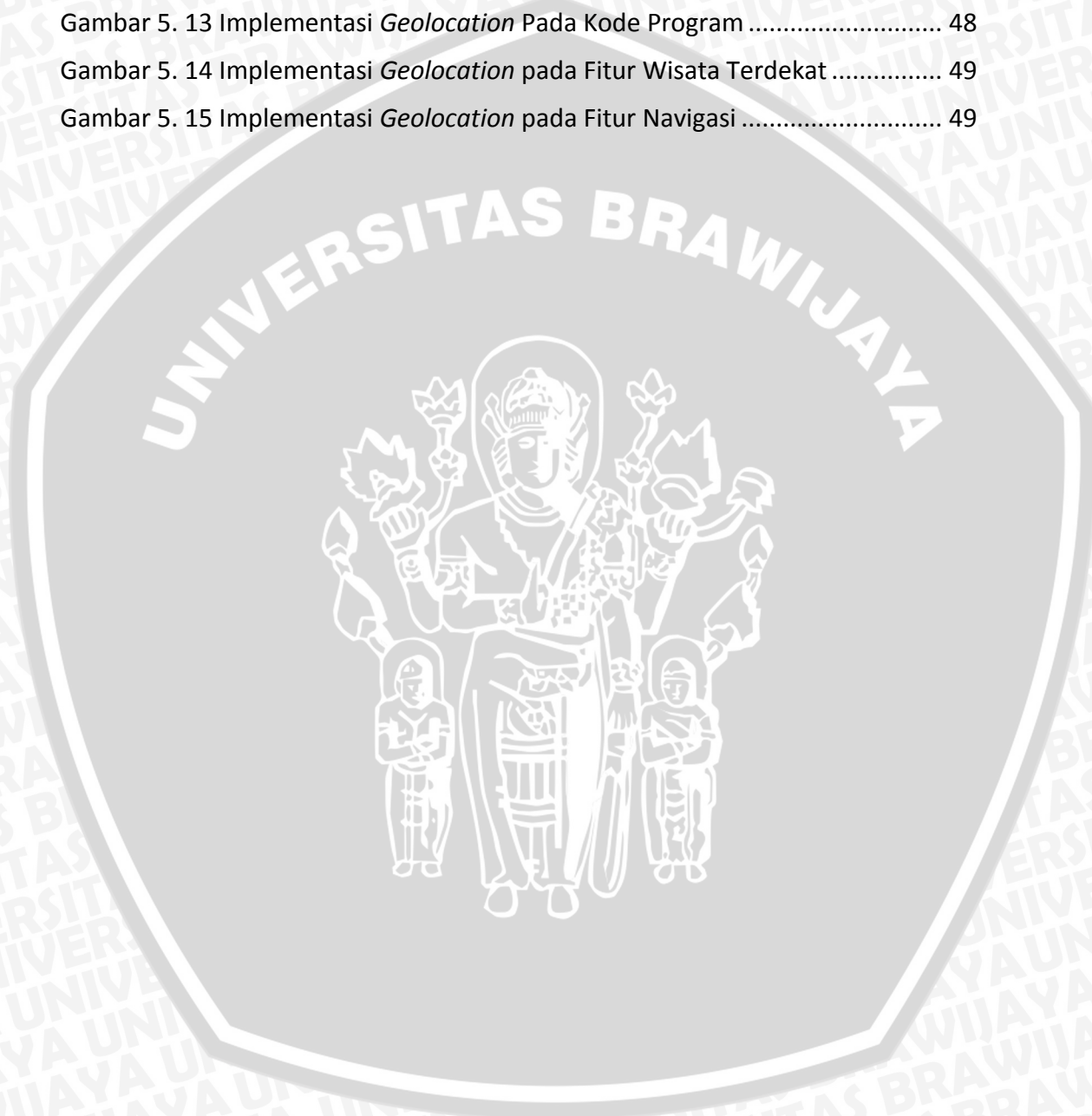
DAFTAR TABEL

Tabel 4. 1 Judul Penelitian Beserta Fitur yang Digunakan	16
Tabel 4. 2 Identifikasi Aktor	17
Tabel 4. 3 Spesifikasi Kebutuhan Fungsional Pengguna	18
Tabel 4. 4 Spesifikasi Kebutuhan Non-Fungsional	21
Tabel 4. 5 Struktur Tabel Wisata	25
Tabel 4. 6 Struktur Tabel Kuliner	25
Tabel 5. 1 Spesifikasi Perangkat Keras Komputer	33
Tabel 5. 2 Spesifikasi Perangkat Keras <i>Smartphone</i>	33
Tabel 5. 3 Spesifikasi Perangkat Lunak Komputer	33
Tabel 5. 4 Spesifikasi Perangkat Lunak <i>Smartphone</i>	34
Tabel 5. 5 Spesifikasi <i>Tools</i> Pengembangan	34
Tabel 5. 6 Implementasi <i>Class</i> pada Kode Program *.java	36
Tabel 5. 7 Implementasi <i>Assets</i> pada Kode Program *.html	36
Tabel 5. 8 Implementasi <i>Assets</i> pada Kode Program *.js	37
Tabel 5. 9 Implementasi <i>Assets</i> pada Kode Program *.php	37
Tabel 6. 1 Kasus Uji Melihat Halaman Beranda	50
Tabel 6. 2 Kasus Uji Mencari Tempat Wisata	51
Tabel 6. 3 Kasus Uji Mencari Tempat Kuliner	51
Tabel 6. 4 Kasus Uji Melihat Rute Lokasi Tujuan	52
Tabel 6. 5 Kasus Uji Melihat Peta Lokasi	52
Tabel 6. 6 Kasus Uji Melihat Daftar Wisata Terdekat	53
Tabel 6. 7 Hasil Pengujian Validasi Aplikasi Pariwisata Pulau Lombok	54
Tabel 6. 8 Daftar Pernyataan Kuesioner	55
Tabel 6. 9 Penjelasan Skala <i>Likert 5 Level of Agreement</i>	56
Tabel 6. 10 Perhitungan Kuesioner SUS	57

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi Aplikasi Hybrid	7
Gambar 2. 2 Pemangilan File <i>Javascript</i> Dan <i>CSS JQuery Mobile</i> Melalui File <i>HTML</i>	8
Gambar 2. 3 Fitur yang Didukung PhoneGap pada Beberapa Sistem Operasi	8
Gambar 3. 1 Diagram Alir Penelitian	10
Gambar 4. 1 Diagram Pohon Analisis dan Perancangan	15
Gambar 4. 2 <i>Diagram Context</i> Sistem	19
Gambar 4. 3 <i>Data Flow Diagram Level 1</i>	20
Gambar 4. 4 <i>Data Flow Diagram Level 2</i> Memilih Tempat Wisata	20
Gambar 4. 5 <i>Data Flow Diagram Level 2</i> Memilih Tempat Kuliner	21
Gambar 4. 6 Perancangan Arsitektur Sistem	23
Gambar 4. 7 Perancangan Arsitektur Sistem	23
Gambar 4. 8 Perancangan Basis Data	24
Gambar 4. 9 <i>Sitemap</i> Aplikasi	26
Gambar 4. 10 Antarmuka Halaman <i>Beranda</i>	26
Gambar 4. 11 Antarmuka Halaman Kategori Wisata	27
Gambar 4. 12 Antarmuka Halaman Kategori Kuliner	27
Gambar 4. 13 Antarmuka Halaman <i>Detail</i> Wisata dan Kuliner	28
Gambar 4. 14 Antarmuka Halaman Tentang	28
Gambar 4. 15 Algoritma Mencari Tempat Wisata	29
Gambar 4. 16 Algoritma Mencari Tempat Kuliner	30
Gambar 4. 17 Algoritma Melihat Daftar Wisata terdekat	31
Gambar 5. 1 Diagram Pohon Implementasi	32
Gambar 5. 2 <i>Physical Diagram</i>	35
Gambar 5. 3 <i>Query Create Table</i> Wisata	35
Gambar 5. 4 <i>Query Create Table</i> Kuliner	36
Gambar 5. 5 Kode Program Mencari Tempat Wisata	39
Gambar 5. 6 Kode Program Menampilkan Halaman <i>Detail</i>	40
Gambar 5. 7 Kode Program Melihat Daftar Wisata Terdekat	41
Gambar 5. 8 Antarmuka Halaman <i>Beranda</i>	41

Gambar 5. 9 Antarmuka Halaman Kategori Wisata	42
Gambar 5. 10 Antarmuka Halaman Daftar Wisata	42
Gambar 5. 11 Antarmuka Halaman Detail	43
Gambar 5. 12 Antarmuka Halaman Tentang	44
Gambar 5. 13 Implementasi <i>Geolocation</i> Pada Kode Program	48
Gambar 5. 14 Implementasi <i>Geolocation</i> pada Fitur Wisata Terdekat	49
Gambar 5. 15 Implementasi <i>Geolocation</i> pada Fitur Navigasi	49



BAB 1 PENDAHULUAN

1.1 Latar belakang

Pulau Lombok merupakan salah satu destinasi wisata yang diminati di Indonesia saat ini. Menurut survei yang dilakukan Badan Pusat Statistik (BPS) menunjukkan sepanjang bulan Maret 2014 terjadi lonjakan jumlah wisatawan asing yang cukup signifikan di Lombok (Hartono, T., 2014). Seiring dengan banyaknya wisatawan yang datang ke Pulau Lombok, mengakibatkan berkembangnya tempat wisata dan fasilitas pendukung lainnya. Saat ini terdapat banyak lokasi wisata baru yang ada di pulau Lombok. Tapi tidak semua tempat-tempat wisata itu diketahui oleh para wisatawan. Dengan adanya akses internet, informasi mengenai pariwisata yang ada di pulau Lombok dapat dengan mudah diinformasikan kepada wisatawan.

Namun dalam proses pencarian informasi wisata maupun kuliner kurang efisien karena pencarian informasi tidak terpusat pada satu sumber saja. Sehingga dalam proses pencariannya memerlukan waktu yang cukup lama, dan hasil yang diberikan tidak langsung sesuai dengan keinginan. Hal ini dikarenakan kurangnya sarana informasi (terutama pada aplikasi perangkat bergerak) yang memberikan informasi terbaru mengenai pariwisata di Pulau Lombok. Untuk itu perlu adanya suatu sistem informasi berbasis perangkat bergerak *Android* sebagai media informasi pariwisata pulau Lombok karena populasi *Android* di dunia telah lebih dari satu miliar pengguna, dimana *Android* sebagai pilihan utama *smartphone* di Indonesia (Heriyanto, T., 2014).

Aplikasi perangkat bergerak yang dibutuhkan untuk menunjang penyampaian informasi wisata di Pulau Lombok adalah dengan penggunaan layanan dari *Google Map API* pada sistem informasi. Penerapan layanan *Google Map API* tersebut memungkinkan sistem informasi yang dibuat dapat diakses secara luas. Adanya kemudahan dalam mengakses semakin memudahkan pengguna khususnya para wisatawan dalam mencari informasi. Informasi yang dimaksud yaitu peta lokasi wisata, jalur menuju lokasi, dan informasi lain berupa data text seperti deskripsi singkat dan gambar-gambar terkait. Penggunaan layanan *Google Map API* nantinya dapat membuat aplikasi yang dibangun menjadi lebih atraktif.

Oleh karena itu, penulis mengangkat penelitian berjudul **“Rancang Bangun Sistem Informasi Pariwisata Pulau Lombok Berbasis Perangkat Bergerak”**. Aplikasi ini diharapkan dapat memberikan informasi dan mempermudah pencarian wisatawan yang ingin menikmati pesona Pulau Lombok dengan sejuta keindahan alam dan ragam kuliner. Sehingga popularitas wisata Pulau Lombok dapat terdongkrak dan menjadi salah satu ikon pariwisata di Indonesia sebagai destinasi wisata yang harus dikunjungi baik oleh wisatawan lokal maupun mancanegara.

1.2 Rumusan masalah

Berdasarkan latar belakang dan identifikasi masalah diatas, maka penulis merumuskan hal-hal yang harus dianalisis diantaranya:

1. Bagaimana merancang dan mengimplementasikan sistem informasi pariwisata Pulau Lombok berbasis perangkat bergerak yang dapat menjadi wadah bagi wisatawan dalam mencari informasi dan lokasi wisata?
2. Bagaimana hasil pengujian *usability* dari sistem informasi pariwisata Pulau Lombok berbasis perangkat bergerak yang dikembangkan?

1.3 Tujuan

Tujuan dari dilaksanakannya penelitian ini yaitu untuk merancang dan mengimplementasikan aplikasi perangkat bergerak yang dapat menjadi sarana bagi para wisatawan dalam mencari informasi pariwisata, baik tempat wisata atau tempat kuliner yang ada di Pulau Lombok.

1.4 Manfaat

Berdasarkan yang dapat diperoleh dari penelitian ini antara lain:

- a. Bagi Penulis
 1. Mengaplikasikan ilmu dan pengetahuan yang didapat selama mengikuti perkuliahan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
 2. Mendapatkan pengetahuan tentang bagaimana cara menciptakan suatu sistem informasi digital pada perangkat bergerak yang dapat berfungsi sebagai wadah dalam mencari informasi wisata dan kuliner.
- b. Bagi Pengguna
 1. Memberikan wadah bagi wisatawan dalam mencari informasi pariwisata di Pulau Lombok.
 2. Memudahkan wisatawan dalam pencarian lokasi tempat wisata atau tempat kuliner yang ingin dituju.

1.5 Batasan masalah

Untuk menghindari kemungkinan semakin berkembangnya masalah, maka batasan masalah dalam skripsi ini dibatasi dalam hal:

1. Informasi mengenai pariwisata hanya mencakup wisata dan kuliner di Pulau Lombok.
2. Aplikasi ini hanya menampilkan informasi wisata dan kuliner berupa nama tempat, lokasi, kategori, alamat, deskripsi, navigasi, peta dan wisata terdekat.

3. Sebagian informasi dan data dari aplikasi ini merupakan data sekunder, yaitu data yang diambil melalui pencarian secara *online*.
4. Aplikasi perangkat bergerak dibangun dengan menggunakan bahasa pemrograman HTML5, CSS3, dan Javascript.
5. Aplikasi dibangun pada platform *Android* dengan menggunakan pemrograman sistem *Hybrid* untuk memudahkan pengembangan selanjutnya.

1.6 Sistematika pembahasan

Penyusunan skripsi ini menggunakan kerangka pembahasan yang tersusun sebagai berikut:

BAB I PENDAHULUAN

Bab ini merupakan dasar dari penyusunan skripsi ini yang terdiri dari latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi tentang kajian pustaka, referensi atau sumber-sumber yang berhubungan dengan permasalahan dalam skripsi.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan bagaimana metodologi untuk perancangan dan pembuatan Sistem Informasi Pariwisata Pulau Lombok Berbasis Perangkat Bergerak.

BAB IV ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisa kebutuhan dari Aplikasi Sistem Informasi Pariwisata Pulau Lombok Berbasis Perangkat Bergerak kemudian merancang hal-hal yang berhubungan dengan analisa tersebut.

BAB V IMPLEMENTASI

Bab ini menjelaskan tentang hasil perancangan dari analisis kebutuhan dan implementasi Aplikasi Sistem Informasi Pariwisata Pulau Lombok Berbasis Perangkat Bergerak *Platform Android* yang telah direalisasikan.

BAB VI PENUTUP

Bab ini menjelaskan kesimpulan yang dapat diambil dari pengujian dan analisis pembuatan Sistem Informasi Pariwisata Pulau Lombok Berbasis Perangkat Bergerak dan disertai saran yang mendukung untuk pengembangan selanjutnya.

BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini membahas penelitian-penelitian terkait yang pernah dilakukan sebelumnya, serta dasar teori yang diperlukan dalam pembuatan perangkat lunak.

2.1 Kajian Pustaka

Pembahasan terhadap kajian pustaka dilakukan dengan menganalisis penelitian yang pernah dilakukan sebelumnya dan memiliki kemiripan topik pembahasan dengan penelitian yang sedang dilakukan.

2.1.1 Aplikasi Wisata

Khawlah A. Al- Rayes dkk dalam penelitiannya yang berjudul *"A Mobile Tourist Guide for Trip Planning"* menjelaskan aplikasi berbasis *Android* dapat memandu para wisatawan di perjalanan mereka. Pada aplikasi perangkat bergerak ini menginformasikan tentang tempat wisata dan menyarankan rencana perjalanan dengan mengambil preferensi wisata, kondisi cuaca dan waktu untuk menjadi bahan pertimbangan. Aplikasi ini dirancang untuk kota Istanbul (Al- Rayes, K. A. et al., 2011).

Aplikasi yang akan dibangun nantinya akan diimplementasikan pada perangkat bergerak. Selain itu aplikasi yang akan dibangun menginformasikan tentang tempat dan penggunaan fitur-fitur yang membuat wisatawan terbantu seperti dapat menampilkan daftar wisata terdekat. Selain agar dapat digunakan kapanpun dan dimanapun penggunaan perangkat bergerak juga memudahkan penggunaannya dalam mengakses aplikasi yang membutuhkan koneksi internet.

2.1.2 Penggunaan Google Map API

Djoni HS dkk dari Jurusan Teknik Informatika Universitas Kristen Petra Surabaya dalam penelitiannya yang berjudul *"Sistem Informasi Geografis Dengan Fitur Peta Dan Rute Perjalanan Studi Kasus Di Kabupaten Malang"* melakukan penelitian mengenai penggunaan layanan dari *Google Map API* pada Sistem Informasi Geografis (SIG) Wisata (Setiabudi, D. H., Rostianingsih, S. & Tj, L. J., 2014). Penerapan layanan *Google Map API* tersebut memungkinkan SIG yang dibuat dapat diakses secara luas. Adanya kemudahan dalam mengakses semakin memudahkan pengguna khususnya wisatawan dalam mencari informasi.

Selain kemudahan bagi pengguna aplikasi, teknologi *Google Map API* juga memberikan kemudahan bagi pengembangnya dalam membangun sistem informasi berbasis peta. Pengembang dapat langsung menambahkan layanan *Google Map API* pada aplikasi yang dibuat tanpa menghabiskan waktu untuk mengembangkan kode program dari fitur peta yang digunakan. Penggunaan layanan *Google Map API* nantinya dapat membuat aplikasi yang dibangun menjadi lebih atraktif. Selain itu pengguna aplikasi juga dimudahkan dengan adanya layanan peta untuk mencari dan menuju lokasi wisata atau kuliner yang diinginkan.

2.1.3 Penggunaan Platform Android Pada Perangkat Bergerak

Penggunaan *mobile platform Android* nantinya dapat memberikan beberapa kelebihan dalam membangun suatu aplikasi *mobile*. Seperti yang telah dijelaskan oleh Alicia S dan Xaverius N dalam penelitiannya yang berjudul "*Prototipe Aplikasi Sistem Informasi Akademik Pada Perangkat Android*". Penelitian ini bertujuan untuk merancang sistem informasi akademik berbasis android, yang dapat diakses melalui perangkat bergerak (*mobile*) untuk memudahkan akses yang lebih fleksibel (Sinsuw, A. & Najooan, X., 2013). Berbeda dengan *mobile platform* lainnya, *Android* tersedia sebagai perangkat lunak *open source*. Hal tersebut memungkinkan seluruh perangkat untuk diisi dengan platform *Android* dengan penyesuaian yang tanpa batas. Oleh karena itu *Android* menjadi *platform* yang menarik karena pengembangan suatu aplikasi dapat digunakan pada beberapa perangkat berbeda.

Dengan adanya dukungan yang baik dari berbagai produsen perangkat terhadap sistem operasi *Android*, nantinya aplikasi yang dibangun dapat memiliki kemampuan penyesuaian yang baik pada berbagai perangkat dengan sistem operasi *Android*. Hal tersebut memungkinkan aplikasi akan memiliki banyak pengguna karena aplikasi yang dibangun dapat digunakan pada sebagian besar perangkat yang tersedia.

2.2 Dasar Teori

Dasar teori memberikan pembahasan tentang teori-teori dasar yang digunakan dalam penelitian terkait penggunaan perangkat lunak, teknologi, dan metode.

2.2.1 Pariwisata Pulau Lombok

Pulau Lombok memang tempat terbaik yang selalu layak untuk dikunjungi. Pulau Lombok merupakan bagian dari Provinsi Nusa Tenggara Barat. Pulau utamanya dikelilingi oleh beberapa pulau kecil. Masyarakat setempat menyebut pulau-pulau itu dengan sebutan "Gili".

Untuk Anda yang mencintai binatang serta tumbuhan, jangan terkejut bila Anda menemukan Flora dan Fauna yang mirip dengan yang ada di Australia. Karena Pulau Lombok Indonesia ini merupakan garis batas antara Flora-Fauna Asia dan Australia. Di bagian sebelah Barat pulau inilah Anda bisa menemukan hewan dan tumbuhan yang mirip dengan yang ada di Australia. Sedangkan di tengah-tengah pulau terdapat pemisah, yaitu Gunung Rinjani (sekitar 3.726 Mdpl) yang memiliki danau di ketinggian. Danau tersebut bernama Segara Anak (LombokIndonesia.org, 2015).

2.2.2 Android

Android adalah sebuah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka

sendiri untuk digunakan oleh bermacam peranti bergerak (updatekeren.com, 2012).

Salah satu penyebab mengapa sistem operasi Android begitu gampang diterima pasar dan dengan begitu cepatnya menjadi "raja" adalah karena kelebihanannya sebagai software yang memakai basis kode komputer yang bisa didistribusikan secara terbuka (open source) sehingga pengguna bisa membuat aplikasi baru di dalamnya. Hal tersebut mengakibatkan banyaknya pengembang software yang berbondong-bondong mengembangkan aplikasi berbasis Android.

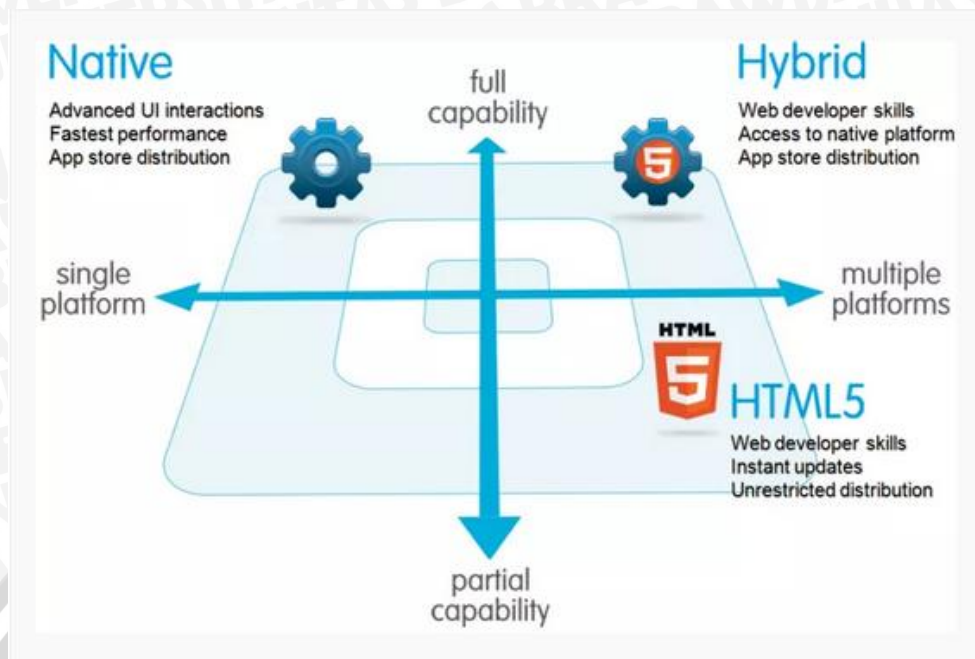
2.2.3 Aplikasi Hybrid

Hybrid Application adalah aplikasi yang sebagian terdiri dari Native Apps dan Web Apps (karena itu, banyak sebagian orang salah menyebutnya "Web Apps"). Seperti Native Apps, Aplikasi ini juga tersedia di App Store dan menggunakan fitur yang tersedia pada setiap perangkat seperti yang dilakukan Native Apps. Seperti Web Apps, aplikasi ini menggunakan kode HTML yang berjalan pada browser yang ditanamkan pada perangkat (Yudira, F., 2015).

Kelebihan Hybrid application:

1. Dapat berfungsi dalam kondisi baik terhubung jaringan (termasuk internet) maupun tidak.
2. Integrasi dengan file system perangkat.
3. Integrasi dengan Web-services.
4. Embed dengan browser untuk meningkatkan akses ke konten online secara dinamis.

Bila diilustrasikan dengan gambar maka seperti berikut ini:



Gambar 2. 1 Ilustrasi Aplikasi Hybrid

Dalam berhubungan data, Hybrid application bisa memiliki database server sendiri atau langsung melalui web service atau juga keduanya. Bila mendalami lebih lanjut mengenai hal ini, maka akan memahami alasan sebenarnya menggunakan format JSON dalam webservice bukan SOAP ataupun XML-RPC. Mungkin alasan itu pulalah yang mendasari twitter menggunakan REST-nya dengan JSON. Untuk penggunaannya JSON lebih ringan dikompres dengan JSONH dan dengan pengamanan enkripsi AES-256 atau RSA.

2.2.4 JQuery Mobile

JQuery Mobile merupakan framework yang digunakan untuk membangun aplikasi web mobile, dioptimalkan untuk perangkat layar sentuh. Seperti yang kita ketahui saat ini merupakan era peralihan dari perangkat desktop ke perangkat mobile seperti smartphone dan tablet. JQuery Mobile dapat dijalankan hampir di semua perangkat smartphone seperti *Android*, *Windows Phone*, *iOS*, *Blackberry*, *Symbian*, dan lain-lain, jadi kita tidak perlu terpacu pada salah satu merek atau model. Kita dapat mengembangkan satu aplikasi yang dapat didistribusikan ke semua perangkat. Pada dasarnya *jQuery Mobile* merupakan pengembangan ke arah mobile dari *jQuery*, yaitu dengan berdasarkan pada HTML5 dan CSS3, serta ditambah dengan *JavaScript* dan *Ajax*. Theming framework yang memungkinkan pembuatan tema sesuai keinginan (broari.com, 2014).

Berikut ini merupakan contoh pemangilan *file Javascript* dan *CSS jQuery Mobile* melalui *file HTML*:

1	<!DOCTYPE html>
2	<html>
3	<head>
4	<meta name="viewport" content="width=device-width, initial-
5	scale=1">

```

6 <link rel="stylesheet" href="jquery.mobile-1.4.2.css">
7 <script src="jquery.js"></script>
8 <script src="jquery.mobile-1.4.2.js"></script>
9 </head>
10 </html>

```

Gambar 2. 2 Pemanggilan File Javascript Dan CSS Jquery Mobile Melalui File HTML

2.2.5 PhoneGap

PhoneGap adalah sebuah kerangka kerja/framework open source yang dipakai untuk membuat aplikasi cross-platform mobile dengan HTML5, CSS, dan JavaScript. PhoneGap menjadi suatu solusi yang ideal untuk seorang web developer yang tertarik dalam pembuatan aplikasi di smartphone. dengan framework phonegap kita hanya melakukan satu kali coding langsung bisa di compile kesemua platform sekaligus (Pincuran.com, 2012).

Cara Kerja PhoneGap yaitu PhoneGap SDK menyediakan sebuah API, yaitu sebuah lapisan abstrak yang menyediakan pengembang dengan akses ke fitur hardware dan platform tertentu. PhoneGap menjelaskan bahwa dengan kode yang sama dapat digunakan pada sejumlah platform mobile dengan sedikit perubahan koding ataupun tidak, yang membuat aplikasi kita dapat dipakai untuk kalangan pengguna yang lebih luas.

	amazon- fireos	android	blackberry10	Firefox OS	ios	Ubuntu	wp7 (Windows Phone 7)	wp8 (Windows Phone 8)	win8 (Windows 8)	tizen
cordova CLI	✓ Mac, Windows, Linux	✓ Mac, Windows, Linux	✓ Mac, Windows	✓ Mac, Windows, Linux	✓ Mac	✓ Ubuntu	✓ Windows	✓ Windows	✓	✗
Embedded WebView	✓ (see details)	✓ (see details)	✗	✗	✓ (see details)	✓	✗	✗	✗	✗
Plug-in Interface	✓ (see details)	✓ (see details)	✓ (see details)	✗	✓ (see details)	✓	✓ (see details)	✓ (see details)	✓	✗
Platform APIs										
Accelerometer*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Capture*	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗
Compass*	✓	✓	✓	✗	✓ (3GS+)	✓	✓	✓	✓	✓
Connection*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Contacts*	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
Device*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Events	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
File*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
Geolocation*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Globalization*	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗
InAppBrowser*	✓	✓	✓	✗	✓	✓	✓	✓	(partial)uses iframe	✗
Media*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Notification*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Splashscreen*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
Storage	✓	✓	✓	✗	✓	✓	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓

Gambar 2. 3 Fitur yang Didukung PhoneGap pada Beberapa Sistem Operasi

2.2.6 Google Map API

Google maps API adalah fungsi fungsi pemrograman yang disediakan oleh Google maps agar Google maps bisa di integrasikan kedalam Web atau aplikasi yang sedang buat (Putra, C. A., 2012).

Google maps API sendiri menyediakan fungsi yang sangat banyak, berikut ini adalah pembagiannya:

- Google maps Javascript API v3
API ini tersedia dalam bahasa *javascript*.
- Google Maps API Webservice
Layanan Google maps API untuk fungsi lanjutan seperti *direction*, *Geocoding*, *Distance Matrix API* dan *elevation API*.
- Google Place API
Ini adalah API wajib yang harus dikuasai untuk membuat aplikasi *Location based service*.

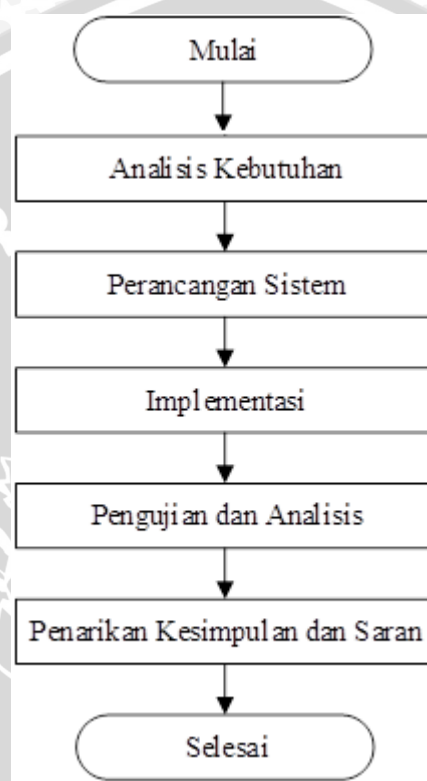
2.2.7 Pengujian Usability

Pengujian *Usability* merupakan pengujian untuk mengukur sejauh mana suatu produk dapat digunakan oleh pengguna untuk mencapai target yang ditetapkan dengan efektif, efisien, dan mencapai kepuasan pengguna.

Usability meliputi: *Learnability* (fungsi-fungsi dasar produk mudah dioperasikan sejak awal), *Efficiency* (fungsi-fungsi produk dapat dengan cepat dipergunakan), *Memorability* (setelah pengguna meninggalkan system dalam waktu lama, lalu bisa dengan mudah mengerti lagi cara penggunaannya), *Errors* (seberapa banyak dan serius kesalahan penggunaan produk, dan bagaimana pengguna dapat memperbaiki kesalahan), *Satisfaction* (seberapa puaskah pengguna?) (Shahab, Q., 2012).

BAB 3 METODOLOGI PENELITIAN

Dalam bab ini akan dijelaskan langkah-langkah yang nantinya akan dilakukan dalam perancangan, implementasi dan pengujian dari aplikasi yang akan dibuat. Metodologi penelitian ini terdiri dari beberapa tahapan, berikut merupakan diagram alir runtutan tahapan pengerjaan penelitian ini:



Gambar 3. 1 Diagram Alir Penelitian

3.1 Studi Literatur

Studi literatur dilakukan dengan mempelajari dan mengumpulkan data-data dari berbagai literatur yang dapat berasal dari buku, paper, dan dokumentasi dari internet. Data yang dimaksud merupakan data-data yang berkaitan dengan penelitian terkait yang pernah dilakukan sebelumnya, serta dasar teori yang diperlukan dalam proses pembuatan perangkat lunak.

Pembahasan terhadap kajian pustaka dilakukan dengan menganalisis penelitian yang pernah dilakukan sebelumnya dan memiliki kemiripan topik pembahasan dengan penelitian yang sedang dilakukan. Sedangkan dasar teori memberikan pembahasan tentang teori-teori dasar yang digunakan dalam penelitian yang dilakukan terkait penggunaan perangkat lunak, teknologi, dan metode. Hal-hal yang menjadi bahan studi literatur adalah sebagai berikut:

- Pariwisata Pulau Lombok

- Penggunaan Perangkat Bergerak *Platform Android*
- Pengembangan Aplikasi *Hybrid*
- *jQuery Mobile*
- *PhoneGap*
- *Google Map API*
- Pengujian *Usability*

3.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk memperoleh kebutuhan yang diperlukan dari sistem yang akan dibuat. Analisis kebutuhan dilakukan dengan menganalisis semua kebutuhan sistem dan siapa saja yang terlibat di dalamnya. Selain itu, analisis dilakukan untuk mengetahui kondisi lapangan agar dapat mengetahui perangkat apa saja yang digunakan. Dengan menggunakan pendekatan berorientasi terstruktur (*Structured Oriented Programming*) menggunakan DFD (*Data Flow Diagram*). Analisis kebutuhan dalam penelitian ini dilakukan dengan mengidentifikasi kebutuhan aplikasi perangkat bergerak pariwisata pulau Lombok.

Proses analisis kebutuhan ini terdiri dari beberapa tahap yaitu dari analisis fitur pilihan, gambaran umum aplikasi, indentifikasi aktor, analisis kebutuhan fungsional dan non-fungsional. Fitur pilihan digunakan untuk dapat menentukan fungsi apa saja yang harus dapat ditangani oleh sistem, maka digunakan referensi yang berasal dari penelitian terkait yang telah dibahas sebelumnya. Gambaran umum aplikasi memberikan penjabaran tentang deskripsi aplikasi yang dibangun dan lingkungan aplikasi ketika nantinya digunakan.

Tahap indentifikasi aktor adalah tahap dimana pengidentifikasian unsur-unsur yang dapat berinteraksi dan yang dapat melakukan perubahan pada sistem. Identifikasi aktor menjelaskan deskripsi dari setiap aktor.

Analisis kebutuhan fungsional dilakukan untuk mengetahui fungsi apa saja yang harus ada dalam aplikasi dan sistem. Proses analisis dimulai dengan menjabarkan spesifikasi kebutuhan fungsional berdasarkan jenis aktor. Spesifikasi kebutuhan fungsional didapatkan dari fitur-fitur yang dipilih melalui referensi berupa penelitian atau aplikasi terkait. Setelah itu dilakukan pengujian validitas untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan yang dibutuhkan.

Sedangkan analisis kebutuhan non-fungsional merupakan proses analisis untuk menentukan atribut atau kualitas secara keseluruhan dari sistem. Kategori kebutuhan non-fungsional yang dibutuhkan adalah *usability*. Sama halnya seperti analisis kebutuhan fungsional, analisis kebutuhan non-fungsional juga dilakukan terhadap aplikasi perangkat bergerak sistem informasi pariwisata pulau Lombok.

3.3 Perancangan Perangkat Lunak

Analisis kebutuhan sistem dibutuhkan untuk mendukung kinerja sistem, dalam hal ini apakah sistem yang dibuat telah sesuai dengan kebutuhan atau belum karena sistem akan mendukung tercapainya tujuan. Langkah-langkah yang nantinya akan dikerjakan dalam tahap perancangan perangkat lunak yaitu:

3.3.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem dilakukan dengan merepresentasikan relasi dari masing-masing komponen penyusun sistem. Komponen-komponen tersebut yaitu perangkat *mobile* sebagai media dari sistem, kode program yang terdiri dari *HTML*, *CSS*, *PHP*, dan *Javascript*, serta *framework* yang terdiri dari *jQuery Mobile* dan *PhoneGap*.

3.3.2 Perancangan Arsitektur Diagram

Perancangan arsitektur diagram merupakan gambaran umum dalam pembuatan sistem yang akan dirancang. Nantinya perancangan ini didasarkan oleh semua fungsi yang telah didefinisikan pada *data flow diagram* pada kebutuhan fungsional.

3.3.3 Perancangan Basis Data

Perancangan ini dilakukan dengan merepresentasikan struktur penyimpanan data yang digunakan oleh sistem. Setelah rancangan basis data selesai dirancang kemudian dilanjutkan dengan menyusun struktur dari masing-masing tabel.

3.3.4 Perancangan Antarmuka

Perancangan desain antarmuka dilakukan dengan membuat *layout* dari masing-masing halaman. *Layout* tersebut berisi komponen-komponen halaman *web* seperti *header*, *content*, dan *button* yang akan ditampilkan pada halaman *web*.

3.3.5 Perancangan Algoritma

Perancangan algoritma dilakukan untuk mengetahui bagaimana aplikasi nantinya berjalan pada kode program yang ada.

3.4 Implementasi

Sistem akan dibangun menggunakan bahasa pemrograman *HTML*, *CSS*, *PHP* dan *JavaScript* dengan memanfaatkan beberapa perangkat lunak pembantu yaitu:

- Brackets*: Sebagai perangkat lunak yang digunakan untuk membangun aplikasi
- XAMPP v3.2.1*: Menghubungkan antara aplikasi yang dibuat dengan basisdata *MySQL*
- jQueryMobile 1.4.5*: Sebagai *framework* yang digunakan dalam membangun aplikasi

- d. *PhoneGap*: Sebagai *framework* untuk aplikasi *mobile* yang *cross platform* dengan menggunakan bahasa *HTML*, *Javascript*, dan *CSS*.
- e. *Integrated Development Environment (IDE) Eclipse*: Sebagai perangkat pembantu dalam menjalankan kode program yang telah dibuat dengan perangkat *smartphone*.

3.5 Pengujian dan Analisis

Pengujian dilakukan berdasarkan kebutuhan fungsional dan non-fungsional yang telah ditentukan. Pengujian kebutuhan fungsional nantinya dilakukan dengan menggunakan pengujian validitas, sedangkan pengujian kebutuhan non-fungsional dilakukan dengan menggunakan pengujian *usability*. Pengujian validasi nantinya dilakukan dengan menggunakan teknik pengujian *Black Box*. Untuk pengujian *usability* nantinya menggunakan model *System Usability Scale (SUS)* untuk mendapatkan evaluasi dari pengguna aplikasi.

SUS merupakan pengujian *usability* yang menggunakan metode kuesioner sebagai media pengujian. Pernyataan pada kuesioner SUS terdiri dari dua jenis yaitu pernyataan positif dan pernyataan negatif. Pernyataan positif terletak pada pernyataan nomor ganjil. Sedangkan pernyataan negatif terletak pada pernyataan nomor genap.

Pengujian *usability* dengan menggunakan metode SUS ini dilakukan dengan langkah-langkah sebagai berikut:

1. Memberikan penjelasan kepada responden mengenai Aplikasi Pariwisata Pulau Lombok
2. Memberikan tugas-tugas yang harus dikerjakan oleh responden meliputi:
 - Mencari tempat wisata berdasarkan kategori
 - Melihat peta lokasi
3. Memberikan kuesioner kepada responden

Setelah itu dilakukan proses perhitungan untuk mengetahui hasil dari kuisisioner telah disebar. Perhitungan hasil kuesioner SUS dilakukan dengan langkah sebagai berikut:

1. Perhitungan skor pada tiap pernyataan dilakukan dengan ketentuan:
 - a. Pernyataan negatif (nomor pernyataan 2, 4, 6, 8, dan 10) dihitung dengan Persamaan 3.1.

$$Skor = x - 1 \quad (3.1)$$

- b. Pernyataan positif (nomor pernyataan ganjil 1, 3, 5, 7 dan 9) dihitung dengan Persamaan 3.2.

$$Skor = 5 - x \quad (3.2)$$

Dimana x adalah nilai yang diberikan responden pada kusioner yang telah diberikan.

2. Lakukan penjumlahan terhadap semua skor pada masing-masing pernyataan.
3. Hasil dari langkah sebelumnya dikalikan dengan nilai 2,5.

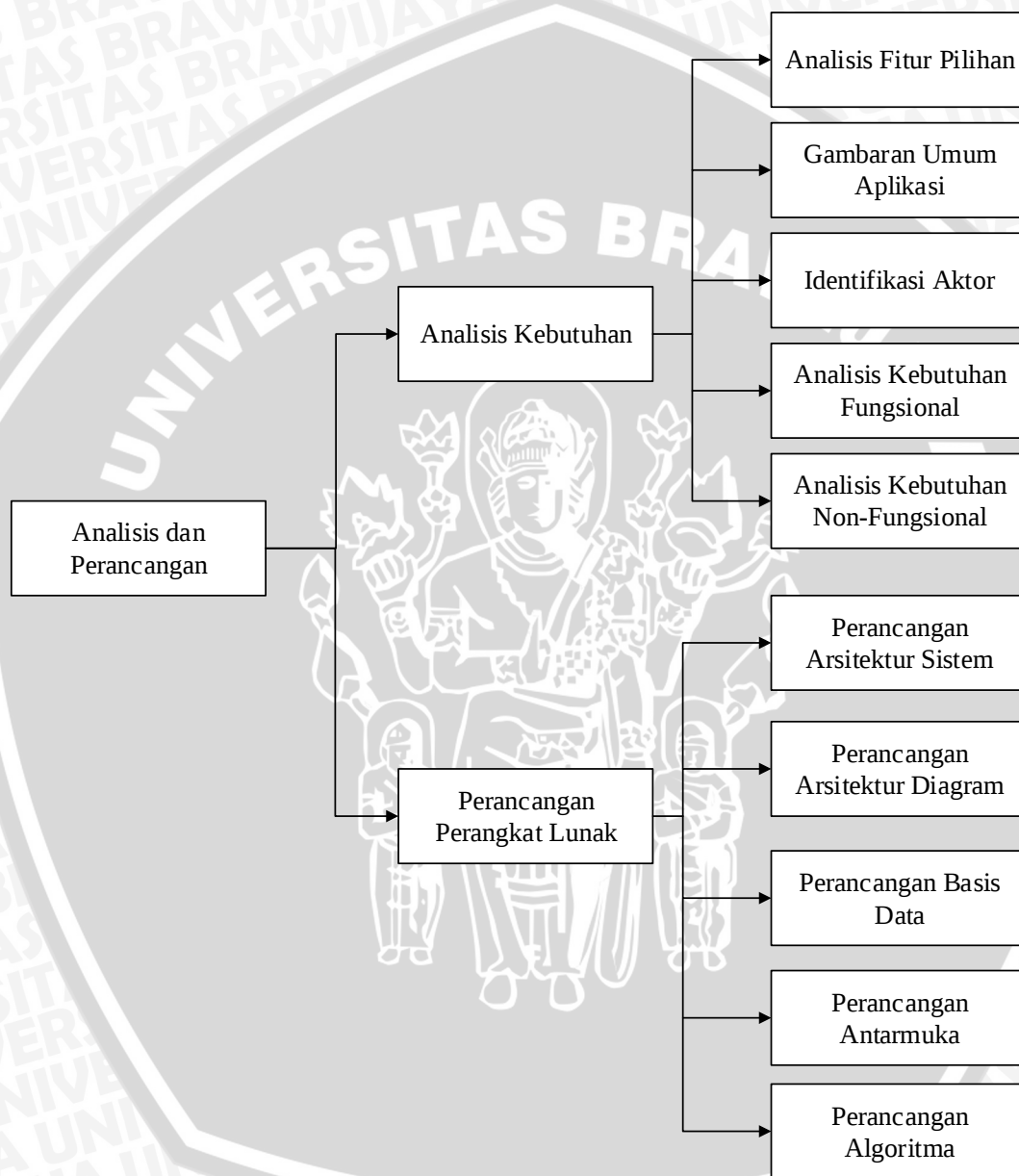
3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah tahap perancangan, implementasi dan pengujian telah selesai dilakukan. Kesimpulan dapat diambil dari hasil pengujian dan analisis terhadap sistem yang dibuat. Kemudian dilakukan pemberian saran yang ditujukan untuk menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan selanjutnya.



BAB 4 ANALISIS DAN PERANCANGAN

Pada bab ini dijelaskan mengenai proses perancangan yang dilakukan meliputi dua tahap yaitu analisis kebutuhan dan perancangan perangkat lunak. Tahapan proses analisis dan perancangan ditunjukkan dalam bentuk diagram pohon pada Gambar 4.1 berikut.



Gambar 4. 1 Diagram Pohon Analisis dan Perancangan

4.1 Analisis Kebutuhan

Analisis kebutuhan merupakan tahap untuk mendapatkan daftar kebutuhan-kebutuhan yang nantinya harus dapat ditangani oleh sistem. Proses analisis

kebutuhan terdiri dari analisis fitur pilihan, gambaran umum aplikasi, indentifikasi aktor, analisis kebutuhan fungsional dan non-fungsional. Untuk analisis kebutuhan fungsional nantinya merupakan proses analisis untuk mengetahui apa saja yang harus dapat ditangani oleh sistem dari permintaan pengguna aplikasi atau aktor. Sedangkan analisis kebutuhan non-fungsional nantinya merupakan proses analisis untuk mengetahui kemampuan apa saja yang harus dimiliki oleh sistem sehingga aplikasi memiliki kinerja yang baik ketika digunakan.

4.1.1 Analisis Fitur Pilihan

Analisis fitur pilihan merupakan proses analisis untuk mengetahui fungsi apa saja yang harus dapat ditangani oleh sistem dari pengguna aplikasi atau aktor. Untuk dapat menentukan fungsi apa saja yang harus dapat ditangani oleh sistem, maka digunakan referensi yang berasal dari penelitian terkait yang telah dibahas sebelumnya. Dari penelitian terkait tersebut diambil fungsi atau fitur yang pada umumnya dimiliki, kemudian diterapkan pada aplikasi yang akan dibangun. Tabel 4.1 menunjukkan judul penelitian beserta fiturnya yang nantinya akan diterapkan pada aplikasi yang akan dibangun.

Tabel 4. 1 Judul Penelitian Beserta Fitur yang Digunakan

No	Judul Penelitian	Fitur yang Digunakan
1	<i>A Mobile Tourist Guide for Trip Planning</i>	- Melihat daftar lokasi terdekat - Menampilkan deskripsi tempat - Menampilkan kategori
2	<i>Sistem Informasi Geografis Dengan Fitur Peta Dan Rute Perjalanan Studi Kasus Di Kabupaten Malang</i>	- Melihat Peta Lokasi - Melihat rute menuju lokasi

4.1.2 Gambaran Umum Aplikasi

Gambaran umum aplikasi memberikan penjelasan secara umum tentang deskripsi aplikasi yang dibangun serta lingkungan aplikasi ketika digunakan nantinya.

4.1.2.1 Deskripsi Aplikasi

Aplikasi *perangkat bergerak* yang dibangun bernama "iLombok" dan bertemakan wisata. Tujuan utama dari aplikasi ini yaitu sebagai media bagi pengguna atau wisatawan yang ingin mengetahui dan mencari tempat-tempat wisata di Lombok. Pengguna dapat mengetahui informasi tiap tempat wisata dan memanfaatkan fitur-fitur yang ada didalamnya.

Fitur yang dimiliki dalam aplikasi ini nantinya berupa pemanfaatan *Google Map API* seperti peta lokasi, navigasi dan pencarian wisata terdekat. Pengguna dapat menggunakan fitur navigasi sebagai pemandu jalan untuk mencapai tempat

wisata yang ingin dituju. Sedangkan fitur pencarian wisata terdekat dapat digunakan pengguna untuk melihat daftar tempat wisata lain berdasarkan jarak terdekat dari tempat wisata sebelumnya yang telah dilihat menggunakan aplikasi ini.

4.1.2.2 Lingkungan Aplikasi

Aplikasi yang dibuat dapat digunakan pada perangkat bergerak *smartphone* dengan sistem operasi *Android* sebagai media berjalannya aplikasi. Model pengembangan aplikasi yang digunakan yaitu pengembangan aplikasi *hybrid*.

Aplikasi membutuhkan koneksi *internet* pada perangkat *smartphone* yang digunakan untuk melakukan proses pertukaran data dengan *server*. Selain itu perangkat *smartphone* yang digunakan harus didukung dengan layanan *GPS* untuk dapat menggunakan fitur-fitur *Google Map API* yang telah disediakan oleh aplikasi.

Aplikasi yang dibuat ditujukan kepada orang-orang atau wisatawan yang ingin berlibur dan tertarik pada tempat-tempat wisata di pulau Lombok.

4.1.3 Identifikasi Aktor

Tahap identifikasi aktor dilakukan dengan menjabarkan aktor-aktor pengguna aplikasi beserta deskripsinya. Tabel 4.2 menunjukkan aktor yang terlibat dalam aplikasi.

Tabel 4. 2 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna aplikasi merupakan pengguna atau wisatawan yang ingin mencari informasi tempat wisata yang ingin diketahui dan pengguna akan mendapatkan informasi umum serta dapat memanfaatkan fitur-fitur yang ada. Fitur-fitur yang dimaksud adalah pengguna dapat menggunakan navigasi dan pencarian tempat wisata terdekat berdasarkan tempat wisata sebelumnya yang telah dikunjungi menggunakan aplikasi ini.

4.1.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan proses analisis untuk mengetahui fungsi apa saja yang harus dapat ditangani oleh sistem dari pengguna aplikasi atau aktor.

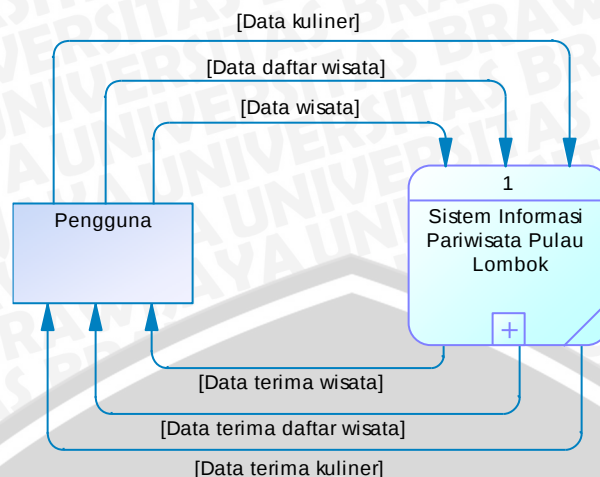
Proses analisis dimulai dengan menjabarkan spesifikasi kebutuhan fungsional berdasarkan jenis aktor. Spesifikasi kebutuhan fungsional didapatkan dari fitur-fitur yang dipilih melalui referensi berupa penelitian atau aplikasi terkait. Pada Tabel 4.3 di bawah ini menunjukkan spesifikasi kebutuhan fungsional pengguna.

Tabel 4. 3 Spesifikasi Kebutuhan Fungsional Pengguna

Nomor SRS	Kebutuhan Fungsional	Fungsi
SRS_A_001	Aplikasi harus menyediakan fasilitas untuk menampilkan halaman beranda kepada pengguna.	Melihat Halaman Beranda
SRS_A_002	Aplikasi harus menyediakan fasilitas untuk mencari tempat wisata berdasarkan kategori dan menampilkan informasi tempat wisata sesuai dengan yang dipilih pengguna.	Mencari Tempat Wisata
SRS_A_003	Aplikasi harus menyediakan fasilitas untuk mencari tempat kuliner berdasarkan kategori dan menampilkan informasi tempat kuliner sesuai dengan yang dipilih pengguna.	Mencari Tempat Kuliner
SRS_A_004	Aplikasi harus menyediakan fasilitas untuk fitur navigasi yang digunakan untuk melihat rute lokasi tujuan ke tempat wisata yang diinginkan pengguna.	Melihat Rute Lokasi Tujuan
SRS_A_005	Aplikasi harus menyediakan fasilitas untuk fitur peta lokasi yang digunakan untuk menampilkan letak lokasi tempat wisata yang diinginkan pengguna.	Melihat Peta Lokasi
SRS_A_006	Aplikasi harus menyediakan fasilitas untuk fitur wisata terdekat yang digunakan untuk melihat daftar wisata terdekat berdasarkan jarak dari tempat wisata yang telah dipilih pengguna sebelumnya dalam aplikasi.	Melihat Daftar Wisata Terdekat

4.1.4.1 Diagram Context (DC)

Konteks diagram mencakup suatu simbol proses yang mewakili sistem informasi pariwisata pulau Lombok dengan entitas pengguna yang mempengaruhinya. Konteks diagram ini ditunjukkan pada Gambar 4.2.



Gambar 4. 2 *Diagram Context Sistem*

Pada diagram konteks menjelaskan mengenai data yang mengalir pada proses *request* dan *output*. Maksud data *request* dari sistem ini adalah permintaan data yang dilakukan oleh pengguna pada aplikasi yaitu data wisata, data daftar wisata, dan data kuliner. Didalam sistem semua data *request* diolah sehingga mendapatkan data *output* sesuai dengan permintaan pengguna. Data *output* tersebut berupa data terima wisata, data terima daftar wisata, dan data kuliner.

4.1.4.2 Data Flow Diagram Level 1

Pemodelan *data flow diagram* (DFD) ini diperlukan karena perancangan aplikasi ini menggunakan *structured oriented programming*, serta untuk memahami aplikasi ini secara logika, terstruktur dan jelas. Data flow diagram dibuat untuk menunjukkan proses fungsional aplikasi yang saling berhubungan satu sama lain dengan alur data dan mengenai bagaimana data ditransformasikan melalui sistem serta menggambarkan fungsi-fungsi yang mentransformasi aliran data.

Berdasarkan data flow diagram pada Gambar 4.3 dijelaskan bahwa sistem informasi pariwisata pulau lombok berbasis perangkat bergerak ini memiliki 6 fungsi yang terdiri dari melihat daftar wisata, memilih kategori wisata, memilih kategori kuliner, melihat daftar kuliner, memilih tempat wisata dan memilih tempat kuliner.

Untuk melihat tempat wisata dan melihat tempat kuliner akan dijabarkan pada *data flow diagram level 2*. *Data flow diagram level 1* sistem informasi pariwisata pulau Lombok ditunjukkan pada Gambar 4.3.

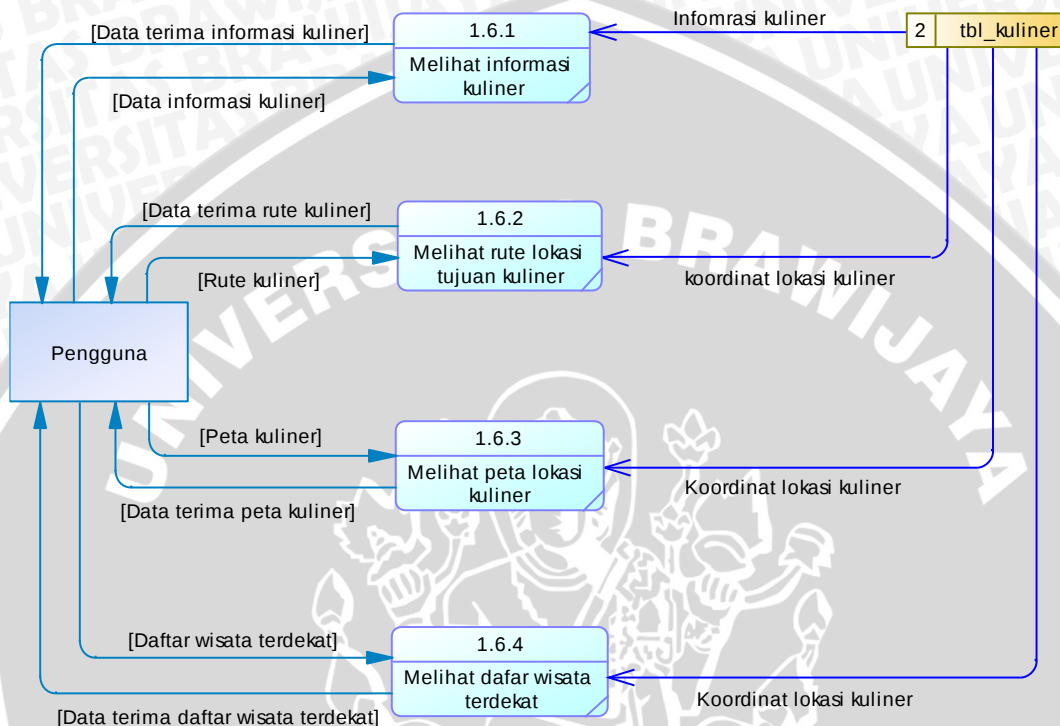


4.1.4.3 Data Flow Diagram Level 2 Memilih Tempat Wisata



Gambar 4.4 merupakan data flow diagram level 2 dari memilih tempat wisata. DFD ini menggambarkan fungsi aplikasi untuk melihat detail tempat wisata. Pada fungsi memilih tempat wisata, terdapat 4 fungsi didalamnya yaitu melihat informasi wisata, melihat rute lokasi tujuan wisata, melihat peta lokasi wisata dan melihat daftar wisata terdekat.

4.1.4.4 Data Flow Diagram Level 2 Memilih Tempat Kuliner



Gambar 4. 5 Data Flow Diagram Level 2 Memilih Tempat Kuliner

Gambar 4.5 merupakan data flow diagram level 2 dari memilih tempat kuliner. DFD ini menggambarkan fungsi aplikasi untuk melihat detail tempat kuliner. Pada fungsi memilih tempat kuliner, terdapat 4 fungsi didalamnya yaitu melihat informasi kuliner, melihat rute lokasi tujuan kuliner, melihat peta lokasi kuliner dan melihat daftar wisata terdekat.

4.1.5 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dilakukan dengan cara menganalisis kemampuan apa yang harus dimiliki oleh sistem sehingga memiliki kinerja yang baik ketika digunakan nanti. Kateogori kebutuhan non-fungsional yang ditentukan yaitu *usability*. Pada tabel 4.4 di bawah ini dijelaskan spesifikasi kebutuhan non-fungsional dari sistem.

Tabel 4. 4 Spesifikasi Kebutuhan Non-Fungsional

Parameter	Deskripsi Kebutuhan
<i>Usability</i>	Aplikasi harus dapat memberikan manfaat dan kepuasan bagi penggunanya, serta memberikan kemudahan dalam

	menggunakan aplikasi dengan teger skor <i>System Usability Scale</i> (SUS) ≥ 68 .
--	--

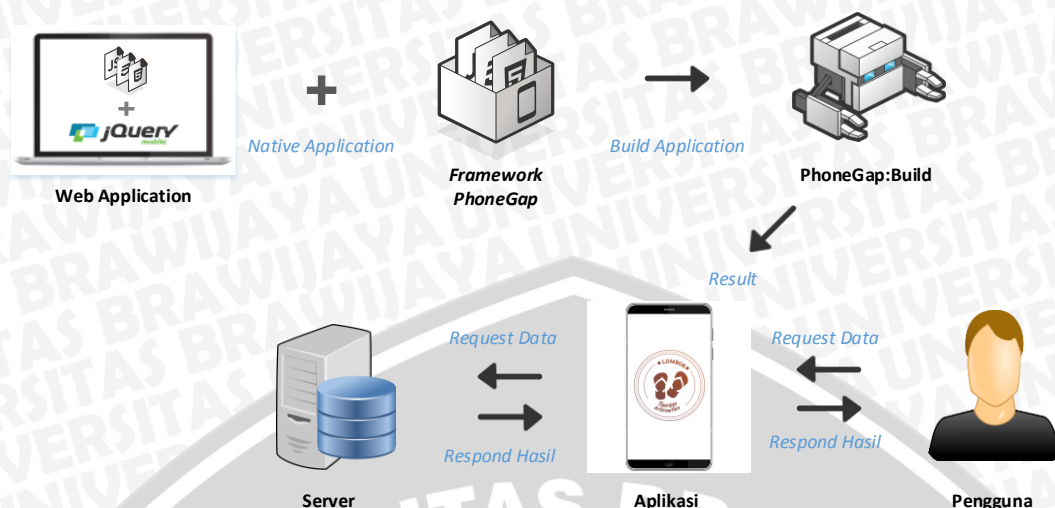
4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan dalam empat tahap, yaitu perancangan arsitektural sistem, perancangan arsitektur diagram, perancangan basis data, dan perancangan antarmuka.

4.2.1 Perancangan Arsitektur Sistem

Aplikasi informasi pariwisata pulau Lombok berbasis perangkat bergerak dibangun dengan menggunakan konsep *Hybrid Mobile Application* pada perangkat *Android*. Dikembangkan menggunakan bahasa pemrograman web yaitu *HTML*, *CSS*, dan *JavaScript*. Untuk memudahkan penyesuaian antarmuka aplikasi yang dibuat menggunakan bahasa pemrograman web ketika nantinya diaplikasikan pada perangkat bergerak digunakan *Framework jQuery Mobile*. Pengimplementasian konsep *Hybrid Mobile Application* pada aplikasi ini menggunakan *Framework PhoneGap*. *Framework PhoneGap* memungkinkan aplikasi web yang telah dibuat dapat berfungsi sebagai aplikasi *native* yang bisa menjalankan fitur yang dimiliki oleh perangkat bergerak. *PhoneGap:Build* merupakan tahap membangun aplikasi web menjadi aplikasi *native* dengan menggunakan *framework PhoneGap*.

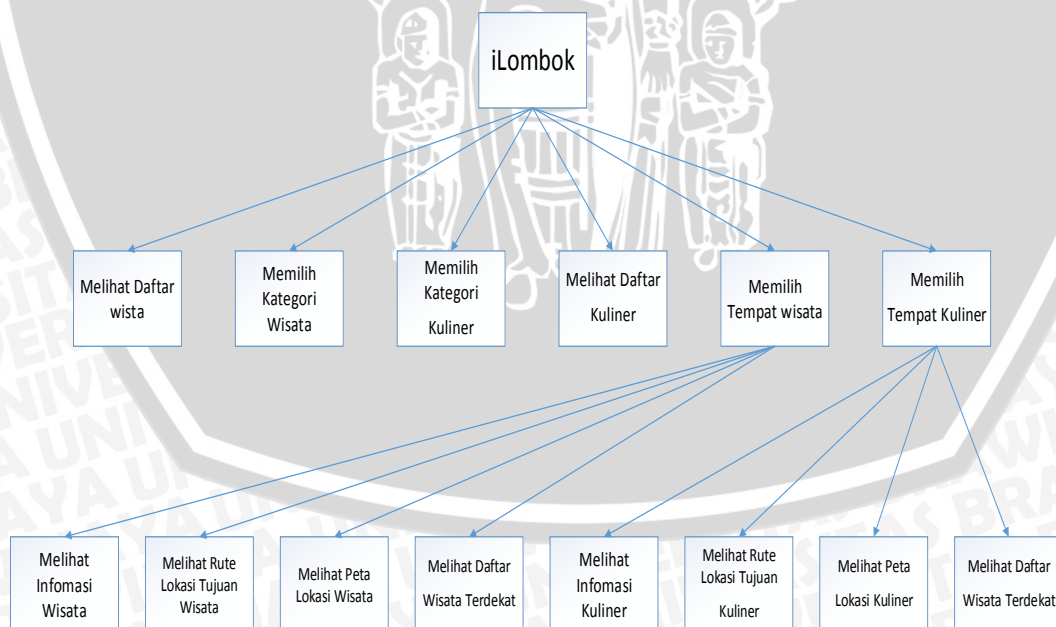
Setelah berhasil membangun aplikasi, dilakukan tahap perancangan umum pada sistem. Tahap perancangan umum sistem dilakukan dengan merepresentasikan perancangan sistem yang dibuat secara umum terkait dengan proses kerja antara pengguna, pusat data (*server*), dan media yang digunakan. Nantinya untuk dapat mengakses *server*, pengguna harus menggunakan perangkat *bergerak berplatform Android* sebagai media menggunakan aplikasi ini. Perangkat *bergerak* berperan sebagai penerima permintaan data dari pengguna, kemudian menerjemahkan permintaan tersebut untuk diteruskan ke *server*. Setelah *server* menerima permintaan data maka *server* akan memberikan respon yang akan ditampilkan melalui perangkat bergerak. Representasi perancangan arsitektur sistem ditunjukkan pada Gambar 4.6.



Gambar 4. 6 Perancangan Arsitektur Sistem

4.2.2 Perancangan Arsitektur Diagram

Perancangan arsitektur diagram merupakan gambaran umum dalam pembuatan sistem yang akan dirancang. Perancangan ini didasarkan oleh semua fungsi yang telah didefinisikan pada data flow diagram level 1 dan level 2. Pada sistem informasi pariwisata pulau Lombok berbasis perangkat bergerak (iLombok) ini memiliki 6 fungsi. 6 fungsi tersebut diantaranya melihat daftar wisata, memilih kategori wisata, memilih kategori kuliner, melihat daftar kuliner, memilih tempat wisata dan memilih tempat kuliner. Dimana fungsi memilih tempat wisata dan memilih tempat kuliner memiliki fungsi-fungsi didalamnya. Perancangan arsitektur diagram ditunjukkan pada Gambar 4.7.



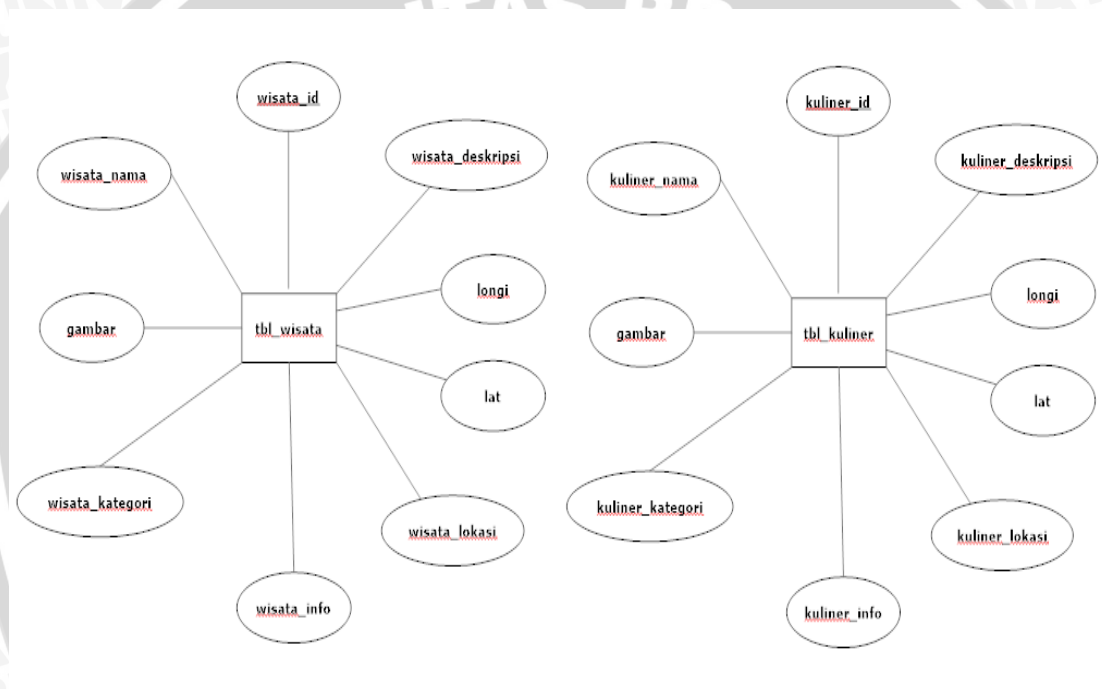
Gambar 4. 7 Perancangan Arsitektur Sistem

4.2.3 Perancangan Basis Data

Aplikasi menggunakan basis data yang terpusat pada server sebagai media penyimpanan. Proses perancangan basis data diawali dengan merepresentasikan struktur penyimpanan data yang digunakan oleh sistem.

2 entitas yaitu *tbl_kuliner* dan *tbl_wisata* dengan masing-masing atributnya dipilih untuk memungkinkan data-data yang diperlukan dapat disimpan dengan baik tanpa menyebabkan kehilangan informasi.

Entitas *tbl_kuliner* dan *tbl_wisata* berisi atribut-atribut yang digunakan untuk menunjang penyampaian informasi wisata dan kuliner. Namun antar entitas tidak memiliki hubungan satu sama lain. Gambar 4.8 dibawah ini menunjukkan Perancangan Basis Data.



Gambar 4. 8 Perancangan Basis Data

Berdasarkan perancangan basis data yang telah dirancang, terdapat 2 entitas yaitu *tbl_kuliner* dan *tbl_wisata*. Masing-masing entitas tersebut merepresentasikan tabel di dalam basis data. Sedangkan atribut merupakan representasi *field* dari masing-masing tabel. Berikut ini merupakan struktur dari masing-masing tabel yang terdapat dalam basis data.

4.2.3.1 Struktur Tabel Wisata

Entitas *tbl_wisata* merepresentasikan tabel wisata di dalam basis data. Tabel wisata mencakup semua kebutuhan informasi dalam wisata. Tabel 4.5 menampilkan bagaimana struktur dari tabel wisata.

Tabel 4. 5 Struktur Tabel Wisata

No.	Nama Field	Tipe Data	Keterangan
1	wisata_id	Int	Kode wisata
2	wisata_nama	Text	Nama wisata
3	Gambar	Text	Gambar wisata
4	wisata_kategori	Text	Kategori wisata
5	wisata_lokasi	Text	Lokasi wisata
6	wisata_deskripsi	Varchar	Deskripsi tempat wisata
7	Lat	Text	Latitude lokasi wisata
8	Longi	Text	Longitude lokasi wisata
9	wisata_info	Text	Informasi tempat wisata

4.2.3.2 Struktur Tabel Kuliner

Entitas `tbl_kuliner` merepresentasikan tabel kuliner di dalam basis data. Tabel wisata mencakup semua kebutuhan informasi dalam kuliner. Tabel 4.6 menampilkan bagaimana struktur dari tabel kuliner.

Tabel 4. 6 Struktur Tabel Kuliner

No.	Nama Field	Tipe Data	Keterangan
1	kuliner_id	Int	Kode kuliner
2	kuliner_nama	Text	Nama kuliner
3	gambar	Text	Gambar kuliner
4	kuliner_kategori	Text	Kategori kuliner
5	kuliner_lokasi	Text	Lokasi kuliner
6	kuliner_deskripsi	Varchar	Deskripsi tempat kuliner
7	Lat	Text	Latitude lokasi kuliner
8	Longi	Text	Longitude lokasi kuliner
9	kuliner_info	Text	Informasi menu favorit dan harga

4.2.4 Perancangan Antarmuka

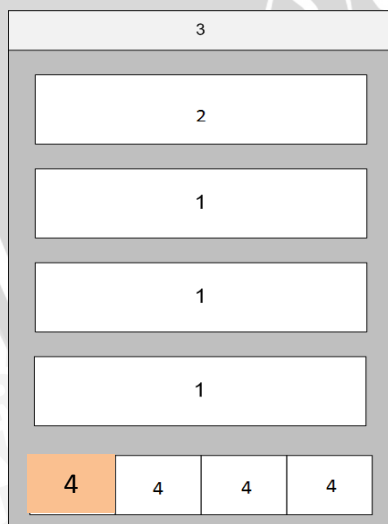
Proses perancangan antarmuka diawali dengan membuat *sitemap* dari aplikasi, kemudian dilanjutkan dengan menggambarkan *layout* antarmuka halaman aplikasi. Gambar 4.9 berikut menunjukkan *sitemap* dari aplikasi Pariwisata Pulau Lombok.



Gambar 4. 9 Sitemap Aplikasi

4.2.4.1 Halaman Beranda

Halaman beranda merupakan tampilan awal ketika pengguna menjalankan aplikasi. Halaman ini menampilkan *daftar* beranda. Gambar 4.10 di bawah ini menampilkan perancangan antarmuka halaman beranda.



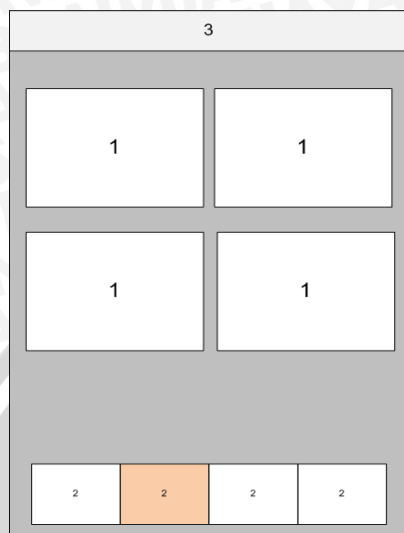
Keterangan:

1. Daftar Wisata
2. Searchbox
3. Header
4. Tab Beranda

Gambar 4. 10 Antarmuka Halaman Beranda

4.2.4.2 Halaman Kategori Wisata

Halaman kategori wisata merupakan halaman ketika pengguna memilih bar navigasi kategori wisata. Halaman ini menampilkan *sub menu* kategori wisata. Gambar 4.11 di bawah ini menampilkan perancangan antarmuka halaman Kategori Wisata.



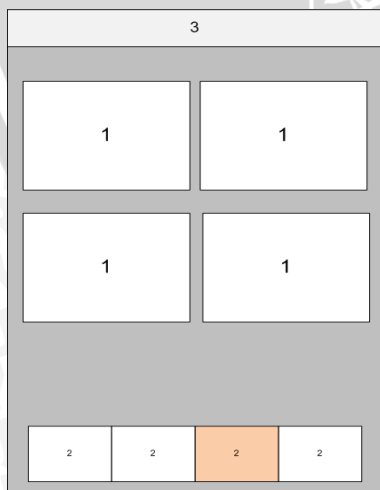
Keterangan:

1. *Sub Menu* Kategori Wisata
2. Tab Wisata
3. *Header*

Gambar 4. 11 Antarmuka Halaman Kategori Wisata

4.2.4.3 Halaman Kategori Kuliner

Halaman kategori wisata merupakan halaman ketika pengguna memilih bar navigasi kategori kuliner. Halaman ini menampilkan *sub menu* kategori kuliner. Gambar 4.12 di bawah ini menampilkan perancangan antarmuka halaman Kategori Kuliner.



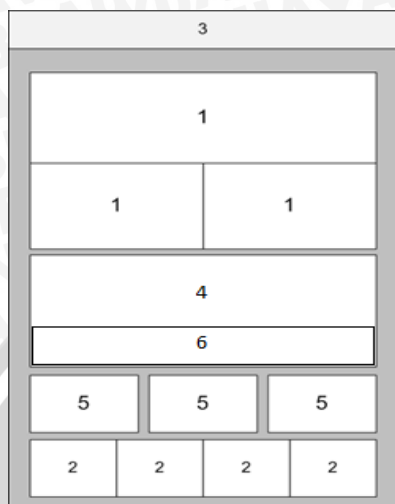
Keterangan:

1. *Sub Menu* Kategori Kuliner
2. Tab Kuliner
3. *Header*

Gambar 4. 12 Antarmuka Halaman Kategori Kuliner

4.2.4.4 Halaman Detail

Tampilan halaman detail wisata sama dengan halaman detail kuliner, hanya dibedakan dari segi isi informasinya. Halaman ini menampilkan *detail* informasi dari tempat wisata atau kuliner yang dipilih pengguna. Gambar 4.13 di bawah ini menampilkan perancangan antarmuka halaman Detail Wisata dan Kuliner.



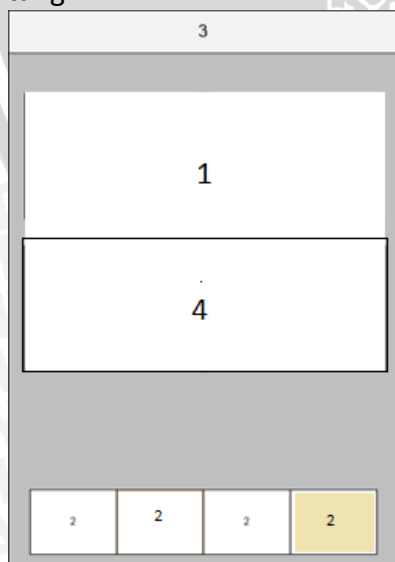
Keterangan:

1. Gambar
2. Bar Navigasi
3. Header
4. Deskripsi
5. Fitur
6. Informasi / Menu Favorit

Gambar 4. 13 Antarmuka Halaman *Detail* Wisata dan Kuliner

4.2.4.5 Halaman Tentang

Halaman tentang ini menampilkan deskripsi aplikasi dan profil pembuat aplikasi. Gambar 4.14 di bawah ini menampilkan perancangan antarmuka halaman tentang.



Keterangan:

1. Logo Aplikasi
2. Bar Navigasi
3. Header
4. Deskripsi Aplikasi

Gambar 4. 14 Antarmuka Halaman Tentang

4.2.5 Perancangan Algoritma

Aplikasi iLombok ini mempunyai beberapa proses (*method*) utama. Pada penulisan skripsi ini hanya dicantumkan algoritma dari beberapa proses saja sehingga tidak semua algoritma method akan dicantumkan. Algoritma proses yang dicantumkan antara lain adalah Mencari Tempat Kuliner, Mencari Tempat Wisata, dan Melihat Daftar Wisata Terdekat (*nearby*).

a. Algoritma Mencari Tempat Wisata

Mencari tempat wisata dilakukan dengan pengguna menekan bar navigasi wisata pada aplikasi. Algoritma mencari tempat wisata akan dijalankan saat pengguna telah memilih salah satu sub menu kategori wisata yang ingin dilihat. Gambar 4.15 merupakan algoritma untuk mencari tempat wisata.

NAMA ALGORITMA : Mencari Tempat Wisata

DEKLARASI

- var link : untuk menyimpan link halaman yang sedang aktif
- var kategori : untuk menyimpan kategori wisata yang didapatkan dari variabel link

DESKRIPSI

Proses :

1. melakukan request ke sever menggunakan ajax dengan parameter kategori
2. melakukan pengecekan terhadap hasil request:
 - a. jika hasilnya kosong maka akan ditampilkan string 'Kosong'
 - b. jika ada hasilnya maka hasil ditampilkan dengan atribut-atribut yang sesuai dengan database

Keluaran : daftar wisata yang sesuai dengan kategori wisata yang dipilih.

Gambar 4. 15 Algoritma Mencari Tempat Wisata

b. Algoritma Mencari Tempat Kuliner

Mencari tempat kuliner dilakukan dengan pengguna menekan bar navigasi kuliner pada aplikasi. Algoritma mencari tempat kuliner akan dijalankan saat pengguna telah memilih salah satu sub menu kategori kuliner yang ingin dilihat. Gambar 4.16 merupakan algoritma untuk mencari tempat kuliner.

NAMA ALGORITMA : Mencari Tempat Kuliner

DEKLARASI

- var link : untuk menyimpan link halaman yang sedang aktif
- var kategori : untuk menyimpan kategori kuliner yang didapatkan dari variabel link

DESKRIPSI

Proses :

1. Melakukan request ke sever menggunakan ajax dengan parameter kategori.
2. Melakukan pengecekan terhadap hasil request:
 - a. Jika hasilnya kosong maka akan ditampilkan string 'Kosong'.
 - b. Jika ada hasilnya maka hasil ditampilkan dengan atribut-atribut yang sesuai dengan database.

Keluaran : daftar kuliner yang sesuai dengan kategori wisata yang dipilih.

Gambar 4. 16 Algoritma Mencari Tempat Kuliner

c. Algoritma Melihat Daftar Wisata Terdekat

Melihat daftar wisata terdekat dilakukan dengan pengguna memilih tombol wisata terdekat (Nearby) di detail wisata atau kuliner pada aplikasi. Algoritma melihat daftar wisata terdekat akan dijalankan saat pengguna menekan tombol wisata terdekat (Nearby). Gambar 4.17 merupakan algoritma untuk melihat daftar wisata terdekat.

NAMA ALGORITMA : Melihat Daftar Wisata Terdekat

DEKLARASI

- latitude : menentukan nilai latitude
- longitude : menentukan nilai longitude
- session page : menyimpan halaman sebelumnya pada session

DESKRIPSI

Proses :

1. melakukan request location ke server dengan parameter longitude longitude
2. setelah berhasil melakukan request, data ditampilkan dalam bentuk list pada halaman yang terdapat didalam session page

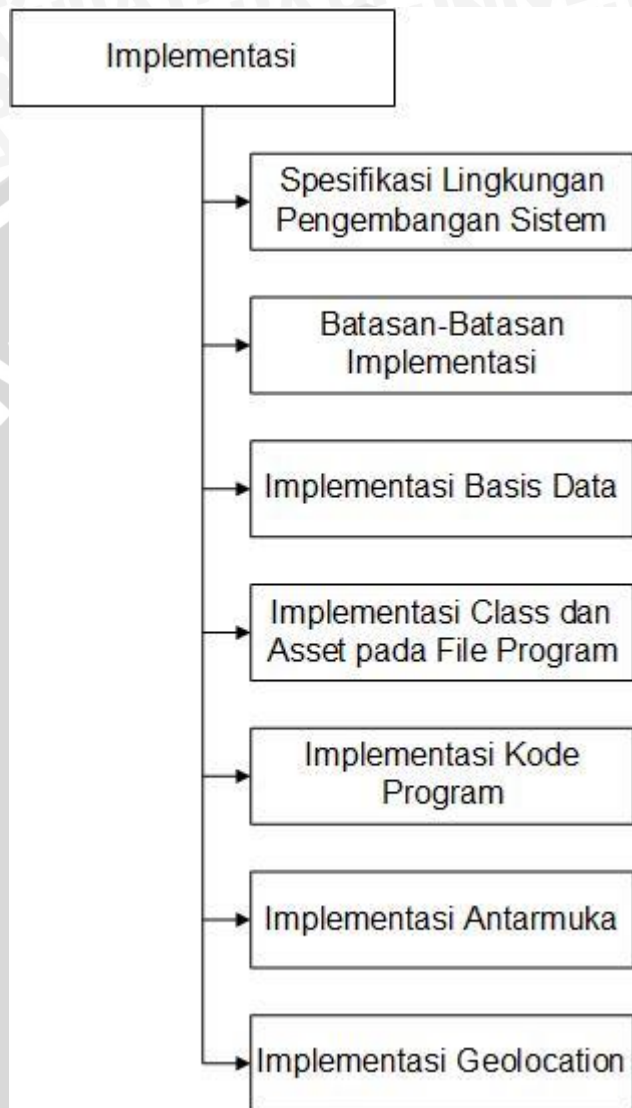
Keluaran : daftar wisata yang diurutkan berdasarkan jarak terdekat

Gambar 4. 17 Algoritma Melihat Daftar Wisata terdekat



BAB 5 IMPLEMENTASI

Bab ini membahas implementasi dari aplikasi yang dibangun. Diagram dari tahap implementasi ditunjukkan pada Gambar 5.1.



Gambar 5. 1 Diagram Pohon Implementasi

5.1 Implementasi

Implementasi merupakan proses perwujudan hasil dari proses analisis dan perancangan sistem yang telah dilakukan. Proses implementasi dilakukan dengan menjabarkan spesifikasi lingkungan pengembangan sistem, batasan implementasi, implementasi basis data, implementasi *class* dan *asset* pada *file* program, implementasi kode program, implementasi antarmuka, dan implementasi *geolocation*.

5.1.1 Spesifikasi Lingkungan Pengembangan Sistem

Spesifikasi lingkungan pengembangan sistem menjelaskan spesifikasi dari perangkat keras dan perangkat lunak yang digunakan dalam proses pembuatan dan implementasi dari aplikasi.

5.1.1.1 Spesifikasi Perangkat Keras

Proses pembuatan aplikasi dilakukan dengan menggunakan perangkat komputer dengan spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.1.

Tabel 5. 1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
<i>System Model</i>	Acer Aspire 4745G
<i>Processor</i>	Intel(R) Core(TM) i5 CPU M 460 @ 2.53GHz (4 CPUs), ~2.5GHz
<i>Memory</i>	8192MB RAM
<i>Display</i>	ATI Mobility Radeon HD 5650

Sedangkan proses implementasi aplikasi digunakan perangkat *smartphone* dengan spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.2.

Tabel 5. 2 Spesifikasi Perangkat Keras *Smartphone*

Nama Komponen	Spesifikasi
<i>System Model</i>	Lenovo A6000 Plus
<i>Processor</i>	Qualcomm MSM8916 Snapdragon 410, Quad-core 1.2 GHz Cortex-A53
<i>Memory</i>	2 GB RAM
<i>Display</i>	720 x 1280 pixels, 5.0 inches

5.1.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak dari komputer yang digunakan dalam proses pembuatan aplikasi ditunjukkan pada Tabel 5.3.

Tabel 5. 3 Spesifikasi Perangkat Lunak Komputer

Nama Komponen	Spesifikasi
Sistem Operasi	Microsoft Windows 7 64-bit SP-1
<i>IDE (Integrated Development Environmet)</i>	Eclipse Juno dengan plug-in ADT (Android Development Tools)
<i>Programming Environment</i>	XAMPP

Database Management System	MySQL
----------------------------	-------

Spesifikasi perangkat lunak dari *smartphone* yang digunakan dalam proses implementasi aplikasi ditunjukkan pada Tabel 5.4.

Tabel 5. 4 Spesifikasi Perangkat Lunak *Smartphone*

Nama Komponen	Spesifikasi
Sistem Operasi	Android OS v5.0.2 (Lollipop)

Spesifikasi *tools* pengembangan yang digunakan dalam proses aplikasi ditunjukkan pada Tabel 5.5.

Tabel 5. 5 Spesifikasi *Tools* Pengembangan

Nama Komponen	Spesifikasi
<i>Programming Language</i>	PHP, Javascript, CSS, HTML
<i>Editor Tools</i>	Brackets Build 1.5.0-1653
<i>Framework</i>	jQuery Mobile 1.4.5, PhoneGap 2.9.0

5.1.2 Batasan-Batasan Implementasi

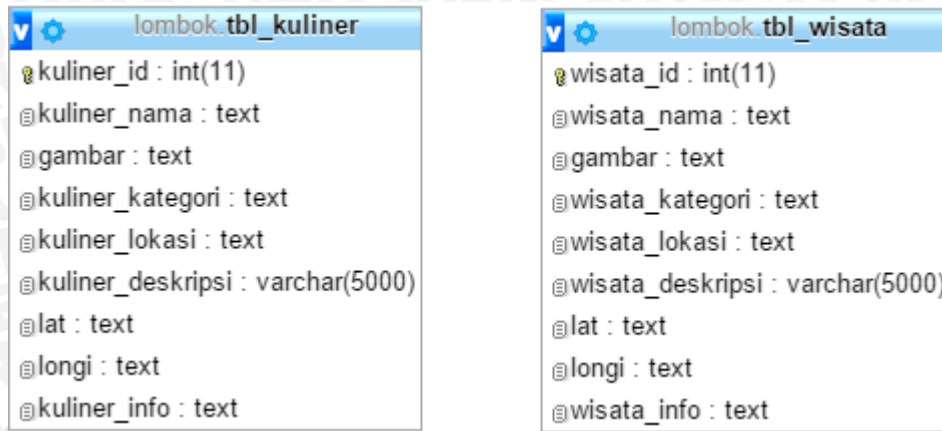
Dalam proses implementasi terdapat beberapa batasan yang diterapkan diantaranya yaitu:

1. Aplikasi dibuat dengan model pengembangan aplikasi *hybrid* dan hanya untuk perangkat *Android* dengan sistem operasi *Android* minimal versi 3.0 (*Honeycomb*).
2. Aplikasi dibuat menggunakan bahasa pemrograman *HTML*, *CSS*, *JavaScript*, dan *PHP* serta menggunakan *framework jQuery Mobile* untuk memudahkan penyesuaian antarmuka aplikasi yang dibuat menggunakan bahasa pemrograman *web* ketika nantinya diaplikasikan pada perangkat bergerak.
3. *Framework PhoneGap* digunakan untuk memungkinkan aplikasi yang telah dibuat dapat berfungsi sebagai aplikasi *native* dan dapat menggunakan fitur yang tersedia pada perangkat *smartphone*.
4. Aplikasi menggunakan media penyimpanan data yang terpusat pada pusat data (*server*).
5. Aplikasi dapat berjalan pada perangkat bergerak *Android* yang memiliki koneksi *internet* untuk dapat melakukan proses pertukaran data serta menggunakan fitur *Google Map API*.

5.1.3 Implementasi Basis Data

Implementasi basis data dilakukan dengan mengaplikasikan hasil perancangan basis data pada media penyimpanan data yang digunakan oleh aplikasi. Aplikasi

menggunakan media penyimpanan terpusat yang terdapat pada server. Implementasi basis data direpresentasikan menggunakan physical diagram yang ditunjukkan pada Gambar 5.2.



Gambar 5. 2 Physical Diagram

5.1.3.1 Tabel Wisata

Query create table *tbl_wisata* ditunjukkan pada Gambar 5.3 di bawah ini.

```

1 CREATE TABLE tbl_wisata (
2     wisata_id INTEGER NOT NULL,
3     wisata_nama TEXT,
4     gambar TEXT,
5     wisata_kategori TEXT,
6     wisata_lokasi TEXT,
7     wisata_deskripsi VARCHAR(5000),
8     lat TEXT,
9     longi TEXT,
10    wisata_info TEXT
11 );

```

Gambar 5. 3 Query Create Table Wisata

5.1.3.2 Tabel Kuliner

Query create table *tbl_kuliner* ditunjukkan pada Kode 5.4 di bawah ini.

```

1 CREATE TABLE tbl_kuliner (
2     kuliner_id INTEGER NOT NULL,
3     kuliner_nama TEXT,
4     gambar TEXT,
5     kuliner_kategori TEXT,
6     kuliner_lokasi TEXT,
7     kuliner_deskripsi VARCHAR(5000),
8     lat TEXT,
9     longi TEXT,
10    kuliner_info TEXT
11 );

```

Gambar 5. 4 Query Create Table Kuliner

5.1.4 Implementasi *Class* dan *Assets* pada *File* Program

Komponen-komponen penyusun aplikasi atau *assets* diimplementasikan dalam bentuk *file* program dengan format *extension file* *.html, *.js dan *.php.

5.1.4.1 Implementasi *Class* pada Kode Program *.java

Class MainActivity.java merupakan *file* pertama yang dijalankan ketika proses membangun aplikasi *native* oleh PhoneGap. MainActivity.java memungkinkan aplikasi agar dapat berjalan pada perangkat bergerak *Android*. Implementasi *class* pada kode program *.java ditunjukkan pada Tabel 5.6.

Tabel 5. 6 Implementasi *Class* pada Kode Program *.java

Folder	Package	Nama	Nama File Program
src	com.lombok	Main Activity	MainActivity.java

5.1.4.2 Implementasi *Class* pada Kode Program *.html

Fileasset dengan *extension* *.html berfungsi untuk membangun antarmuka dari aplikasi. Implementasi *assets* pada kode program *.html ditunjukkan pada Tabel 5.7.

Tabel 5. 7 Implementasi *Assets* pada Kode Program *.html

Folder	Nama	Nama File Program
assets/www	Index	index.html
assets/www	Kategori Wisata	kategori-wisata.html
assets/www	Kategori Kuliner	kuliner.html
assets/www	Daftar Kategori Wisata	daftar-wisata-kategori.html
assets/www	Daftar Kategori Kuliner	daftar-wisata-kuliner.html
assets/www	Detail Wisata	detail-wisata.html
assets/www	Detail Kuliner	detail-kuliner.html
assets/www	Map Wisata	map-page.html
assets/www	Map Kuliner	map-kuliner.html
assets/www	Navigasi Wisata	nav-wisata.html
assets/www	Navigasi Kuliner	nav-kuliner.html
assets/www	Wisata Terdekat	map-terdekat.html
assets/www	Main	main.html

5.1.4.3 Implementasi Assets pada Kode Program *.js

Fileassets dengan *extension* *.js berfungsi untuk menjalankan proses-proses yang terdapat dalam fungsi sesuai dengan kebutuhan dari aplikasi ketika aplikasi dijalankan. Implementasi *assets* pada kode program *.js ditunjukkan pada Tabel 5.8.

Tabel 5. 8 Implementasi Assets pada Kode Program *.js

Folder	Nama	Nama File Program
assets/www/ js	Index Kategori Wisata Kategori Kuliner Daftar Kategori Wisata Daftar Kategori Kuliner Detail Wisata Detail Kuliner Map Wisata Map Kuliner Navigasi Wisata Navigasi Kuliner	app.js
assets/ www/js	Wisata Terdekat Map	map.js jquery.ui.map.js jquery.ui.map.extensions.js jquery.ui.map.service.js

5.1.4.4 Implementasi Assets pada Kode Program *.php

Fileassets dengan *extension* *.php berfungsi sebagai penghubung antara *server* dengan antarmuka aplikasi dalam proses pertukaran data. Implementasi *assets* pada kode program *.js ditunjukkan pada Tabel 5.9.

Tabel 5. 9 Implementasi Assets pada Kode Program *.php

Folder	Nama	Nama File Program
assets/ www/php	Index Daftar Kategori Wisata Daftar Kategori Kuliner Detail Wisata Detail Kuliner	model_wisata.php model_kategori_wisata.php model_kategori_kuliner.php model_detail_wisata.php model_detail_kuliner.php

assets/ www/php	Wisata Terdekat	model_get_location.php
-----------------	-----------------	------------------------

5.1.5 Implementasi Kode Program

Kode program terdiri dari kumpulan fungsi yang dapat menjalankan proses tertentu sesuai dengan kebutuhan dari aplikasi. Selain itu kode program juga berfungsi sebagai penghubung antara *server* dengan antarmuka aplikasi dalam proses pertukaran data. Di bawah ini ditampilkan implementasi kode program dari beberapa fungsi utama aplikasi.

5.1.5.1 Implementasi Kode Program Mencari Tempat Wisata

Kode program mencari tempat wisata dijalankan ketika pengguna memilih salah satu menu dari bar navigasi kategori wisata. Proses mencari tempat wisata diawali dengan pengguna memilih kategori wisata yang diinginkan. Kemudian sistem menampilkan daftar wisata yang dipilih pengguna (misalnya dalam kategori wisata tersebut pengguna memilih kategori pantai, maka sistem akan menampilkan daftar wisata pantai saja). Implementasi kode program mencari tempat wisata ditampilkan pada Gambar 5.5.

```

1  $(document).on("pageinit", "#list-wisata-kategori",
2  function(event, data){
3      var link = window.location.href;
4      var kategori = link.split("=")[1];
5      kategori = kategori.replace("%20", " ");
6      kategori = kategori.replace("%20", " ");
7      $("#div#wisata_nama").append(" "+kategori);
8      localStorage.setItem("kategori", kategori);
9      $.ajax({
10         url: urlRequest+'model_kategori_wisata.php',
11         type: 'POST',
12         async: 'true',
13         dataType: 'json',
14         data: {
15             cat:kategori
16         },
17         success: function (result) {
18             var cek = JSON.stringify(result);
19             if(cek=="No rows found"){
20                 var output = 'Kosong';
21                 $('#wisata-kategori').html(output);
22             }
23             else{
24                 var output = '';
25                 $.each(result, function(i, f){
26                     var gambar = f.gambar;
27                     var igambar = gambar.split(";");
28                     output += '<li data-
29 icon="false"><a href="detail-wisata.html"
30 onclick="set_wisata_id(\''+f.wisata_id+\'')"' class="detail"
31 id=\''+ f.wisata_id + \' " data-ajax="false"><a
29 href="detail-kuliner.html"                class="detail"
30 onclick="set_kuliner_id(\''+f.kuliner_id+\'\'")    id=\'\'    +
31 f.kuliner_id + \'\' data-ajax="false" data-rel="external"><img
32 src=\'\'    +igambar[0]+    \'\'>    <h2>\'    +f.kuliner_nama+    \'</h2>\'
33 +f.kuliner_lokasi+ \' </a></li>\'
34     });
35     $('#wisata-kuliner').html(output);
36     $('#wisata-kuliner').listview('refresh');
37 }
38 }
39 });
40
41 $(document).on("click", ".detail", function(){
42     var id = $(this).attr("id");
43     localStorage.setItem("id", id);
44 });
45 });

```

Gambar 5. 6 Kode Program Menampilkan Halaman *Detail*

5.1.5.3 Implementasi Kode Program Melihat Daftar Wisata Terdekat

Kode program melihat daftar wisata terdekat dilakukan dengan pengguna memilih tombol wisata terdekat (*Nearby*) di detail wisata atau kuliner pada aplikasi. Proses melihat daftar wisata terdekat diawali dengan pengguna memilih tempat wisata atau kuliner pada aplikasi. Kemudian pada halaman detail, pengguna memilih fitur wisata terdekat (*Nearby*). Sistem menampilkan daftar wisata terdekat berdasarkan tempat wisata yang dipilih oleh pengguna pada aplikasi. Implementasi kode program melihat daftar wisata terdekat ditampilkan pada Gambar 5.7.

```

1  function get_nearby(latitude, longitude) {
2      $.ajax({
3          type : "POST",
4          dataType: "json",
5          async: 'true',
6          url: urlRequest+'model_get_location.php',
7          data : $.param({
8              latitude
9 sessionStorage.getItem('session_lat'),
10             longitude
11 sessionStorage.getItem('session_longi'),
12             page
13 sessionStorage.getItem('session_page')
14         }),
15         success : function(result){
16             var cek = JSON.stringify(result);
17             var output = '';

```

```

18     $.each(result, function(i, f){
19         output += '<li>';
20         output += ' <a href="detail-wisata.html"
21 class="detail" id="' + f.wisata_id + '"
22 onclick="set_wisata_id(\''+f.wisata_id+'\')" style="height:
23 40px; border: 0px; padding: 2px 0px 0px 2px; border-radius:
24 0px" data-ajax="false" rel="external">';
25         output += '<strong style="font-
26 size: 12px">'+f.wisata_kategori+':
27 '+f.wisata_nama+ '</strong><br>';
28         output += '<strong style="font-
29 size: 12px; font-weight: normal">+/- '+f.distance+'
30 km</strong></a>';
31         output += '<div style="height:
32 1px; background: #dadada"></div>';
33         output += '</li>';
34     });
35     $('#list-nearby').html(output);
36     $('#list-nearby').listview('refresh');
37 }
38 });
39 }

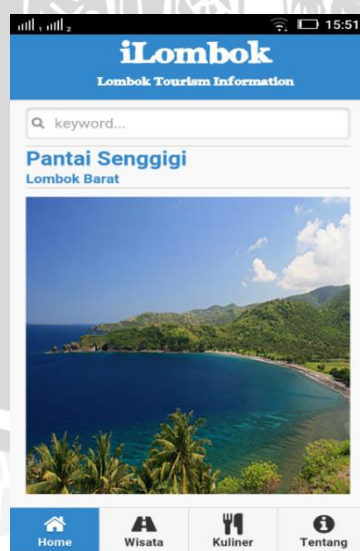
```

Gambar 5. 7 Kode Program Melihat Daftar Wisata Terdekat

5.1.6 Implementasi Antarmuka

Implementasi antarmuka dibuat berdasarkan hasil perancangan antarmuka yang telah dilakukan. Berikut ini merupakan implementasi beberapa halaman utama dari aplikasi.

5.1.6.1 Antarmuka Halaman Beranda



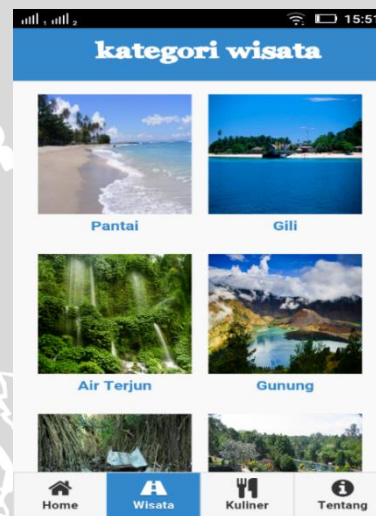
Gambar 5. 8 Antarmuka Halaman Beranda

Gambar 5.8 merupakan implementasi antarmuka halaman *Beranda*. Halaman Beranda merupakan halaman utama aplikasi ini. Halaman Beranda menampilkan

halaman beranda yang dimana *header* yang berisi logo aplikasi, sedangkan footer berisi tombol menuju halaman beranda, kategori wisata, kategori kuliner dan tentang aplikasi ini.

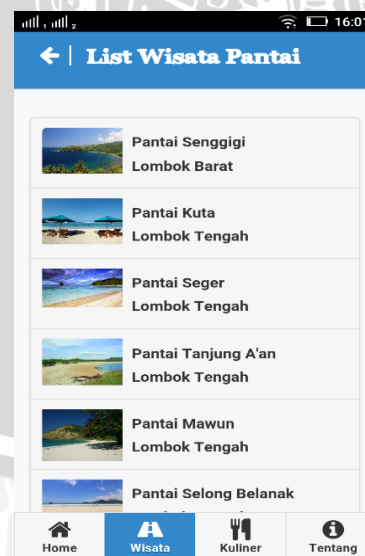
5.1.6.2 Antarmuka Halaman Kategori Wisata

Halaman kategori wisata ditampilkan ketika pengguna ingin melihat daftar wisata berdasarkan kategori. Halaman kategori wisata menampilkan *header* yang berisi tulisan kategori wisata, footer berisi tombol menuju halaman beranda, kategori wisata, kategori kuliner dan tentang aplikasi ini. Implementasi antarmuka halaman *kategori wisata* ditunjukkan pada Gambar 5.9 di bawah ini.



Gambar 5. 9 Antarmuka Halaman Kategori Wisata

5.1.6.3 Antarmuka Halaman Daftar Wisata



Gambar 5. 10 Antarmuka Halaman Daftar Wisata

Gambar 5.10 merupakan implementasi antarmuka halaman *daftar wisata* ditunjukkan pada Halaman daftar wisata ditampilkan ketika pengguna memilih

salah satu kategori yang ada pada kategori wisata. Halaman daftar menampilkan *header* yang berisi tulisan “Daftar” yang dilengkapi dengan nama kategori wisata dan footer berisi tombol menuju halaman beranda, kategori wisata, kategori kuliner dan tentang aplikasi ini. Halaman daftar ini juga menampilkan konten daftar wisata yang meliputi foto, nama tempat wisata, dan lokasi tempat wisata.

5.1.6.4 Antarmuka Halaman Detail

Halaman detail ditampilkan ketika pengguna telah memilih salah satu dari daftar wisata atau kuliner yang terdapat di dalam kategori, atau yang terdapat pada beranda. Halaman detail menampilkan *header* yang berisi tulisan yaitu nama tempat yang dipilih dan footer berisi tombol menuju halaman beranda, kategori wisata, kategori kuliner dan tentang aplikasi ini. Halaman detail wisata ini juga menampilkan konten detail wisata yang meliputi foto, deskripsi tempat wisata, informasi tempat wisata, penunjuk arah, peta, dan wisata terdekat. Implementasi antarmuka halaman detail ditunjukkan pada Gambar 5.11 di bawah ini.



Gambar 5. 11 Antarmuka Halaman Detail

5.1.6.5 Antarmuka Halaman Tentang

Halaman tentang ditampilkan ketika pengguna ingin melihat deskripsi aplikasi dan profil pembuat aplikasi. Halaman tentang menampilkan *header* yang berisi tulisan yaitu “tentang iLombok” dan footer berisi tombol menuju halaman beranda, kategori wisata, kategori kuliner dan tentang aplikasi ini. Halaman daftar ini juga menampilkan konten berupa deskripsi mengenai aplikasi ini. Implementasi antarmuka halaman tentang ditunjukkan pada Gambar 5.12 di bawah ini.



Gambar 5. 12 Antarmuka Halaman Tentang

5.1.7 Implementasi Geolocation

Pada halaman detail terdapat beberapa pilihan fitur menggunakan *Google Map API* untuk menampilkan *geolocation*. *Geolocation* memungkinkan aplikasi yang dibuat dapat menampilkan informasi lokasi dari perangkat bergerak yang dimana telah terkoneksi *internet dan GPS*. Kode *program geolocation* akan diimplementasikan pada *file java Script* yang nantinya akan diakses oleh halaman pada fitur-fitur yang terdapat pada halaman detail aplikasi. Fitur-fitur tersebut akan dijalankan ketika pengguna memilih fitur tersebut. Gambar 5.16 merupakan implementasi *geolocation* pada kode program.

```

1  function locSuccess(position) {
2      initialize(position.coords.latitude,
3      position.coords.longitude);
4  }
5
6  function locError(error) {
7      alert("Your location undetected");
8  }
9
10 function initialize(lat, lon) {
11     currentPosition = new google.maps.LatLng(lat, lon);
12
13     if(sessionStorage.getItem('session_page') == 'view
14 map'){
15         destPosition = new google.maps.LatLng(destLatitude,
16 destLongitude);
17
18         map = new
19 google.maps.Map(document.getElementById('map'), {

```

```

20         zoom: 15,
21         center: destPosition,
22         mapTypeId: google.maps.MapTypeId.ROADMAP
23     });
24
25     var destPositionMarker = new
26     google.maps.Marker({
27         position: destPosition,
28         map: map,
29         title: destName
30     });
31 }
32
33     else if(sessionStorage.getItem('session_page') == 'get
34     nearby'){
35         get_nearby(lat, lon);
36         //get_nearby(Ltt, Lng);
37
38         var addr;
39
40         var reverseGeocoder = new
41         google.maps.Geocoder();
42         reverseGeocoder.geocode({'latLng':
43         currentPosition}, function(results, status) {
44             if (status == google.maps.GeocoderStatus.OK)
45             {
46                 if (results[0]) {
47
48                     $('#current').text(results[0].formatted_address);
49                     addr =
50                     results[0].formatted_address;
51                     //
52                     navigator.notification.alert('Address : ' +
53                     results[0].formatted_address + ', ' + 'Type : ' +
54                     results[0].types);
55                     $('#address').val(addr);
56                 }
57                 else {
58                     //
59                     navigator.notification.alert('Unable to detect your
60                     address. ');
61                 }
62                 } else {
63                     // navigator.notification.alert('Unable
64                     to detect your address. ');
65                 }
66             });
67
68             $('#latitude').val(lat);
69             $('#longitude').val(lon);
70             // $('#latitude').val(Ltt);

```

```

71 // $('#longitude').val(Lng);
72
73 $.ajax({
74     type : "POST",
75     dataType : "json",
76     async: 'true',
77     url: urlRequest+'model_get_location.php',
78     data : $.param({
79         latitude :
80 sessionStorage.getItem('session_lat'),
81         longitude :
82 sessionStorage.getItem('session_longi')
83     }),
84     success : function(data){
85
86         // var cek = JSON.stringify(data);
87         // alert(cek);
88
89         $.each( data, function(i, marker) {
90             roomMarker = new
91             google.maps.Marker({
92                 position: new
93                 google.maps.LatLng(marker.latitude, marker.longitude),
94                 //position: new
95                 google.maps.LatLng(marker.Ltt, marker.Lng),
96                 map: map
97             });
98         });
99     }
100 });
101
102
103     else if(sessionStorage.getItem('session_page') == 'view
104     navigation'){
105         $(document).on('pageinit',function() {
106             $('#map').gmap({
107                 'center': start,
108                 'zoom': endlocation.zoom,
109                 'disableDefaultUI':true,
110                 'callback': function() {
111                     themap = this;
112                 }
113             });
114         });
115
116         destPosition = new google.maps.LatLng(destLatitude,
117         destLongitude);
118
119         directionsDisplay = new
120         google.maps.DirectionsRenderer();
121

```

```

122         directionsService = new
123         google.maps.DirectionsService();
124
125         map = new
126         google.maps.Map(document.getElementById('map'), {
127             zoom: 15,
128             center: destPosition,
129             mapTypeId: google.maps.MapTypeId.ROADMAP
130         });
131
132         directionsDisplay.setMap(map);
133
134         //getRoute(lat, lon);
135
136         var currentPositionMarker = new
137         google.maps.Marker({
138             position: currentPosition,
139             map: map,
140             title: 'You are here!',
141             //icon: 'img/pin/pin-user.png'
142         });
143
144         var infowindow = new google.maps.InfoWindow();
145
146         google.maps.event.addListener(currentPositionMarker,
147         'click', function() {
148             // infowindow.setContent("Current position:
149         latitude: " + lat + " longitude: " + lon);
150             infowindow.setContent('<div class =
151         "MarkerPopUp" style= "width: 80px;height: 20px"
152         ><center>You are here</center></div>');
153             infowindow.open(map, currentPositionMarker);
154         });
155
156         var destPositionMarker = new google.maps.Marker({
157             position: destPosition,
158             map: map,
159             title: destName,
160             //icon: 'img/pin/pin-location.png'
161         });
162
163         var infowindow = new google.maps.InfoWindow();
164
165         google.maps.event.addListener(destPositionMarker,
166         'click', function() {
167             // infowindow.setContent("Current position:
168         latitude: " + lat + " longitude: " + lon);
169             infowindow.setContent('<div><center
170         style="margin: 0px !important; padding: 0px
171         !important">'+destName+'</center></div>');
172             infowindow.open(map, destPositionMarker);

```

```

173     });
174
175     calculateRoute(lat,lon);
176 }
177 }
178
179 function calculateRoute(lat,lon) {
180     if(lat){
181         // destLatitude =
182         sessionStorage.getItem('session_latitude');
183         // destLongitude =
184         sessionStorage.getItem('session_longitude');
185         var targetDestination = new
186         google.maps.LatLng(destLatitude , destLongitude);
187     }else{
188         var targetDestination = $("#textinput-
189 hide").val();
190     }
191
192     if (currentPosition && currentPosition != '' &&
193     targetDestination && targetDestination != '') {
194         var request = {
195             origin:currentPosition,
196             destination:targetDestination,
197             travelMode:
198             google.maps.DirectionsTravelMode["DRIVING"]
199         };
200
201         directionsService.route(request, function(response,
202 status) {
203             if (status == google.maps.DirectionsStatus.OK)
204             {
205
206                 directionsDisplay.setPanel(document.getElementById("directi
207 ons"));
208
209                 directionsDisplay.setDirections(response);
210                 $("#results").show();
211             }
212             else {
213                 $("#results").hide();
214             }
215         });
216     }
217     else {
218         $("#results").hide();
219     }
220 }

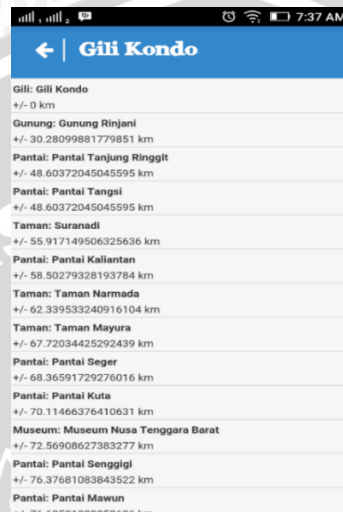
```

Gambar 5. 13 Implementasi *Geolocation* Pada Kode Program

Kode pada baris 1-8 merupakan pencarian koordinat pada perangkat bergerak *smartphone*. Jika sistem berhasil menemukan nilai koordinat dari perangkat

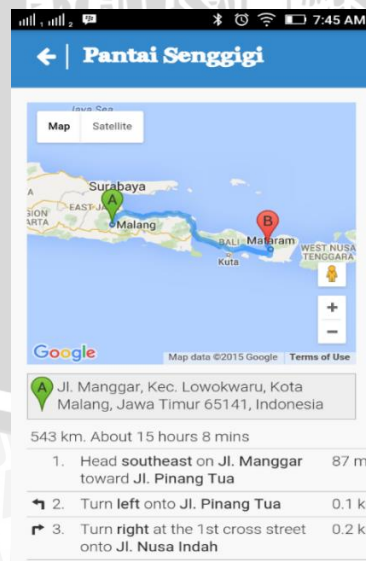
bergerak *smartphone*, maka akan menjalankan kode pada baris 1-4. Sedangkan apabila tidak mendapatkan nilai koordinat, maka akan menjalankan kode pada baris 6-8.

Kode baris 53-101 merupakan fungsi untuk mendapatkan daftar lokasi terdekat yang dipanggil ketika pengguna memilih fitur *nearby* (*wisata terdekat*). Implementasi *geolocation* pada fitur Wisata Terdekat ditunjukkan pada Gambar 5.14.



Gambar 5. 14 Implementasi *Geolocation* pada Fitur Wisata Terdekat

Kode baris 105-176 merupakan fungsi untuk menggunakan fitur *map* yang digunakan ketika fitur navigasi dibuka. Pada baris 178 dipanggil fungsi *calculateRoute* yang berfungsi untuk memberikan deskripsi rute menuju lokasi ketika fitur navigasi dijalankan. Implementasi *geolocation* pada fitur Navigasi ditunjukkan pada Gambar 5.17.



Gambar 5. 15 Implementasi *Geolocation* pada Fitur Navigasi

BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dilakukan untuk mengetahui apakah proses implementasi yang dilakukan telah mencapai tujuan dan fungsi dari dibuatnya aplikasi, serta dapat memenuhi kebutuhan fungsional dan non-fungsional yang telah didefinisikan sebelumnya. Proses pengujian dilakukan menggunakan 2 jenis pengujian, yaitu pengujian validasi dan *usability*. Pengujian validasi dilakukan untuk menguji aplikasi dari segi kebutuhan fungsional. Sedangkan pengujian *usability* untuk menguji aplikasi dari segi kebutuhan non-fungsional.

6.1 Pengujian Validasi

Pengujian validasi digunakan untuk menguji apakah sistem yang dibangun sudah sesuai dengan yang dibutuhkan. Daftar kebutuhan yang telah dirumuskan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan teknik pengujian *black box*. Pada pengujian validasi hanya dilakukan pengujian dengan kondisi yang valid atau tidak valid.

6.1.1 Kasus Uji

Pada skripsi ini dilakukan pengujian validasi terhadap aplikasi perangkat bergerak sistem informasi pariwisata pulau Lombok. Untuk mengetahui kesesuaian antara daftar kebutuhan dan kinerja sistem, maka pada setiap kebutuhan sistem dilakukan proses pengujian dengan kasus uji. Kasus uji yang dibuat untuk aplikasi perangkat bergerak sistem informasi pariwisata pulau Lombok memiliki nomor kasus uji dengan format UJI_A_XXX. Dimana XXX yang terdapat pada nomor kasus uji tersebut dimulai dari 001, 002, 003 dan seterusnya. Proses pengujian dengan kasus uji masing-masing yang akan ditunjukkan pada tabel-tabel di bawah ini.

6.1.1.1 Kasus Uji Melihat Halaman Beranda

Kasus uji melihat halaman beranda ditunjukkan pada Tabel 6.1 di bawah ini.

Tabel 6. 1 Kasus Uji Melihat Halaman Beranda

Nomor Kasus Uji	UJI_A_001
Nama Kasus Uji	Melihat Halaman Beranda
Objek Uji	SRS_A_001
Tujuan Pengujian	Memastikan aplikasi dapat memenuhi kebutuhan fungsional menampilkan halaman beranda.
Data Masukan	-
Prosedur Pengujian	1. Membuka aplikasi 2. Masuk halaman Beranda

Hasil yang Diharapkan	Data pada server dapat dimuat dan ditampilkan pada halaman beranda.
-----------------------	---

6.1.1.2 Kasus Uji Mencari Tempat Wisata

Kasus uji mencari tempat wisata ditunjukkan pada Tabel 6.2 di bawah ini.

Tabel 6. 2 Kasus Uji Mencari Tempat Wisata

Nomor Kasus Uji	UJI_A_002
Nama Kasus Uji	Mencari Tempat Wisata
Objek Uji	SRS_A_002
Tujuan Pengujian	Memastikan aplikasi dapat memenuhi kebutuhan fungsional pencarian tempat wisata berdasarkan kategori.
Data Masukan	-
Prosedur Pengujian	1. Membuka aplikasi 2. Klik tab wisata 3. Melihat kategori wisata 4. Memilih kategori wisata 5. Menampilkan daftar tempat wisata berdasarkan kategori yang telah dipilih
Hasil yang Diharapkan	Sistem menampilkan daftar tempat wisata dari menu kategori wisata yang dipilih oleh pengguna.

6.1.1.3 Kasus Uji Mencari Tempat Kuliner

Kasus uji mencari tempat kuliner ditunjukkan pada Tabel 6.3 di bawah ini.

Tabel 6. 3 Kasus Uji Mencari Tempat Kuliner

Nomor Kasus Uji	UJI_A_003
Nama Kasus Uji	Mencari Tempat Kuliner
Objek Uji	SRS_A_003
Tujuan Pengujian	Memastikan aplikasi dapat memenuhi kebutuhan fungsional pencarian tempat kuliner berdasarkan kategori.
Data Masukan	-
Prosedur Pengujian	1. Membuka aplikasi 2. Klik tab kuliner

	3. Melihat kategori kuliner 4. Memilih kategori kuliner 5. Menampilkan daftar tempat kuliner berdasarkan kategori yang telah dipilih
Hasil yang Diharapkan	Sistem menampilkan daftar tempat kuliner dari menu kategori kuliner yang dipilih oleh pengguna.

6.1.1.4 Kasus Uji Melihat Rute Lokasi Tujuan

Kasus uji melihat rute lokasi tujuan ditunjukkan pada Tabel 6.4 di bawah ini.

Tabel 6. 4 Kasus Uji Melihat Rute Lokasi Tujuan

Nomor Kasus Uji	UJI_A_004
Nama Kasus Uji	Melihat Rute Lokasi Tujuan
Objek Uji	SRS_A_004
Tujuan Pengujian	Memastikan aplikasi dapat memenuhi kebutuhan fungsional untuk melihat rute menuju lokasi tujuan.
Data Masukan	-
Prosedur Pengujian	1. Membuka aplikasi 2. Mencari tempat wisata atau kuliner 3. Memilih tempat wisata atau kuliner 4. Menampilkan detail tempat wisata 5. Memilih fitur navigasi
Hasil yang Diharapkan	Sistem menampilkan peta yang dimana sebelumnya dilakukan pencarian letak pengguna berada saat itu dan dihubungkan ke lokasi tempat wisata yang telah dipilih. Kemudian sitem memberikan deskripsi rute untuk menuju ke lokasi tujuan.

6.1.1.5 Kasus Uji Melihat Peta Lokasi

Kasus uji melihat peta lokasi ditunjukkan pada Tabel 6.5 di bawah ini.

Tabel 6. 5 Kasus Uji Melihat Peta Lokasi

Nomor Kasus Uji	UJI_A_005
Nama Kasus Uji	Melihat Peta Lokasi
Objek Uji	SRS_A_005
Tujuan Pengujian	Memastikan aplikasi dapat memenuhi kebutuhan fungsional untuk melihat peta lokasi.

Data Masukan	-
Prosedur Pengujian	1. Membuka aplikasi 2. Mencari tempat wisata atau kuliner 3. Memilih tempat wisata atau kuliner 4. Menampilkan detail tempat wisata 5. Memilih fitur map
Hasil yang Diharapkan	Sistem menampilkan peta lokasi tempat wisata atau kuliner yang telah dipilih.

6.1.1.6 Kasus Uji Melihat Daftar Wisata Terdekat

Kasus uji melihat daftar wisata terdekat ditunjukkan pada Tabel 6.6 di bawah ini.

Tabel 6. 6 Kasus Uji Melihat Daftar Wisata Terdekat

Nomor Kasus Uji	UJI_A_006
Nama Kasus Uji	Melihat Daftar Wisata Terdekat
Objek Uji	SRS_A_006
Tujuan Pengujian	Memastikan aplikasi dapat memenuhi kebutuhan fungsional untuk memberikan daftar wisata terdekat berdasarkan jarak tempat wisata yang dipilih.
Data Masukan	-
Prosedur Pengujian	1. Membuka aplikasi 2. Mencari tempat wisata atau kuliner 3. Memilih tempat wisata atau kuliner 4. Menampilkan detail tempat wisata 5. Memilih fitur <i>Nearby</i>
Hasil yang Diharapkan	Sistem menampilkan daftar wisata terdekat dari tempat wisata yang dilihat oleh pengguna.

6.1.2 Hasil Pengujian Validasi

Setelah kasus uji dibuat, tahap selanjutnya dilakukan pengujian sesuai dengan prosedur pengujian yang tertera di setiap kasus uji. Hasil dari pengujian tersebut kemudian disajikan dalam bentuk tabel. Hasil pengujian dikatakan valid apabila hasil yang didapatkan sama dengan hasil yang diharapkan. Tabel 6.7 menyajikan data hasil pengujian validasi terhadap aplikasi pariwisata pulau Lombok.

Tabel 6. 7 Hasil Pengujian Validasi Aplikasi Pariwisata Pulau Lombok

No.	Nomor Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status Validitas
1.	UJI_A_001	Data pada server dapat dimuat dan ditampilkan pada halaman beranda.	Data pada server dapat dimuat dan ditampilkan pada halaman beranda.	Valid
2.	UJI_A_002	Sistem menampilkan daftar tempat wisata dari menu kategori wisata yang dipilih oleh pengguna.	Sistem menampilkan daftar tempat wisata dari menu kategori wisata yang dipilih oleh pengguna.	Valid
3.	UJI_A_003	Sistem menampilkan daftar tempat kuliner dari menu kategori kuliner yang dipilih oleh pengguna.	Sistem menampilkan daftar tempat kuliner dari menu kategori kuliner yang dipilih oleh pengguna.	Valid
4.	UJI_A_004	Sistem menampilkan peta yang dimana sebelumnya dilakukan pencarian letak pengguna berada saat itu dan dihubungkan ke lokasi tempat wisata yang telah dipilih. Kemudian sitem memberikan deskripsi rute untuk menuju ke lokasi tujuan.	Sistem menampilkan peta yang dimana sebelumnya dilakukan pencarian letak pengguna berada saat itu dan dihubungkan ke lokasi tempat wisata yang telah dipilih. Kemudian sitem memberikan deskripsi rute untuk menuju ke lokasi tujuan.	Valid
5.	UJI_A_005	Sistem menampilkan peta lokasi tempat wisata atau kuliner yang telah dipilih.	Sistem menampilkan peta lokasi tempat wisata atau kuliner yang telah dipilih.	Valid
6.	UJI_A_006	Sistem menampilkan daftar wisata	Sistem menampilkan daftar wisata	Valid

		terdekat dari tempat wisata yang dilihat oleh pengguna.	terdekat dari tempat wisata yang dilihat oleh pengguna.	
--	--	---	---	--

6.2 Pengujian Usability

Pengujian *usability* dilakukan untuk mengukur apakah sistem yang sudah dibangun sesuai dengan *usability* yang dimana merupakan kebutuhan non-fungsional dari sistem. Proses pengujian *usability* ini menggunakan model *SUS* (*System Usability Scale*) yang berupa kuesioner untuk mendapatkan evaluasi dari pengguna aplikasi. *SUS* terdiri dari 10 pertanyaan yang memberikan suatu pandangan global mengenai penilaian yang subjektif tentang *usability*. Daftar pernyataan kuesioner disajikan pada Tabel 6.8.

Tabel 6. 8 Daftar Pernyataan Kuesioner

1.	Saya berpikir bahwa saya ingin menggunakan aplikasi (sistem) ini dengan sering	1	2	3	4	5
2.	Saya menemukan sistem rumit yang tidak perlu	1	2	3	4	5
3.	Saya pikir aplikasi (sistem) ini mudah digunakan	1	2	3	4	5
4.	Saya berpikir bahwa saya akan membutuhkan dukungan dari teknisi untuk dapat menggunakan aplikasi (sistem) ini	1	2	3	4	5
5.	Saya menemukan berbagai fungsi dalam aplikasi (sistem) ini diintegrasikan dengan baik	1	2	3	4	5
6.	Saya pikir ada terlalu banyak inkonsistensi dalam aplikasi (sistem) ini	1	2	3	4	5
7.	Saya membayangkan bahwa kebanyakan orang akan belajar untuk menggunakan aplikasi (sistem) ini dengan sangat cepat	1	2	3	4	5
8.	Saya menemukan aplikasi (sistem) yang sangat rumit untuk digunakan	1	2	3	4	5
9.	Saya merasa sangat yakin menggunakan aplikasi (sistem) ini	1	2	3	4	5
10.	Saya perlu belajar banyak hal sebelum saya bisa pergi dengan aplikasi (sistem) ini	1	2	3	4	5

Sebelum kuesioner disebarakan kepada responden, terlebih dahulu dilakukan pengukuran reliabilitas terhadap poin-poin pernyataan yang disertakan dalam kuesioner. Proses pengukuran dilakukan dengan menyebarkan kuesioner yang digunakan sebagai sampel. Jumlah yang disarankan dalam melakukan pengumpulan data secara kuantitatif adalah 20 responden demi mendapatkan hasil akhir yang baik (Ersa, A. M., 2015). Kuesioner sampel yang digunakan terdiri dari 5 poin pernyataan dengan skala *likert 5 level of agreement*. Penggunaan skala *likert 5 level of agreement* mewakili pendapat responden terkait poin-poin pernyataan yang disertakan sebagai parameter pengujian. Penjelasan skala likert *5 level of agreement* ditunjukkan pada Tabel 6.9 di bawah ini.

Tabel 6. 9 Penjelasan Skala *Likert 5 Level of Agreement*

Level of Agreement	Keterangan
1	Sangat tidak setuju, yang berarti aplikasi jauh diluar ekspektasi dari responden terkait pernyataan yang diberikan.
2	Tidak setuju, yang berarti pernyataan yang diberikan tidak sesuai dengan pendapat responden.
3	Netral, yang berarti responden tidak memiliki pendapat atau pandangan terkait pernyataan yang diberikan.
4	Setuju, yang berarti pernyataan yang diberikan sesuai dengan pendapat dari responden.
5	Sangat setuju, yang berarti aplikasi memberikan sesuatu yang melebihi ekspektasi dari responden terkait pernyataan yang diberikan.

6.2.1 Perhitungan Kuisoner SUS

Cara untuk menghitung hasil dari responden dilakukan dengan menggunakan rumus perhitungan SUS untuk mendapatkan skor kontribusi. Untuk mendapatkan skor kontribusi tersebut, nilai yang telah diberikan responden pada tiap pernyataan dihitung. Perhitungannya dengan cara pernyataan nomor ganjil dikurangi 1. Sedangkan pernyataan nomor genap, didapatkan dengan mengurangi 5 dengan nilai jawaban responden. Berikut adalah hasil perhitungan kuisoner SUS ditunjukkan pada Tabel 6.10.

Tabel 6. 10 Perhitungan Kuesioner SUS

NO	Pernyataan	Responden																				SKOR SUS	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		JUMLAH
1	Saya berpikir bahwa saya ingin menggunakan aplikasi (sistem) ini dengan sering	3	3	4	4	3	3	3	3	3	3	4	2	3	4	4	4	4	4	4	4	69	172.5
2	Saya menemukan sistem rumit yang tidak perlu	4	4	4	4	4	4	4	4	2	3	4	3	4	3	1	4	3	4	3	4	70	175
3	Saya pikir aplikasi (sistem) ini mudah digunakan	3	4	4	4	4	4	4	4	3	3	3	3	4	3	3	4	3	3	3	4	70	175
4	Saya berpikir bahwa saya akan membutuhkan dukungan dari teknisi untuk dapat menggunakan sistem ini	3	4	4	4	3	4	3	4	3	3	3	0	4	4	1	3	3	2	3	3	61	152.5
5	Saya menemukan berbagai fungsi dalam sistem ini diintegrasikan dengan baik	4	4	3	4	3	3	4	4	3	3	4	1	3	4	4	4	4	4	3	3	69	172.5
6	Saya pikir ada terlalu banyak inkonsistensi dalam sistem ini	3	4	4	4	4	4	4	4	3	3	4	2	3	3	0	4	3	4	2	3	65	162.5
7	Saya membayangkan bahwa kebanyakan orang akan belajar untuk menggunakan sistem ini dengan sangat cepat	3	4	4	4	4	4	3	3	3	3	4	3	3	3	4	4	3	3	4	4	70	175
8	Saya menemukan sistem yang sangat rumit untuk digunakan	3	4	4	4	3	4	4	4	3	3	3	4	4	4	1	4	3	3	3	4	69	172.5
9	Saya merasa sangat yakin menggunakan sistem	3	3	4	4	4	3	4	4	3	3	4	3	4	4	3	4	3	4	4	4	72	180
10	Saya perlu belajar banyak hal sebelum saya bisa pergi dengan sistem ini	2	4	4	4	4	4	4	4	3	3	4	0	4	3	0	4	3	4	4	3	65	162.5
TOTAL SUS																						1700	
RATA-RATA SKOR SUS																						85	

6.2.2 Hasil Kuisisioner SUS

Untuk mendapatkan skor SUS, nilai pada kolom jumlah dikalikan dengan 2,5. Total SUS merupakan nilai total keseluruhan nilai SUS. Total SUS tersebut kemudian dibagi dengan banyaknya responden yaitu 20 orang, sehingga didapatkan rata-rata skor SUS sebesar 85. Nilai tersebut termasuk dalam kategori *Acceptable* karena berada diantara nilai 70 dan 100 (Bangor, A., Kortum, P. & Miller, J., 2009).

6.3 Analisis Hasil Pengujian

Analisis hasil pengujian dilakukan untuk mendapatkan kesimpulan dari hasil pengujian aplikasi pariwisata pulau lombok. Proses analisis mengacu pada hasil pengujian yang telah didapatkan. Analisis dilakukan untuk setiap pengujian yang telah dilakukan meliputi analisis hasil pengujian validasi dan analisis hasil pengujian *usability*.

6.3.1 Analisis Pengujian Hasil Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat kesesuaian fungsi yang diuji dengan hasil perancangan sistem. Berdasarkan hal tersebut, dapat disimpulkan bahwa aplikasi perangkat bergerak informasi pariwisata pulau Lombok telah memenuhi seluruh kebutuhan fungsional yang telah dirancang pada tahap analisis dan perancangan.

6.3.2 Analisis Pengujian Hasil *Usability*

Hasil dari pengujian *usability* mendapatkan rata-rata skor SUS 85. Dengan *Grade Scale "B"* dan *Adjective Rating "Excellent"* (Bangor, A., Kortum, P. & Miller, J., 2009). Hasil tersebut menunjukkan bahwa aplikasi perangkat bergerak informasi pariwisata pulau lombok mudah digunakan dan diterima oleh pengguna akhir. Dengan demikian, kebutuhan non fungsional telah terpenuhi.

BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Dari hasil pengamatan selama perancangan, implementasi dan proses pengujian, dapat diambil kesimpulan sebagai berikut:

1. Perancangan aplikasi sistem informasi pariwisata pulau Lombok berbasis perangkat bergerak telah dibuat sesuai dengan spesifikasi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan dengan pendekatan berorientasi terstruktur (*Structured Oriented Programming*) menggunakan DFD (*Data Flow Diagram*).
2. Implementasi aplikasi informasi pariwisata pulau Lombok telah memenuhi seluruh kebutuhan fungsional yang dirancang pada tahap analisis dan perancangan. Hal tersebut dapat dibuktikan dengan hasil pengujian validasi menggunakan metode *Black-Box Testing*.
3. Berdasarkan hasil pengujian *usability* menggunakan kuisioner SUS (*System Usability Scale*), mendapatkan rata-rata skor SUS 85. Hasil tersebut menunjukkan bahwa aplikasi perangkat bergerak informasi pariwisata pulau lombok berbasis mudah digunakan dan diterima oleh pengguna akhir.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan selanjutnya dalam aplikasi ini adalah:

1. Dilakukan pengembangan terhadap aplikasi ini agar bisa berjalan pada mode *offline*. Mode *offline* adalah kondisi dimana tidak adanya koneksi internet.

DAFTAR PUSTAKA

- Al- Rayes, K. A. et al., 2011. A Mobile Tourist Guide for Trip Planning. *IEEE MULTIDISCIPLINARY ENGINEERING EDUCATION MAGAZINE*, Volume 6, pp. 1-6.
- Bangor, A., Kortum, P. & Miller, J., 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3), p. 121.
- broari.com, 2014. *Pengenalan JQuery Mobile*. [Online]
Tersedia di: <http://www.broari.com/2014/05/pengenalan-jquery-mobile.html>
[Diakses pada 15 September 2015].
- Brooke, J., 2015. SUS - A quick and dirty usability scale. pp. 1-7.
- Ersa, A. M., 2015. Usability Evaluation Website E-Government Layanan Aspirasi Dan Pengaduan Online (Lapor!): Perbandingan Antara Existing Product Dan Development Product. p. 15.
- Hartono, T., 2014. *Lombok Tetap Jadi Favorit Wisatawan Asing*. [Online]
Tersedia di: <http://bisnis.tempo.co/read/news/2014/05/04/090575210/lombok-tetap-jadi-favorit-wisatawan-asing>
[Diakses pada 3 September 2014].
- Heriyanto, T., 2014. *Indonesia Masuk 5 Besar Negara Pengguna Smartphone*. [Online]
Tersedia di: <http://inet.detik.com/read/2014/02/03/171002/2485920/317/indonesia-masuk-5-besar-negara-pengguna-smartphone>
[Diakses pada 4 September 2014].
- LombokIndonesia.org, 2015. *Tempat Wisata Di Pulau Lombok Yang Harus Dikunjungi*. [Online]
Tersedia di: <http://id.lombokindonesia.org/tempat-wisata-di-pulau-lombok-yang-harus-dikunjungi/>
[Diakses pada 11 June 2015].
- Pincuran.com, 2012. *Pengertian dan Penjelasan Tentang Apa itu Phonegap?*. [Online]
Tersedia di: <http://pincuran.com/index.php/78-pemograman-android/90-pengertian-dan-penjelasan-apa-itu-phonegap>
[Diakses pada 20 September 2015].
- Putra, C. A., 2012. *Pengantar Google Maps API*. [Online]
Tersedia di: <http://www.candra.web.id/pengantar-google-maps-api/>
[Diakses pada 17 September 2015].

Setiabudi, D. H., Rostianingsih, S. & Tj, L. J., 2014. Sistem Informasi Geografis Dengan Fitur Peta Dan Rute Perjalanan Studi Kasus Di Kabupaten Malang. pp. 1-5.

Shahab, Q., 2012. *Pengujian Usability dan Contoh Kasusnya*. [Online] Tersedia di: <https://dailysocial.id/post/pengujian-usability-dan-contoh-kasusnya/> [Diakses pada 16 September 2015].

Sinsuw, A. & Najooan, X., 2013. Prototipe Aplikasi Sistem Informasi Akademik Pada Perangkat Android. *E-journal Teknik Elektro dan Komputer*, pp. 1-10.

updatekeren.com, 2012. *Pengertian Android, Apa Itu Android? Android adalah*. [Online] Tersedia di: <http://www.updatekeren.com/2012/07/pengertian-android-apa-itu-android.html> [Diakses pada 17 September 2015].

Yudira, F., 2015. *Perbedaan Aplikasi Native, Aplikasi Web, dan Aplikasi Hybrid*. [Online] Tersedia di: <http://blog.dika.web.id/2015/08/perbedaan-aplikasi-native-aplikasi-web-dan-aplikasi-hybrid.html> [Diakses pada 9 September 2015].

