**RANCANG BANGUN SISTEM INFORMASI GEOGRAFIS UNTUK**

**PEMETAAN LOKASI WISATA ALAM DENGAN MENGGUNAKAN**

**METODE PROTOTYPING BERBASIS MOBILE (STUDI KASUS:**

**PULAU LOMBOK)**

Tugas akhir  
untuk memenuhi sebagian persyaratan  
mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



**Oleh:**

**GILANG RAMDHANI PUTRA**

**F1D 016 031**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**2020**

# TUGAS AKHIR

# RANCANG BANGUN SISTEM INFORMASI GEOGRAFIS PEMETAAN LOKASI WISATA ALAM DENGAN MENGGUNAKAN METODE PROTOTYPING BERBASIS MOBILE ( STUDI KASUS : PULAU LOMBOK)

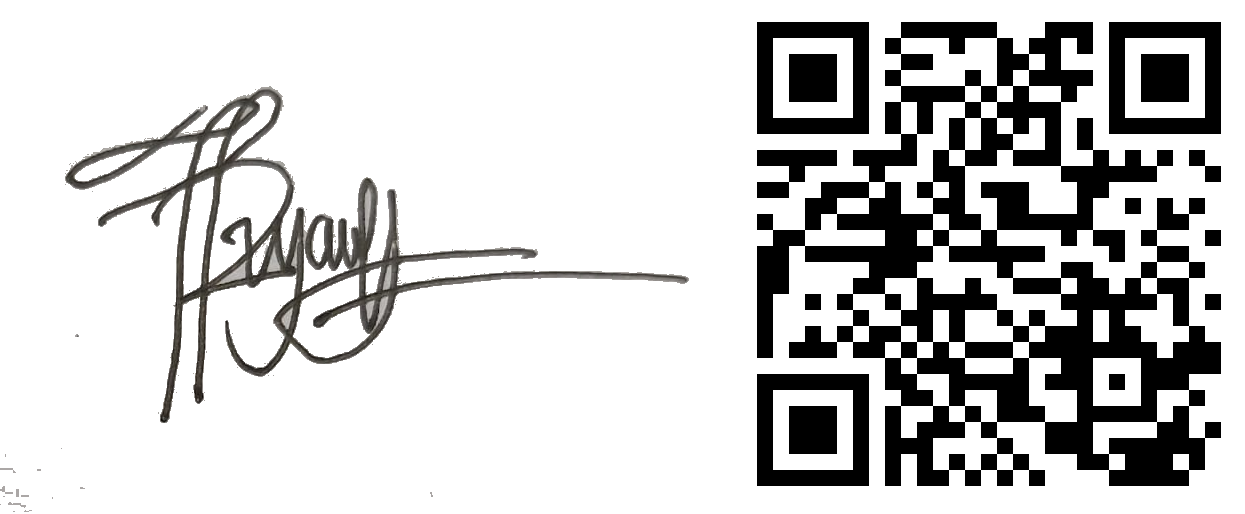
Oleh :

**GILANG RAMDHANI PUTRA**

F1D016031

Telah diperiksa oleh Tim Pembimbing :

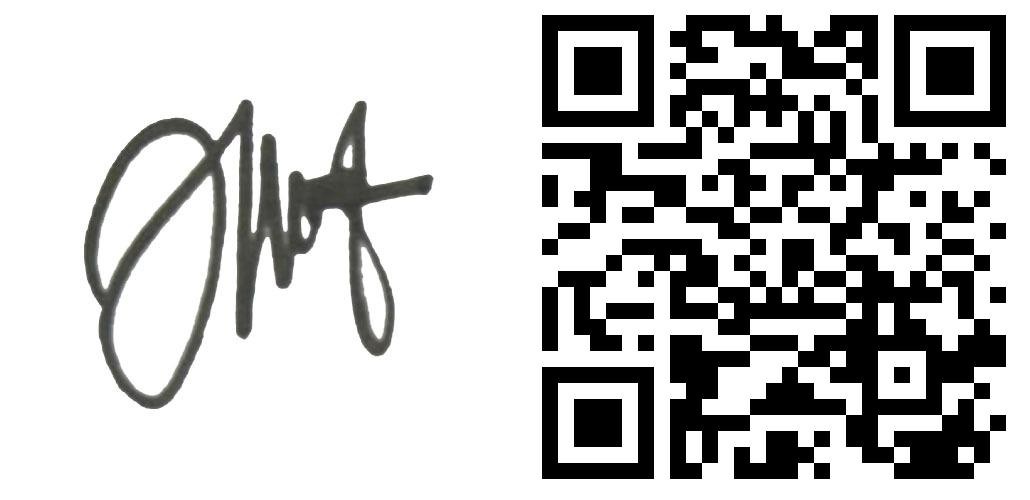
1. Pembimbing Utama



Tanggal: 23/07/2020

**Royana Afwani, ST., MT.**

**NIP. 198507072014042001**

1. Pembimbing Pendamping

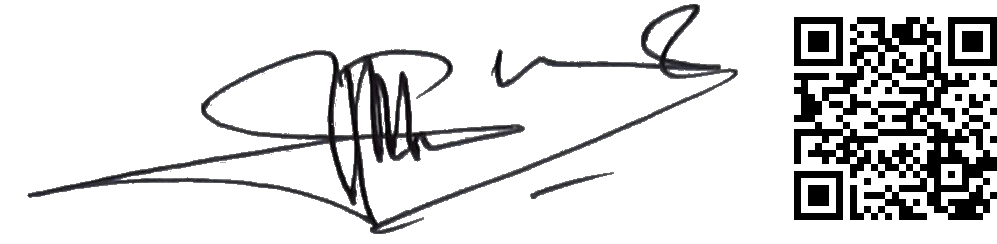
Tanggal: 23/07/2020

**Ir. Sri Endang Anjarwani, M.Kom   
NIP. 196604032006042001**

Mengetahui,

****Ketua Program Studi Teknik Informatika Fakultas Teknik

Universitas Mataram



**Prof. Dr. Eng. I Gede Pasek Suta Wijaya, ST., MT.**

**NIP. 197311302000031001**

**RANCANG BANGUN SISTEM INFORMASI GEOGRAFIS PEMETAAN LOKASI WISATA ALAM DENGAN MENGGUNAKAN METODE PROTOTYPING BERBASIS MOBILE ( STUDI KASUS : PULAU LOMBOK)**

Oleh :

**GILANG RAMDHANI PUTRA**

F1D016031

Telah diujikan di depan penguji

Pada tanggal 13 Juli 2020

Dan dinyatakan telah memenuhi syarat mencapai derajat Sarjana S-1

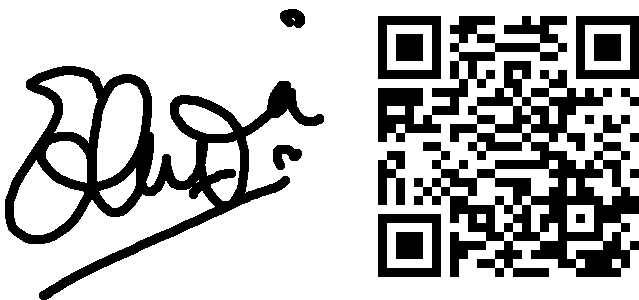
Program Studi Teknik Informatika

Susunan Tim Penguji :

1. Penguji 1

Tanggal: 21/07/2020

**Nadiyasari Agitha, S.Kom., M.MT  
NIP. 198608132018032001**

1. Penguji 2

Tanggal: 23/07/2020

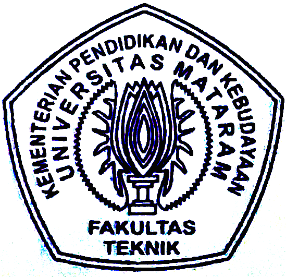
**Dr. Eng. Budi Irmawati. S.Kom., MT.**

**NIP. 197210191999032001**

1. Penguji 3

Tanggal: 21/07/2020

**Andy Hidayat Jatmika, S.T., M.Kom.   
NIP. 198312092012121001**

****Mataram, 27 Juli 2020  
 Dekan Fakultas Teknik   
Universitas Mataram

  
 **Akmaluddin, ST., M.Sc Eng., Ph.D.  
 NIP. 196812311994121001**

# HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan bahwa tugas akhir ini dengan judul “Sistem Informasi Geografis Untuk Pemetaan Lokasi Wisata Alam Menggunakan Metode Prototyping Berbasis Mobile” sepenuhnya adalah karya sendiri. Tidak ada bagian di dalamnya yang merupakan plagiat dari karya orang lain dan saya tidak melakukan penjiplakan atau pengutipan dengan cara yang tidak sesuai dengan etika keilmuan yang berlaku. Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Mataram, 21 Juli 2020

Yang membuat pernyataan,

**Gilang Ramdhani Putra**

# PRAKATA

*Assalamualaikum Wr. Wb.*

Segala puji bagi Tuhan Yang Maha Esa yang telah memberikan karunia-NYA sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Rancang Bangun Sistem Informasi Geografis Untuk Pemetaan Lokasi Wisata Alam Menggunakan Metode Prototyping Berbasis Mobile” tepat pada waktunya. Pada kesempatan ini pula, penulis menghaturkan terimakasih yang sebesar-besarnya kepada seluruh pihak yang telah mendukung agar terselesaikanya Tugas Akhir ini.

Penulis tentu menyadari Tugas Akhir ini masih jauh dari kata sempurna dan masih terdapat kesalahan dan kekurangan di dalamnya. Untuk itu, diharapkan kritik serta saran dari pembaca untuk Tugas Akhir ini, agar Tugas Akhir ini dapat menjadi lebih baik lagi. Penulis juga mengucapkan banyak-banyak terimakasih kepada semua pihak yang telah membantu penulis sehingga Tugas Akhir ini dapat selesai.

Demikian yang bisa disampaikan, sekali lagi terimakasih atas semua pihak yang telah membantu di dalam penyelesaian Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca.

*Wassalamualaikum Wr. Wb.*

Mataram, 21 Juli 2020

Penulis

# UCAPAN TERIMAKASIH

Penulis menyadari bahwa terselesaikannya Tugas Akhir ini tentunya bukan hanya dari usaha penulis saja. Tugas Akhir ini bisa selesai tepat waktu tentunya berkat dukungan dari semua pihak yang terlibat juga. Oleh karena itu, pada kesempatan ini penulis menghaturkan terimakasih yang sebesar-besarnya kepada:

1. Kedua Orang Tua dan keluarga yang selalu memberikan dukungan dari segala lini kehidupan selama perkuliahan.
2. Ibu Royana Afwani, S.T, M.T.selaku dosen pembimbing utama yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
3. Ibu Ir. Sri Endang Anjarwani, M.Kom.selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
4. Agung Rahmita Illah, Eby Sofyan, Romi Idris Sardi selaku teman diskusi selama pengerjaan Tugas Akhir.
5. Dinas Pariwisata selaku client dari Sistem Informasi Geografis ini.
6. Responden Ketika pengumpulan kebutuhan dan penelitian yang telah meluangkan waktunya sehingga saran dan masukkannya berguna untuk pengembangan sistem.
7. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah memberikan do’a dan dukungan baik moril maupun materil sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir dengan baik

Semoga Tuhan Yang Maha Kuasa selalu memberikan rahmat dan hidayah-Nya dan memberikan imbalan yang setimpal atas bantuan yang diberikan kepada penulis

# DAFTAR ISI

[HALAMAN PENGESAHAN i](#_Toc46744599)

[HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iii](#_Toc46744600)

[PRAKATA iv](#_Toc46744601)

[UCAPAN TERIMAKASIH v](#_Toc46744602)

[DAFTAR ISI vi](#_Toc46744603)

[DAFTAR GAMBAR viii](#_Toc46744604)

[DAFTAR TABEL x](#_Toc46744605)

[ABSTRAK xi](#_Toc46744606)

[ABSTRACT xii](#_Toc46744607)

BAB I [PENDAHULUAN 1](#_Toc46744609)

[1.1. Latar Belakang 1](#_Toc46744610)

[1.2. Rumusan Masalah 3](#_Toc46744611)

[1.3. Batasan Masalah 4](#_Toc46744612)

[1.4. Tujuan 4](#_Toc46744613)

[1.5. Manfaat 4](#_Toc46744614)

[1.6. Sistematika Penulisan 5](#_Toc46744615)

BAB II [TINJAUAN PUSTAKA DAN LANDASAN TEORI 6](#_Toc46744617)

[2.1 Tinjauan Pustaka 6](#_Toc46744618)

[2.2 Dasar Teori 8](#_Toc46744619)

[2.2.1. Sistem Informasi Geografis 8](#_Toc46744620)

[2.2.2. Wisata Alam 8](#_Toc46744621)

[2.2.3. Android 9](#_Toc46744622)

[2.2.4. Aplikasi Mobile 10](#_Toc46744623)

[2.2.5. Google maps API 10](#_Toc46744624)

[2.2.6. Android Studio 11](#_Toc46744625)

[2.2.7. Java Development Kit (JDK) 11](#_Toc46744626)

[2.2.8. UML (Unidentified Modelling Language) 11](#_Toc46744627)

[2.2.9. Prototype 15](#_Toc46744628)

[2.2.10.MySQL 16](#_Toc46744629)

[2.2.11.CodeIgniter 17](#_Toc46744630)

[2.2.12.Kotlin 17](#_Toc46744631)

[BAB III](#_Toc46744632) [METODE PENELITIAN 19](#_Toc46744633)

[3.1 Alat dan Bahan 19](#_Toc46744634)

[3.1.1 Alat 19](#_Toc46744635)

[3.1.2 Bahan 19](#_Toc46744636)

[3.2 Metode Pengembangan 19](#_Toc46744637)

[3.2.1 Studi Literatur 20](#_Toc46744638)

[3.2.2 Pengumpulan Kebutuhan 20](#_Toc46744639)

[3.2.3 Desain Sistem 23](#_Toc46744640)

[3.2.4 Pembuatan Prototype 37](#_Toc46744641)

[3.2.5 Evaluasi Prototype 45](#_Toc46744642)

[3.2.6 Pembuatan Produk Sebenarnya 46](#_Toc46744643)

[3.2.7 Pengujian Produk 46](#_Toc46744644)

[3.2.8 Implementasi Sistem 46](#_Toc46744645)

[BAB IV](#_Toc46744646) [HASIL DAN PEMBAHASAN 47](#_Toc46744647)

[4.1 Evaluasi Prototype 47](#_Toc46744648)

[4.1.1. Pengumpulan Kebutuhan 47](#_Toc46744649)

[4.1.2. Pembuatan Prototype Hasil Evaluasi 49](#_Toc46744650)

[4.2 Pembuatan Aplikasi Sebenarnya 54](#_Toc46744651)

[4.2.1. Pembuatan database 55](#_Toc46744652)

[4.2.2. Pembuatan API 57](#_Toc46744653)

[4.2.3. Arsitektur Aplikasi setelah diimplementasi 73](#_Toc46744654)

[4.2.4. Pembuatan SIG Pemetaan Lokasi Wisata Alam 74](#_Toc46744655)

[4.3 Pengujian Aplikasi 103](#_Toc46744656)

[4.3.1. Hasil pengujian *blackbox testing* 104](#_Toc46744657)

BAB V [PENUTUP 114](#_Toc46744659)

[5.1 Kesimpulan 114](#_Toc46744660)

[5.2 Saran 114](#_Toc46744661)

[DAFTAR PUSTAKA 115](#_Toc46744662)

# DAFTAR GAMBAR

[Gambar 2.1. Kelas Diagram 13](#_Toc44949698)

[Gambar 2.2 Sequence Diagram 14](#_Toc44949699)

[Gambar 2.3. Simbol Activity Diagram 15](#_Toc44949700)

[Gambar 3 1**.** Diagram Alur Atau *Flowchart* 20](#_Toc45168860)

[Gambar 3.2. Data Yang Didapatkan Dari Google Maps. Icon Berwarna Kuning Untuk Destinasi Pantai, Hijau Untuk Bukit, Dan Biru Untuk Air Terjun 22](#_Toc45168861)

[Gambar 3.3. Use Case Diagram 23](#_Toc45168862)

[Gambar 3.4. Activity Diagram Login Admin 24](#_Toc45168863)

[Gambar 3.5. Activity Diagram Menambah Lokasi 25](#_Toc45168864)

[Gambar 3.6.Activity Diagram Edit Lokasi 26](#_Toc45168865)

[Gambar 3.7.Activity Diagram Hapus Lokasi 26](#_Toc45168866)

[Gambar 3.8.Activity Diagram Melihat Lokasi Di Peta 27](#_Toc45168867)

[Gambar 3.9. Activity Diagram Lihat Informasi Lokasi 27](#_Toc45168868)

[Gambar 3.10.Activity Diagram List Wisata 28](#_Toc45168869)

[Gambar 3.11. Activity Diagram Filter List Wisata 28](#_Toc45168870)

[Gambar 3.12.Activity Diagram Lihat Jarak Dan Waktu 29](#_Toc45168871)

[Gambar 3.13.Class Diagram 30](#_Toc45168872)

[Gambar 3.14.Proses Login Admin Pada Sequence Diagram 30](#_Toc45168873)

[Gambar 3.15.Proses Lihat Lokasi Wisata 31](#_Toc45168874)

[Gambar 3.16.Proses Menambah Lokasi Wisata 31](#_Toc45168875)

[Gambar 3.17.Proses Edit Lokasi Wisata Pada *Sequence* Diagram 32](#_Toc45168876)

[Gambar 3.18.Proses Hapus Lokasi 32](#_Toc45168877)

[Gambar 3.19.Tampil Halaman Peta 33](#_Toc45168878)

[Gambar 3.20. Detail Lokasi 33](#_Toc45168879)

[Gambar 3.21. *List* Lokasi 34](#_Toc45168880)

[Gambar 3.22**.** Entity Relationship Diagram 34](#_Toc45168881)

[Gambar 3.23.Halaman *Login* Admin 37](#_Toc45168882)

[Gambar 3.24.Halaman Dashboard Admin 38](#_Toc45168883)

[Gambar 3.25.Halaman Maps Tambah 46   
Gambar 3.26.Halaman Form Tambah 39](#_Toc45168884)

[Gambar 3.27.Halaman Form Edit Gambar 3.28.Halaman Maps Edit 40](#_Toc45168885)

[Gambar 3.29.*Home* *User* 41](#_Toc45168886)

[Gambar 3.30.Halaman Maps *User………………………………………………………*…………………...49   
Gambar 3.31.Fitur Filter Kategori 42](#_Toc45168887)

[Gambar 3.32.Halaman Detail 43](#_Toc45168888)

[Gambar 3.33.*List* Wisata *User* 44](#_Toc45168889)

[Gambar 3.34.Halaman Bantuan 45](#_Toc45168890)

[Gambar 4.1.Halaman Tambah Setelah Evaluasi 49](#_Toc45880312)

[Gambar 4.2.Halaman Edit Setelah Evaluasi 50](#_Toc45880313)

[Gambar 4.3.Halaman Detail Setelah Evaluasi 51](#_Toc45880314)

[Gambar 4.4.Halaman Validasi Wisata 52](#_Toc45880315)

[Gambar 4.5.Halaman Akun Wisata 53](#_Toc45880316)

[Gambar 4.6.Halaman Peta Setelah Evaluasi 54](#_Toc45880317)

[Gambar 4.7.Struktur Database 55](#_Toc45880318)

[Gambar 4.8.Tabel Admin 55](#_Toc45880319)

[Gambar 4.9.Tabel Kabupaten 55](#_Toc45880320)

[Gambar 4.10.Tabel Kecamatan 56](#_Toc45880321)

[Gambar 4.11.Tabel Destinasi 56](#_Toc45880322)

[Gambar 4.12.Tabel Kategori 57](#_Toc45880323)

[Gambar 4.13.Tabel Wisatawan 57](#_Toc45880324)

[Gambar 4.14.Kelas Diagram Api 58](#_Toc45880325)

[Gambar 4.15.Arsitektur Aplikasi 73](#_Toc45880326)

[Gambar 4.16.Sitemap Aplikasi Untuk Admin 74](#_Toc45880327)

[Gambar 4.17.Sitemap Aplikasi Untuk Wisatawan 75](#_Toc45880328)

[Gambar 4.18.Kelas Activity 76](#_Toc45880329)

[Gambar 4.19.Adapter 78](#_Toc45880330)

[Gambar 4.20.Kelas Model 80](#_Toc45880331)

[Gambar 4.21.Package Service 84](#_Toc45880332)

[Gambar 4.22.Layout Wisatawan 84](#_Toc45880333)

[Gambar 4.23.Layout Ganti Email 85](#_Toc45880334)

[Gambar 4.24.Layout Ganti Nama 86](#_Toc45880335)

[Gambar 4.25.Layout Ganti Password 86](#_Toc45880336)

[Gambar 4.26.Layout Login 87](#_Toc45880337)

[Gambar 4.27.Layout Daftar 87](#_Toc45880338)

[Gambar 4.28.Sebelum Login……………………………………………………………………………92   
Gambar 4.29.Setelah Login 88](#_Toc45880339)

[Gambar 4.30.Layout Home 88](#_Toc45880340)

[Gambar 4.31.Layout List 89](#_Toc45880341)

[Gambar 4.32.Layout\_Detail 89](#_Toc45880342)

[Gambar 4.33.Layout Maps 90](#_Toc45880343)

[Gambar 4.34.Package Activity 91](#_Toc45880344)

[Gambar 4.35.Package Model 93](#_Toc45880345)

[Gambar 4.36.Package Adapter 96](#_Toc45880346)

[Gambar 4.37.Kelas Sessionmanager 97](#_Toc45880347)

[Gambar 4.38.Package Services 97](#_Toc45880348)

[Gambar 4.39.Package Layout 98](#_Toc45880349)

[Gambar 4.40.Layout Tambah 99](#_Toc45880350)

[Gambar 4.41.Layout Detail 99](#_Toc45880351)

[Gambar 4.42.Layout List 100](#_Toc45880352)

[Gambar 4.43.Layout Login 101](#_Toc45880353)

[Gambar 4.44.Layout Tambah 101](#_Toc45880354)

[Gambar 4.45.Layout Update 102](#_Toc45880355)

[Gambar 4.46.Halaman Verifikasi 103](#_Toc45880356)

[Gambar 4. 47. Grafik Perentase Rata-Rata Responden Pengguna Aplikasi Siwita 113](#_Toc45880357)

# DAFTAR TABEL

[Tabel 3.1. Tabel Admin 35](#_Toc45169204)

[Tabel 3.2.Tabel Kabupaten 35](#_Toc45169205)

[Tabel 3.3.Tabel Kecamatan 35](#_Toc45169206)

[Tabel 3.4.Tabel Destinasi 36](#_Toc45169207)

[Tabel 3.5.Tabel Kecamatan 36](#_Toc45169208)

[Tabel 3.6.Tabel Wisatawan 36](#_Toc45169209)

[Tabel 4.1.Pengujian Black Box Pada Fungsi Login 103](#_Toc45457362)

[Tabel 4.2.Pengujian Black Box Pada Fungsi Tambah 104](#_Toc45457363)

[Tabel 4.3.Pengujian Black Box Pada Fungsi Update 104](#_Toc45457364)

[Tabel 4.4.Pengujian Black Box Pada Fungsi Delete 105](#_Toc45457365)

[Tabel 4.5.Pengujian Black Box Pada Menampilkan Wisata 105](#_Toc45457366)

[Tabel 4.6.Pengujian Black Box Pada Menampilkan Wisata 106](#_Toc45457367)

[Tabel 4.7.Pengujian Black Box Pada Filter Kategori Wisata 107](#_Toc45457368)

[Tabel 4.8.Pengujian Black Box Pada Filter Kabupaten 107](#_Toc45457369)

[Tabel 4.9.Pengujian Black Box Pada Filter Kecamatan 108](#_Toc45457370)

[Tabel 4.10.Hasil Pengujian Kuisoner Dengan Pegawai Dinas Pariwisata Ntb 110](#_Toc45457371)

[Tabel 4.11.Hasil Pengujian Kuisioner Dengan Wisatawan 111](#_Toc45457372)

# ABSTRAK

Lombok merupakan salah satu tujuan wisata yang semakin popular bagi wisawatan nusantara (wisnus) maupun wisata mancanegara (wisman).Pada tahun 2019 Lombok juga mendapatkan gelar sebagai “Destinasi Wisata Halal” terbaik di Indonesia oleh Indonesia Muslim Travel Index (IMTI). Berkat popularitasnya yang meningkat, menjadikan Lombok sebagai salah satu tempat berwisata yang digemari oleh wisatawan. Berdasarkan survei yang telah dilakukan, kendala dari pihak Dinas Pariwisata Nusa Tenggara Barat yaitu masih belum adanya aplikasi terkait dengan wisata alam di pulau Lombok. Narasumber dari pihak Dinas Pariwisata NTB merasa membutuhkan sebuah sistem yang dapat menginformasikan terkait dengan wisata alam di pulau Lombok untuk mempermudah wisatawan dalam mencari lokasi wisata yg diinginkan, serta dapat menjangkau seluruh wisatawan baik domestik maupun mancanegara. Berdasarkan hasil survey terhadap 47 responden terkait dengan pulau Lombok, dan google maps didapatkan hasil 88,2% mengenal pulau Lombok karena wisata alamnya, dan 70.2% responden sering menggunakan aplikasi google maps untuk mencari lokasi yang diinginkan dan 100% responden setuju dengan dibuatnya aplikasi ini. Untuk itulah, dalam penelitian ini akan dibuat suatu Sistem Informasi Geografis untuk Pemetaan Lokasi Wisata Alam Dengan Menggunakan Metode Prototyping Berbasis Mobile dengan harapan dapat membantu Dinas Pariwisata NTB dalam meningkatkan pelayanannya terhadap masyarakat terutama para wisatawan. Pada perancangannya, aplikasi ini akan menggunakan bahasa pemrograman Kotlin, serta framework CodeIgniter sebagai pengaksesan API yang memanggil query pada database.

**Kata kunci**: Sistem Informasi Geografis, Metode Prototyping, Android Apps, Google Maps, Lombok

# ABSTRACT

Lombok is one of the increasingly popular tourist destinations for domestic and foreign tourists. In 2015 and 2016, Lombok Island won World’s Best Halal Honeymoon Destination and World’s Best Halal Tourism Destinations at The World Halal Travel Summit & Exhibition. In addition, in 2019 Lombok also received the title as the best "Halal Tourism Destination" in Indonesia by the Indonesia Muslim Travel Index (IMTI). Because of its increasing popularity, making Lombok as one of the tourist attractions favored by tourists. The West Nusa Tenggara Tourism Office is an element of implementing government affairs in the field of culture and tourism in NTB. Based on a survey conducted by the author, the obstacle of the West Nusa Tenggara Tourism Office is that there are still no applications related to natural tourism on the island of Lombok. A guest speaker from the NTB Tourism Office said that they need a system that can inform Lombok's natural attractions to facilitate tourists in finding the desired tourist location, and can also reach all tourists both domestic and foreign. Based on the results of a survey conducted on 47 respondents related to the island of Lombok and Google Maps, the results obtained 88.2% knew the island of Lombok because of its natural attractions, and 70.2% of respondents often used the Google maps application to find the desired location, and 100% of respondents agreed if this application was made. For these reasons, in this study a Geographic Information System for Mapping Natural Tourism Location Mapping Using the Mobile-Based Prototyping Method is expected to help the NTB Tourism Office improve its services to the community, especially tourists. In its design, this application will use the Kotlin programming language, and the CodeIgniter framework as an API access that calls queries on the database.

Keywords: Geographic Information Systems, Prototyping Methods, Android Apps, Google Maps, Lombok.

# BAB I

# PENDAHULUAN

## Latar Belakang

Pada tahun 2017, NTB telah berhasil mendapatkan target 3.508.903 pengunjung. Sebanyak 2.078.654 orang adalah wisatawan nusantara (wisnus), dan 1.430.249 merupakan wisatawan mancanegara (wisman)[1].Salah satu pulau yang ada di NTB adalah pulau Lombok. Lombok merupakan salah satu tujuan wisata yang semakin popular bagi wisawatan nusantara (wisnus) maupun wisata mancanegara (wisman).Pada tahun 2015 dan 2016, Pulau Lombok meraih penghargaan *World’s Best Halal Honeymoon Destination dan World’s Best Halal Tourism Destination* dalam ajang *The World Halal Travel Summit & Exhibition*[2]. Kemudian pada tahun 2019 Lombok juga mendapatkan gelar sebagai “Destinasi Wisata Halal” terbaik di Indonesia oleh *Indonesia Muslim Travel Index* (IMTI), yang diikuti oleh Aceh, Riau, dan Kepulauan Riau[3].

Di era digital ini, promosi pariwisata melalui internet merupakan cara yang paling efektif dalam mempromosikan pariwisatanya, mengingat kemudahan akses dan jaringan informasi dari internet yang sangat luas yang memungkinkan promosi pariwisata dapat diakses kapan, di mana, dan oleh siapa saja[2]. Lembaga riset *digital marketing* Emarketer memperkirakan pada 2018 jumlah pengguna aktif *smartphone* di Indonesia lebih dari 100 juta orang. Apabila dimanfaatkan secara maksimal, *smartphone* dapat dimanfaatkan sebagai pemberi informasi dalam bentuk peta, seperti *Google maps*[4].

*Google maps* merupakan peta yang dihasilkan oleh foto udara, citra satelit, dan olahan SIG, sehingga gambar muka peta yang ada di *google maps* mencakup seluruh permukaan bumi. Untuk kepuasan penggunanya *google maps* sering melakukan pembaruan peta yang ditampilkan, terutama jika ada sesuatu yang penting terjadi di lokasi tertentu[5].

Pada tanggal 18 Mei 2020, dilakukan wawancara dengan salah satu pegawai Dinas Pariwisata di bidang destinasi yaitu ibu Irma. Dimana informasi yang didapatkan dari pihak Dinas Pariwisata Nusa Tenggara Barat yaitu masih belum adanya aplikasi terkait dengan wisata alam di pulau Lombok, oleh karena itu dinas pariwisata Nusa Tenggara Barat membutuhkan sebuah sistem yang dapat menginformasikan terkait dengan wisata alam di pulau Lombok untuk mempermudah wisatawan dalam mencari lokasi wisata yg diinginkan, serta dapat menjangkau seluruh wisatawan baik domestik maupun mancanegara. Dalam sistem tersebut, narasumber juga menginginkan agar fitur yang akan dibuat diantaranya mencakup fitur pencarian, filter kategori, filter kabupaten dan kecamatan, fitur desa wisata, rute dan informasi wisata. Selain itu, dinas pariwisata juga memberikan data desa wisata yang ada di Pulau Lombok, yaitu sebanyak 59 lokasi desa wisata yang tersebar di seluruh Pulau Lombok. Selanjutnya Penulis melakukan survei menggunakan website *google maps* untuk mendapatkan daftar lokasi wisata alam di Pulau Lombok, didapatkan 140 pantai, 23 air terjun, dan 8 bukit. Data-data tersebut adalah data yang akan diberikan ke dinas pariwisata untuk dilakukan validasi. Pada aplikasi ini, nantinya wisatawan dapat menambahkan lokasi wisata alam yang belum terdaftar pada aplikasi, namun tentunya akan dibedakan antara wisata yang telah terdaftar oleh dinas pariwisata dan belum terdaftar oleh wisatawan.

Penulis juga melakukan survei terkait dengan aplikasi wisata alam di Pulau Lombok pada aplikasi playstore dengan *keyword* pertama yaitu wisata alam Lombok, selanjutnya *keyword* kedua wisata Lombok. Hasilnya belum terdapat aplikasi di Pulau Lombok yang dikhususkan untuk wisata alam. Hal ini menyebabkan penulis melakukan penelusuran lebih lanjut mengenai kebutuhan sebuah aplikasi penyedia informasi seputar Pulau Lombok bagi masyarakat luas.

Berdasarkan beberapa permasalahan di atas, maka dibuatlah sebuah Sistem Informasi Geografis untuk pemetaan lokasi wisata alam berbasis *mobile*. Sebuah sistem yang dapat menampilkan sebagian besar lokasi wisata alam yang valid di Pulau Lombok pada peta digital, memiliki fitur pencarian dan filter kategori yang dapat memudahkan wisatawan mencari dan menemukan wisata yang diinginkan, mengetahui di sebuah kabupaten dan kecamatan berapa banyak destinasi wisata yang dimiliki, serta sebagai penunjuk jalan dari lokasi pengguna ke lokasi tujuan. Lokasi yang dapat dikatakan valid adalah Ketika lokasi tersebut telah di verifikasi melalui observasi langsung dan dicocok-kan dengan data yang ada di Dinas Pariwisata NTB, yang kemudian lokasi wisata di aplikasi terdapat keterangan “Sudah Tervalidasi”. Namun, jika belum valid, maka keterangan lokasi pada aplikasi keterangannya “Belum Tervalidasi”

Metode yang akan digunakan untuk pengembangan sistem informasi geografis ini adalah metode *prototyping*, karena metode ini memiliki daya interaksi antara *developer* dengan *client* (dalam kasus ini adalah Dinas Pariwisata bidang Destinasi) yang cukup tinggi, sehingga dapat menunjang fungsionalitas sistem yang dibuat karena sesuai dengan apa yang *client* inginkan. Dengan menggunakan metode *prototyping* ini dari segi kebutuhanya yang akurat, *user experience*nya meningkat, dan kesalahan serta redudansinya baik karena dilakukan evaluasi secara berkala dan terus menerus oleh Dinas Pariwisata. Diharapkan dengan adanya sistem informasi ini dapat meningkatkan potensi wisata alam di Lombok sehingga dikenal lebih luas. Dapat menjadi acuan sebagai sistem informasi rujukan untuk mengetahui lokasi wisata alam yang valid posisinya. Mudah digunakan dengan berbagai fitur mendukung. Sehingga dapat membangkitkan daya saing dalam menjaga kelestarian wisata alam di daerahnya masing-masing.

## Rumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan, rumusan masalah yang dapat diambil adalah:

1. Bagaimana perancangan sistem informasi geografis untuk pemetaan lokasi wisata alam agar wisatawan mendapatkan informasi lokasi yang valid?
2. Bagaimana membangun sistem informasi geografis untuk pemetaan lokasi wisata alam di pulau Lombok dengan fitur yang memudahkan wisatawan dalam mendapatkan lokasi yang diinginkan, serta dapat ikut berkontribusi dalam penambahan lokasi wisata yang belum terdaftar sehingga dinas pariwisata dapat melakukan validasi terhadap lokasi wisata yang ditambahkan tersebut?
3. Bagaimana membuat aplikasi yang dapat membantu pemerintah daerah untuk mengelola dan memberikan informasi daerah wisata alam di pulau Lombok?

## **Batasan Masalah**

Penelitian ini memiliki batasan-batasan masalah untuk memberikan lingkup penelitian agar lebih terfokus ketika pengerjaan. Adapun batasan masalah yang diberikan adalah sebagai berikut:

1. Sistem informasi ini diperuntukan hanya lokasi wisata alam yang ada di Pulau Lombok.
2. Sistem informasi geografis ini berbasis *mobile*
3. Pembuatan peta wisata alam ini menggunakan peta dari *Google maps*
4. Sistem informasi ini hanya menampilkan lokasi wisata alam (pantai, bukit, air terjun, gunung, taman wisata alam, hutan, gili), tidak dengan yang lain.
5. Sistem ini membutuhkan koneksi untuk dapat diakses.
6. Fitur wilayah yang terdapat pada aplikasi adalah kabupaten dan kecamatan.

## **Tujuan**

Tujuan yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Merancang Aplikasi agar wisatawan mendapatkan lokasi yang valid dalam sistem, dengan melakukan pengumpulan kebutuhan yang dilakukan verifikasi terlebih dahulu oleh Dinas Pariwisata NTB.
2. Membangun aplikasi dengan fitur yang dapat memberikan lokasi yang diinginkan antara lain :
3. Fitur pencarian lokasi berdasarkan kategori ( Pantai, Bukit, Air Terjun, Gili dan Desa Wisata)
4. Fitur *filter* pada *list* yang dapat memberikan informasi lokasi tersebut berada di daerah mana.
5. Fitur rute menuju lokasi yang dituju, nantinya wisatawan akan diberikan beberapa saran rute menuju ke lokasi yang dituju.

## Manfaat

Manfaat dari penelitian bagi wisatawan adalah sebagai berikut:

1. Mendapatkan informasi lokasi wisata alam dengan mudah.
2. Memudahkan wisatawan mencari lokasi wisata alam yang diinginkan.
3. Wisatawan dapat berkontribusi dalam penambahan lokasi wisata.
4. Masyarakat dapat mengetahui total jumlah lokasi wisata alam yang ada di Pulau Lombok yang.

## Sistematika Penulisan

Sistematika penulisan dari penelitian ini disajikan dalam beberapa bab Antara lain sebagai berikut.

1. Bab I Pendahuluan

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat serta sistematika penulisan laporan tugas akhir.

1. Bab II Tinjauan Pustaka dan Dasar Teori

Bab ini membahas tentang penelitian-penelitian terdahulu yang mengimplementasikan metode *Prototyping* serta teori-teori sebagai referensi penulis ketika melakukan penelitian.

1. Bab III Metodelogi Perancangan

Bab ini membahas tentang metodelogi yang digunakan untuk membangun Sistem Informasi Geografis Pemetaan lokasi wisata alam di Pulau Lombok.

1. Bab IV Pembahasan

Pada bab ini merupakan pembahasan tentang analisis perangkat lunak, meliputi analisis masalah, analisis metode, analisis kebutuhan sistem, serta perancangan sistem yang terdiri dari perancangan diagram alir (*flowchart*).

1. Bab V Penutup

Bab ini berisi kesimpulan yang terkait dengan permasalahan dan saran.

# BAB II

# TINJAUAN PUSTAKA DAN LANDASAN TEORI

## Tinjauan Pustaka

Dalam pembuatannya, Sistem Informasi Pemetaan lokasi wisata alam ini merujuk pada beberapa penelitian yang sebelumnya pernah dilakukan. Penelitian-penelitian sebelumnya akan dijadikan sebagai rujukan ketika pelaksanaan pembuatan sistem informasi ini.

Sistem informasi berjudul “Sistem Informasi Geografis tempat wisata” menggunakan teknologi *Geographic Information System* yang berguna untuk mempermudah wisatawan nusantara dan mancanegara untuk eksplorasi ke Gorontalo. Metode pengembangan yang digunakan adalah *prototype* dengan membuat aplikasi yang kemudian akan disempurnakan melalui proses permintaan dari *user*. Sistem ini bertujuan untuk memberikan informasi kepada masyarakat mengenai lokasi objek wisata yang ada di Kabupaten Gianyar beserta fasilitas pendukungnya.[6]

Sistem informasi berjudul “Sistem Informasi Geografis objek wisata alam di Provinsi Sumatera Utara berbasis mobile android” menggunakan Sistem informasi Geografis (SIG), karena mempunyai kemampuan menyimpan data dan memanipulasi informasi berdasarkan data geografis. Hal inilah yang membedakan Sistem Informasi Geografis (SIG) dengan sistem informasi lainnya. Metode yang di pakai dalam peneltian ini dengan menggunakan UML (Unifed Modeling Language). Diharapkan sistem informasi yang dikembangkan mampu memberikan kemudahan dan kecepatan kepada masyarakat dalam memperoleh informasi mengenai Objek Wisata Alam.[7]

Sistem informasi berjudul “Sistem Informasi Geografis Berbasis Android Untuk Pariwisata Di Daerah Magelang” menggunakan SIG berbasis android yang terintegrasi dengan *Global Positioning System* (GPS). Aplikasi yang dibuat terhubung langsung dengan *Google maps*. Tujuan sistem ini dibangun untuk memberikan informasi lokasi pariwisata di Daerah Magelang dengan mudah dan sederhana karena dapat dioperasikan dimanapun *user* berada menggunakan piranti mobile device berbasis android.[8]

Sistem informasi berjudul “Sistem Informasi Geografis Pariwisata Pulau Lombok Berbasis Android” mampu menampilkan lokasi wisata, lokasi wisatawan, jarak wisatawan, informasi wisata, festival budaya, fasilitas dan alat transformasi yang dapat digunakan ke lokasi tujuan. Sistem ini memanfaatkan internet dan teknologi Location Based Service (LBS) dengan menggunakan GPS yang dibangun dengan platform Android. Metode pengembangan sistem yang digunakan adalah waterfall. Hasil pengujian antarmuka sistem menunjukkan bahwa responden sangat baik 50%, baik 30%, dan cukup 20%[9].

Sistem berjudul “Pembuatan Aplikasi Mobile GIS Berbasis Android Untuk Informasi Pariwisata di Kabupaten Gunung Kidul” merupakan penelitian yang memanfaatkan data koordinat dan deskripsi dari masing-masing objek wisata yang dilakukan dengan survey langsung ke lapangan dengan menggunakan GPS handheld. Aplikasi mobile GIS tentang pariwisata ini diharapkan dapat membantu wisatawan lokal maupun asing dalam berwisata. Fiturnya antara lain penunjuk arah ke lokasi yang dituju, jarak ke lokasi dan informasi umum[10].

Sistem Informasi Pemetaan Objek Wisata Lombok Berbasis Web merupakan cara yang dilakukan untuk meningkatkan kunjungan wisatawan di NTB termasuk Lombok, karena pemerintah daerah secara terus menerus memperkenalkan wisata NTB dengan berbagai cara diantaranya melalui media cetak, media TV, internet, radio, mengikuti event pariwisata dan lain sebagainya. Pembuatan sistem informasi pemetaan lokasi objek wisata di Lombok berbasis web menggunakan metodelogi pengembangan perangkat lunak *waterfall*, akan menjadi nilai tambah bagi wisata Lombok guna memberi kenyamanan dan kemudahan para wisiatawan dalam menemukan lokasi dan mengetahui keunggulan dan keuknikan dari masing-masing lokasi objek wisata. Dengan aplikasi ini, diharapkan dapat meningkatkan jumlah wisatawan yang tertarik untuk berkunjung ke Lombok dengan adanya aplikasi yang dapat memanjakan dan memudahkan para wisatawan[11].

Berdasarkan referensi diatas diketahui bahwa beberapa penelitian telah membuat sistem informasi geografis wisata, salah satunya di Lombok akan tetapi berbasis web dan data yang ditampilkan adalah wisata secara keseluruhan bukan dikhususkan untuk wisata alam. Maka pada tugas akhir ini penulis bahwa aplikasi yang akan dibuat nantinya menggunakan *Google maps* API untuk menampilkan peta wisata berbasis mobile. Begitu juga tedapat jurnal yang membahas tentang SIG Pariwisata di Lombok berbasis android, namun menggunakan metode waterfall dan wisata yang digunakan adalah wisata secara keseluruhan, bukan wisata alam saja, karena aplikasi yang dikhususkan untuk wisata alam di Pulau Lombok masih belum ada. Maka pada sistem yang dibuat kali ini menggunakan metode prototyping. Pembuatan sistem informasi geografis ini nantinya dapat memberikan informasi dan kemudahan bagi pengguna yang ingin mengetahui destinasi wisata alam yang ada di Pulau Lombok.

## Dasar Teori

Berikut adalah teori-teori dasar atau umum yang digunakan dalam penelitian ini.

### Sistem Informasi Geografis

Sistem Informasi Geografis adalah sistem yang berbasiskan komputer yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografi.SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek-objek dan fenomena dimana lokasi geografi merupakan karakteristik yang penting atau kritis untuk dianalisis. Dengan demikian, SIG merupakan sistem komputer yang memiliki empat kemampuan berikut dalam menangani data yang bereferensi geografi: masukan, manajemen data (penyimpanan dan pemnanggilan data), analisis dan manipulasi data, keluaran[12].

### Wisata Alam

Wisata alam adalah kegiatan perjalanan atau sebagian dari kegiatan tersebut yang dilakukan secara sukarela bersifat sementara untuk menikmati gejala keunikan dan keindahan alam, di taman nasional, taman hutan raya dan taman wisata alam. Sedangkan pariwisata alam adalah segala sesuatu yang berhubungan dengan wisata alam, termasuk pengusahaan obyek dan daya tarik wisata alam serta usaha-usaha yang terkait di bidang tersebut. Wisata alam atau lebih sering disebut juga sebagai ekowisata atau *ecotourism* juga adalah suatu perjalanan menuju suatu tempat tertentu di permukaan bumi untuk menikmati keindahan dan keajaiban alam tanpa sentuhan pembangunan. Baik berupa panorama alam, gemericik air di sungai, deburan ombak, heningnya suasana gua, hijaunya hutan dan bahkan kehidupan sosial budaya suatu masyarakat pedalaman yang belum tersentuh oleh teknologi modern. Ekowisata seringkali mengandung tantangan-tantangan atau lebih dikenal sebagai adventure tourism, adapun berdasarkan tantangan yang dihadapinya dapat dibedakan menjadi:

1. Petualangan beresiko tinggi yaitu memerlukan kesiapan dan keterampilan khusus, keberanian yang tinggi serta kondisi yang prima, seperti panjat tebing, arung jeram, menyelam, menelusur gua. Wisata ini juga disebut sebagai wisata minat khusus, karena tidak semua orang dapat menikmatinya dengan bebas.
2. Petualangan beresiko rendah, seperti mengunjungi taman nasional, memancing, menikmati sejuknya udara di hutan, dan berkemah. Kegiatan wisata ini pada akhirnya akan menumbuhkan kesadaran dan apresiasi terhadap lingkungannya, menumbuhkan pemahaman dan pengetahuan serta wawasan tentang alam dan kehidupan masyarakat lokal. Adapun ciri utama ekowisata ini menurut Enok Maryani (2002) bahwa :
   1. Pariwisata yang berbasis alam dan budaya masyarakat setempat
   2. Motivasi utama wisatawan adalah observasi dan aspirasi alam dan budaya tradisional setempat
   3. Mempunyai muatan pendidikan dan penambahan wawasan
   4. Umumnya berskala kecil dan pengadaan fasilitas wisata oleh masyarakat setempat
   5. Dampak terhadap lingkungan alam, sosial dan budaya sedikit
   6. Dorongan untuk konservasi alam dan budaya dilakukan bersama-sama antara pemerintah, masyarakat dan wisatawan[13].

### Android

Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.[14]

### Aplikasi Mobile

Aplikasi mobile adalah aplikasi yang telah dirancang khusus untuk platform mobile (misalnya iOS, android, atau windows mobile). Dalam banyak kasus, aplikasi mobile memiliki user interface dengan mekanisme interaksi unik yang disediakan oleh platform mobile, interoperabilitas dengan sumber daya berbasis web yang menyediakan akses ke beragam informasi yang relevan dengan aplikasi, dan kemampuan pemrosesan lokal untuk pengumpulan, analisis, dan format informasi dengan cara yang paling cocok untuk platform mobile. Selain itu aplikasi mobile menyediakan kemampuan penyimpanan persisten dalam platform[15].

### Google maps API

*Google maps* API merupakan pengembangan teknologi dari google yang digunakan untuk menanamkan Google Map di suatu aplikasi yang tidak dibuat oleh Google. *Google maps* API adalah suatu library yang berbentuk javascript 15 yang berguna untuk memodifikasi peta yang ada di *Google maps* sesuai kebutuhan. Dalam perkembangannya *Google maps* API diberikan kemampuan untuk mengambil gambar peta statis. Melakukan geocoding, dan memberikan penuntun arah. *Google maps* API bersifat gratis untuk publik. Penggunaan *Google maps* API pada pengembangan aplikasi android dengan menggunakan Eclipse dan komputer menggunakan sistem operasi windows.

Kekurangan yang ada pada *Google maps* API yaitu jika ingin melakukan akses harus terdapat layanan internet pada perangkat yang digunakan. Sedangkan kelebihan yang ada pada *Google maps* API yaitu[16]:

1. Dukungan penuh yang dilakukan Google sehingga terjamin dan bervariasi fitur yang ada pada *Google maps* API.
2. Banyak pengembang yang menggunakan *Google maps* API sehingga mudah dalam mencari referensi dalam pengembangan aplikasi.

### Android Studio

Android Studio merupakan sebuah Integrated Development Environment (IDE) khusus untuk membangun aplikasi yang berjalan pada platform android. Android studio ini berbasis pada IntelliJ IDEA, sebuah IDE untuk bahasa pemrograman Java. Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau layout, digunakan bahasa XML. Android studio juga terintegrasi dengan Androi *Software* Development Kit (SDK) untuk deploy ke perangkat android[17].

### Java Development Kit (JDK)

Java Development Kit (JDK) adalah sekumpulan perangkat lunak yang dapat kamu gunakan untuk mengembangkan perangkat lunak yang berbasis Java, sedangkan JRE adalah sebuah implementasi dari Java Virtual Machine yang benarbenar digunakan untuk menjalankan program java. Baisanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber compiler java, bundling, debuggers, development libraries dan lain sebagainya[18].

### UML (Unidentified Modelling Language)

Berikut adalah teori-teori dasar mengenai UML (*Unidentified Modelling Language*) yang digunakan dalam penelitian ini:

1. ***Use Case Diagram***

*Use Case* diagram merupakan pemodelan untuk mendeskripsikan perilaku dari sistem informasi yang akan dibuat. Dapat juga dipahami sebagai cara untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi yang ada.

Penamaan *Use Case* didefiniskan sesederhana mungkin dan mudah untuk difahami. Ada dua hal yang utama dalam *Use Case*, yaitu aktor dan *Use Case.* Simbol-simbol yang digunakan dalam *Use Case.*

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Meskipun simbol dari aktor berbentuk orang, tetapi aktor belum tentu merupakan orang.
2. *Use Case* merupakan fungsionalitas dari sebuah sistem. Dengan demikian, antara konsumen dan juga pengendara pada sistem tersebut, akan mengerti atau paham mengenai fungsi sistem yang tengah dibangun[19].
3. **ERD (*Entity Relationship* Diagram**

ERD (*Entity Relationship* Diagram) merupakan suatu *model* untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Umumnya setelah perancangan ERD selesai berikutnya adalah mendesain *database* secara fisik yaitu pembuatan tabel, *index* dengan tetap mempertimbangkan *performance*. Kemudian setelah *database* selesai dilanjutkan dengan merancang aplikasi yang melibatkan *database*[20].

1. ***Class Diagram***

*Class Diagram* adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki 3 bagian utama yaitu *attribute*, *operation*, dan *name.* Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Sususan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas *Main*

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

1. Kelas *Interface*

Kelas yang mendefinisikan dan mengatur tampilan ke pengguna. Biasanya juga disebut kelas *boundaries*.

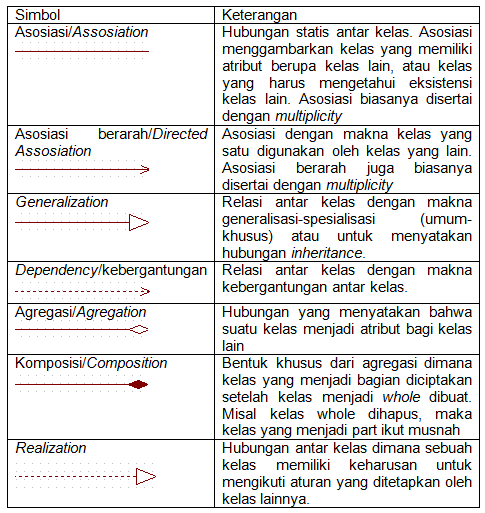
1. Kelas yang diambil dari pendefinisian *User Case*

Merupakan kelas yang menangani fungsi-fungsi yang harus ada dan diambil dari pendefinisian *Use Case.*

1. Kelas Entitas

Merupakan kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data[20].

Berikut adalah symbol-simbol pada Kelas Diagram :



Gambar 2.1. Kelas Diagram

* + - 1. ***Sequence Diagram***

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

Dalam UML, *object* pada *Sequence* diagram digambarkan dengan segi empat yang berisi nama dari *object* yang digaris-bawahi. Pada *object* terdapat 3 cara untuk menamainya yaitu: nama *object*, nama *object* dan *class*, dan nama *class.*

Dalam *Sequence* diagram, setiap *object* hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar *object* digambarkan dengan anak panah dari *object* yang mengirimkan pesan ke *object* yang menerima pesan. Berikut adalah komponen yang ada pada *Sequence* Diagram antara lain:

1. *Object* adalah komponen berbentuk kotak yang mewakili sebuah *class* atau *object*. Mereka mendemonstrasikan bagaimana sebuah *object* berperilaku pada sebuah sistem.
2. *Activation boxes* adalah komponen yang berbentuk persegi panjang yang menggambarkan waktu yang diperlukan sebuah *object* untuk menyelesaikan tugas. Lebih lama waktu yang diperlukan, maka *activation boxes* akan lebih panjang.
3. *Actors* adalah komponen yang berbentuk *stick figure*. Komponen yang mewakili seorang pengendara yang berinteraksi dengan sistem.
4. *Lifeline* adalah komponen yang berbentuk garis putus-putus. *Lifeline* biasanya memuat kotak yang berisi nama dari sebuah *object*. Berfungsi menggambarkan aktifitas dari objek[19].

Berikut adalah Simbol Sequence Diagram :



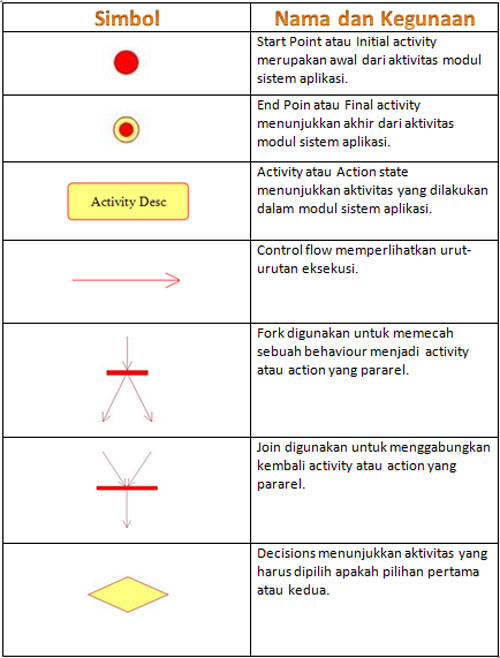
Gambar 2.2 Sequence Diagram

* + - 1. ***Activity* Diagram**

*Activity* Diagram menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah – langkah dalam proses kerja sistem yang kita buat. Struktur diagram ini juga mirip dengan flowchart. *Activity* Diagram berfungsi sebagai:

1. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses
2. Memperlihatkan urutan aktifitas proses pada sistem
3. *Activity* Diagram dibuat berdasarkan sebuah atau beberapa *use case* pada *Use Case* Diagram[19].

Berikut adalah symbol dari activity Diagram :



Gambar 2.3. Simbol Activity Diagram

### Prototype

prototyping merupakan metode pengembangan perangat lunak, yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. Dengan metode prototyping ini akan dihasilkan prototype sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem informasi. Agar proses pembuatan prototype ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan pada tahap awal, yaitu pengembang dan penguna harus satu pemahaman bahwa prototype dibangun untuk mendefinisikan kebutuhan awal. Prototype akan dihilangkan atau ditambahkan pada bagiannya sehingga sesuai dengan perencanaan dan analisis yang dilakukan oleh pengembang sampai dengan ujicoba dilakukan secara simultan seiiring dengan proses pengembangan. prototyping dimulai dengan pengumpulan kebutuhan, melibatkan pengembang dan pengguna sistem untuk menentukan tujuan, fungsi dan kebutuhan operasional sistem. Langkah-langkah dalam prototyping adalah sebagai berikut [21]:

1. Pengumpulan Kebutuhan.

2. Proses desain yang cepat.

3. Membangun prototipe.

4. Evaluasi dan perbaikan.

5. Pembuatan Produk yang sesungguhnya

Mengumpulkan kebutuhan melibatkan pertemuan antara pengembang dan pelanggan untuk menentukan keseluruhan tujuan dibuatnya perangkat lunak; mengidentifikasi kebutuhan berupa garis besar kebutuhan dasar dari sistem yang akan dibuat.

Desain berfokus pada representasi dari aspek perangkat lunak dari sudut pengguna; ini mencakup input, proses dan format output. Desain cepat mengarah ke pembangunan prototipe, prototipe dievaluasi oleh pengguna dan bagian analis desain dan digunakan untuk menyesuaikan kebutuhan perangkat lunak yang akan dikembangkan. prototype diatur untuk memenuhi kebutuhan pengguna, dan pada saat itu pula pengembang memahami secara lebih jelas dan detil apa yang perlu dilakukannya.

Setelah keempat langkah prototyping dijalankan, maka langkah selanjutnya adalah pembuatan atau perancangan produk yang sesungguhnya

### MySQL

MySQL adalah sebuah basis data yang mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel sumber data menjadi berbagai macam informasi yang dibutuhkan oleh pemakai.

Secara sederhana dapat dikatakan bahwa data diolah menjadi suatu informasi. Dan pada tahapan selanjutnya, sebuah informasi akan menjadi data untuk terciptanya informasi yang lain[22].

### CodeIgniter

CodeIgniter merupakan sebuah framework yang dibuat dengan menggunakan bahasa PHP, yang dapat digunakan untuk pengembangan web secara cepat. Adapun framework sendiri dapat diartikan sebagai suatu struktur pustaka-pustaka, kelas-kelas dan infrastruktur run-time yang dapat digunakan oleh programmer untuk mengembangkan aplikasi web secara cepat. Tujuan penggunaan framework adalah untuk mempermudah pengembang web mengembangkan aplikasi web yang robust secara cepat tanpa kehilangan fleksibilitas.

Pola desain dalam pengembangan web dengan CodeIgniter menggunakan MVC (Models-View- Controller). Dimana aplikasi yang dibuat akan dipisahkan antara logika bisnis dan presentasinya, sehingga memungkinkan web programmer dan web designer bekerja secara terpisah antara satu dengan yang lain. Agar bisa mengembangkan web dengan CodeIgniter, maka perlu dipahami terlebih dahulu konsep MVC dan struktur direktori dari CodeIgniter[23].

### Kotlin

Kotlin adalah bahasa pemrograman berbasis Java Virtual Machine (JVM) yang dikembangkan oleh JetBrains. Kotlin merupakan bahasa pemrograman yang pragmatis untuk android yang mengkombinasikan object oriented (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman Java. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis desktop, web dan bahkan untuk backend. Beberapa keuntungan yang mungkin akan didapatkan jika pengembangan aplikasi beralih menggunakan Kotlin untuk mengembangkan aplikasi diatas platform JVM adalah sebagai berikut:

Dapat mengatasi NullPointerException yang umumnya terdapat pada Java.

Penulisan kode lebih ringkas dan mudah dibaca dibandingkan kode yang ditulis dengan menggunakan bahasa Java.

Mudah dipelajari

Dukungan IDE untuk mempermudah dalam pemrograman.

Perbandingan Kotlin dan java :

1. Pengukuran dilakukan dengan membangun aplikasi pada masing-masing bahasa pemrograman Java dan Kotlin. Kemudian mengukur performa aplikasi dengan mengoperasikan setiap fitur yang ada pada aplikasi. Dari operasi tersebut didapatkan hasil penggunaan CPU, memori dan kecepatan eksekusi aplikasi pada perangkat android.
2. Kotlin lebih sedikit dalam penggunaan CPU walaupun perbedaan hanya 0.65%
3. Kotlin lebih kecil dalam penggunaan memori dengan perhitungan lebih dari 2 kali lipat penggunaan memori pada Bahasa java.
4. Kotlin lebih cepat dalam mengeksekusi program[24].

# BAB III

# METODE PENELITIAN

## Alat dan Bahan

Alat dan bahan pada penelitian yang dilakukan berupa *software* dan *hardware* serta data dan informasi yang dibutuhkan selama kegiatan berlangsung.

### Alat

Berikut adalah alat-alat yang digunakan dalam melakukan pembuatan sistem informasi geografis pemetaan lokasi wisata alam :

* 1. Laptop Asus Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz
  2. Sistem Operasi Windows 10 Home
  3. Android Studio,
  4. MySQL,
  5. Apache,
  6. Visual Studio,
  7. Xampp

### Bahan

Bahan-bahan yang akan digunakan dalam pembuatan sistem informasi geografis ini adalah literatur-literatur dari jurnal, buku, penelitian sebelumnya. Selain itu, bahan lain yang didapatkan adalah :

* + - 1. Data geografis untuk sejumlah 140 pantai, 23 air terjun, 15 bukit yang didapatkan dari *google maps*,
      2. 10 data wisata dengan secara langsung turun ke lokasi wisata,
      3. Data kuesioner yang mendukung dalam pembuatan SIG,
      4. Data desa wisata dari dinas pariwisata Nusa Tenggara Barat.

## Metode Pengembangan

Dalam melakukan sebuah penelitian, diperlukannya sebuah metode yang akan menjadi patokan dalam melakukan pengembangan. Metode yang digunakan dalam pembuatan sistem informasi ini adalah metode *prototyping*. Berikut adalah tahapan pengembangan sistem yang diilustrasikan pada *flowchart* pada Gambar 3.1 :



Gambar 3 1**.** Diagram alur atau *flowchart*

### Studi Literatur

Studi literatur adalah suatu metode yang digunakan untuk mengumpukan data dan informasi yang berkaitan dengan penelitian melalui buku, jurnal, internet dan situs-situs internet. Dalam melakukan penelitian, penulis mengumpulkan data dan informasi melalui kuesioner *google maps* dan observasi langsung, kemudian kotlin, Mysql, *google maps* api untuk mendukung penyelesaian penelitian yang dilakukan.

### Pengumpulan Kebutuhan

Analisis Kebutuhan merupakan tahap awal dalam penelitian kali ini. Dilakukan analisis Kebutuhan agar penulis dapat mengetahui kebutuhan yang diperlukan dalam pengembangan sistem. Pengembangan sistem yang dilakukan adalah sistem informasi geografis untuk pemetaan lokasi wisata alam dengan studi kasus di Pulau Lombok, sebelumnya sistem seperti ini sudah diterapkan pada *google maps*, namun sistem tersebut menampilkan keseluruhan terkait dengan destinasi wisata. Sedangkan, data yang dibutuhkan pada sistem yang akan dibangun hanya data dan informasi tentang wisata alam saja seperti nama destinasi, deskripsi, garis bujur dan garis lintang. Sistem ini ditujukan untuk wisatawan yang ingin berwisata di Pulau Lombok.

Sistem pada aplikasi ini dapat memberikan memberikan infromasi terkait dengan destinasi wisata alam seperti lokasi pada peta, jarak dari lokasi pengguna ke tujuan wisata, dan waktu yang dibutuhkan. Berikut adalah cara yang digunakan untuk mengumpulkan kebutuhan:

* + - 1. **Wawancara**

Cara pertama yang dilakukan untuk mendapatkan kebutuhan sistem adalah dengan melakukan wawancara dengan salah satu pegawai dari Dinas Pariwisata di bidang Destinasi. Pertanyaan yang diajukan terkait dengan apakah aplikasi seperti yang akan dibuat ini sudah ada atau tidak sebelumnya, saran fitur apa saja yang dapat memudahkan masyarakat menemukan wisata alam yang diinginkan. Pertanyaan selanjutnya, apakah dinas pariwisata dapat membantu melakukan validasi lokasi dan memberikan data jika masih ada lokasi yang dimuat di aplikasi.

* + - 1. **Kuesioner**

Pengumpulan kebutuhan selanjutnya dengan membuat kuesioner yang akan dibagikan ke masyarakat/wisatawan terkait dengan aplikasi yang akan dibuat. Tujuan dari dibuatnya kuesioner ini untuk mengetahui pandangan dari wisatawan terkait dengan Pulau Lombok dan Aplikasi yang akan dibuat. Pertanyaan-pertanyaan yang diberikan seputar pengetahuan wisatawan tentang Pulau Lombok, lokasi wisata alam yang sering di kunjungi, aplikasi yang biasa digunakan untuk berwisata, setuju atau tidaknya jika aplikasi seperti ini dibuat, dampak kedepannya aplikasi ini bagi masyarakat dan yang terakhir setuju atau tidak jika aplikasi ini dibuat.

* + - 1. **Mengambil data dari *google maps***

Untuk mengumpulkan kebutuhan yang terkait dengan data-data pada sistem, maka digunakan *google maps* untuk mengambil informasi lokasi dengan cara :

* + - 1. Membuat akun google
      2. Membuka akun *google maps*
      3. Membuat peta pribadi untuk pembuatan marker pada peta
      4. Memilih lokasi dengan menambahkan ke *google maps*
      5. Sehingga didapatkan 140 pantai, 23 air terjun, dan 15 bukit seperti pada gambar berikut :



Gambar 3.2. Data yang didapatkan dari google maps. Icon berwarna kuning untuk destinasi pantai, hijau untuk bukit, dan biru untuk air terjun

Nantinya, data yang telah didapatkan dari google maps ini akan dicocokkan dengan data yang dimiliki oleh Dinas Pariwisata berdasarkan yang terdaftar pada Kawasan Strategis Pariwisata Daerah. Baik dari segi informasi maupun kordinat lokasinya.

* + - 1. **Observasi langsung**

Observasi merupakan salah satu cara yang digunakan untuk mengumpulkan kebutuhan sistem, sama seperti *google maps* yaitu untuk mendapatkan garis lintang dan garis bujur dengan cara turun langsung ke lokasi wisata alam yang ada di Lombok baik itu untuk menambakan lokasi wisata yang belum terdapat di *google maps* atau melakukan verikasi terkait dengan lokasi wisata yang sudah ada. Seperti pantai lancing, pantai tampah, pantai ampenan, pantai selong belanak, pantai tanjung karang, pantai kuta, pantai Senggigi, pantai kuranji, pantai loang baloq, pantai serangan, bukit merese, bukit nanggi, pantai kerandangan dan pantai telawas sudah dilakukan verifikasi bahwa lokasi wisata tersebut sudah benar.

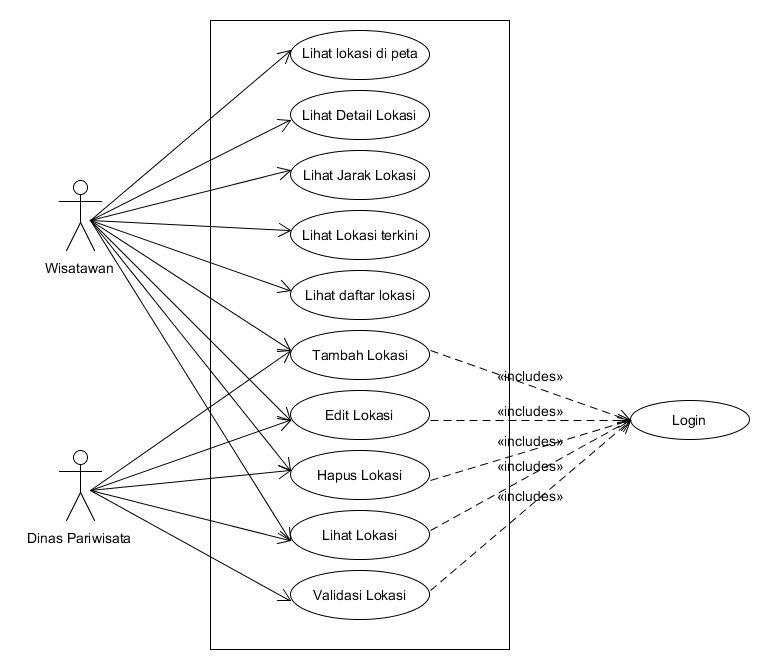
Data yang didapatkan Ketika observasi langsung juga akan dicocokkan dengan data yang sudah dimiliki oleh dinas pariwisata berdasarkan Kawasan Strategis Pariwisata Daerah. Begitu juga untuk kordinat dan informasi lokasinya.

### Desain Sistem

Setelah mengetahui dan mendapatkan seluruh kebutuhan yang diperlukan dalam pengembangan sistem, maka selanjutnya dilakukan pembuatan desain sistem. Pada tahap ini dilakukan desain sistem dengan pendekatan berorientasi obyek yaitu *Unidentified Modelling Language* (UML) dan Design *Prototype* :

**3.2.3.1. Use Case Diagram**

Berikut adalah Use Case Diagram pada Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam :



Gambar 3.3. Use Case Diagram

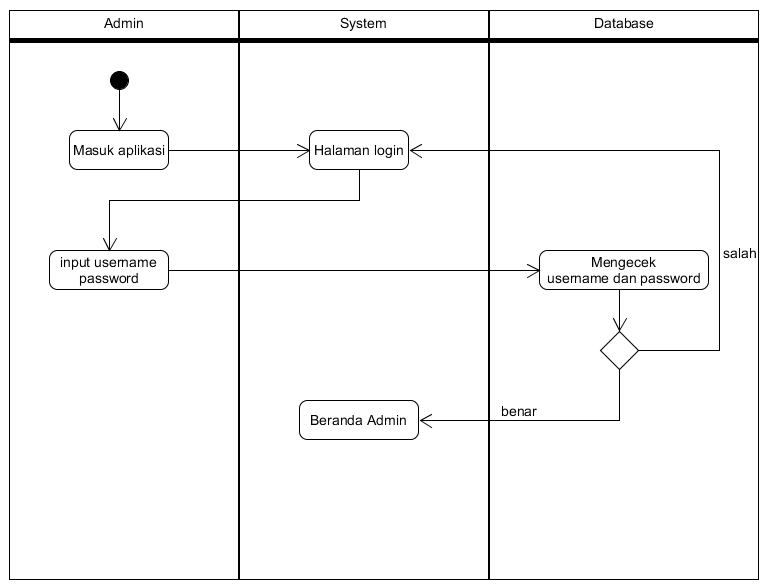
Pada Use Case Diagram Gambar 3.3. hanya terdapat 2 Aktor, yaitu wisatawan dan dinas pariwisata. Dimana wisatawan dapat melakukan berbagai aktivitas seperti melihat lokasi wisata, melihat *list* daftar wisata, melihat lokasi terkini, melihat lokasi wisata dari posisi terkini, dan melihat detail informasi lokasi wisata yang dipilih, menambah lokasi wisata yang belum ada di aplikasi, melakukan edit terhadap lokasi yang ditambahkan oleh wisatawan, menghapus lokasi wisata. Sedangkan untuk dinas pariwisata adalah admin dari aplikasi yang bertujuan untuk melakukan *input* data ketika terdapat lokasi wisata alam yang baru, edit data untuk melakukan update pada data lokasi wisata alam, menghapus data ketika terdapat kesalahan atau kekeliruan pada saat melakukan *input* data, melihat data yang ditampilkan pada halaman utama dari admin, dan melakuka validasi terhadap lokasi wisata yang ditambahkan oleh wisatawan.

* + - 1. **Activity Diagram**

Berikut adalah Activity Diagram Pada Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam :

* + - * 1. **Activity Diagram *Login* Admin/Wisatawan**

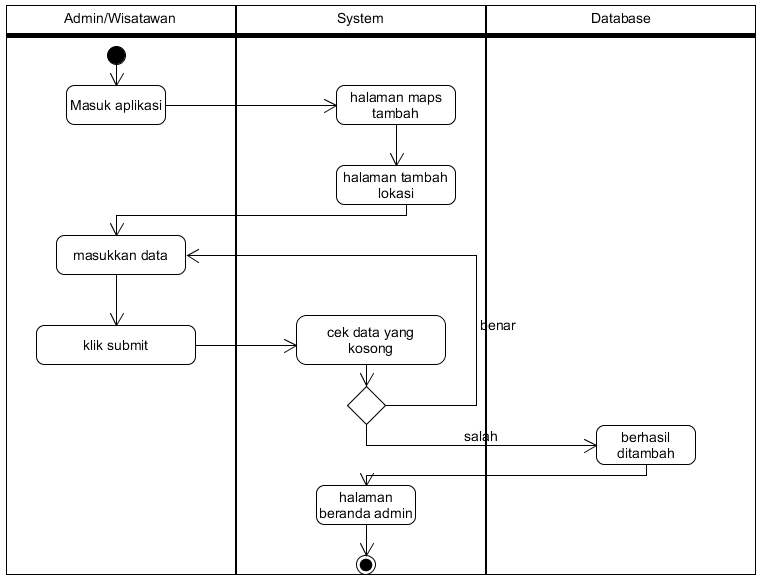
Sistem *login* pada aplikasi pemetaan lokasi wisata ini hanya berlaku untuk admin. Dimana *password* dan *username* admin telah dibuat langsung di database, sehingga tidak terdapat fitur registrasi. Alur dari fitur *login* adalah admin/wisatawan memasukkan *username* dan *password* yang sudah terdaftar, kemudian aplikasi mengecek ke database apakah *password* dan *username* tersebut sudah benar. Jika benar, maka admin akan diarahkan ke halaman utama, sedangkan jika tidak maka admin akan tetap diarahkan ke halaman *login* tersebut. Berikut adalah *activity diagram* *login* admin :



Gambar 3.4. Activity Diagram Login Admin

* + - * 1. **Activity Diagram menambah lokasi**

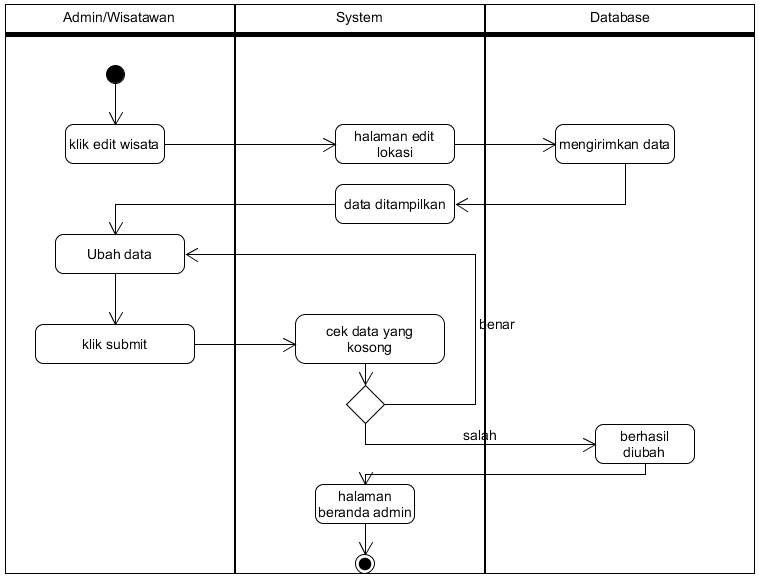
Sistem tambah lokasi pada aplikasi pemetaan lokasi wisata ini bertujuan untuk menambahkan lokasi wisata. Dimana terdapat halaman yang berisikan form untuk informasi detail dari setiap lokasi yang akan di-*input*-kan. Ketika berhasil *login*, maka admin akan diarahkan ke halaman utama, dan pada halaman utama tersebut terdapat tombol tambah lokasi. Kemudian, untuk menambah lokasi admin harus memasukkan informasi terkait dengan lokasi wisata alam yang ingin di tambahkan, seperti latitude, longitude, tempat ibadah, biaya masuk dan sebagainya. Jika terdapat form yang harus diisi, namun tidak di *input*-kan maka akan menampilkan pesan kesalahan pada halaman tersebut. Berikut adalah Diagram menambahkan lokasi wisata :



Gambar 3.5. Activity Diagram menambah lokasi

* + - * 1. **Activity Diagram Edit Lokasi**

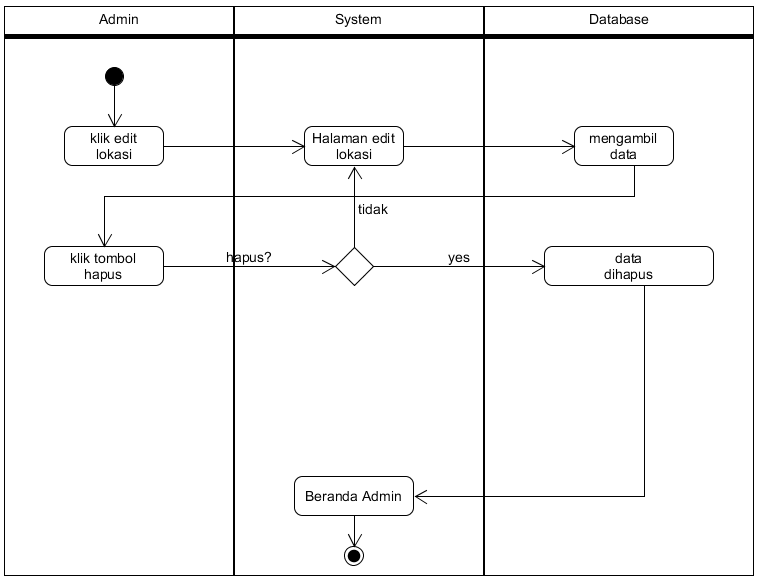
Seperti halnya menambah lokasi, sistem edit lokasi pada aplikasi pemetaan lokasi wisata ini bertujuan untuk mengubah lokasi wisata. Dimana terdapat halaman yang berisikan form untuk informasi detail dari setiap lokasi yang akan di-*input*-kan. Ketika berhasil *login*, maka admin akan diarahkan ke halaman utama, dan pada halaman utama tersebut daftar lokasi wisata. Kemudian, untuk menghapus lokasi admin harus menekan lama pada lokasi yang ingin di dihapus, maka akan menampilkan pilihan edit, lihat, dan hapus. Selanjutnya admin klik edit untuk mengubah informasi terkait dengan lokasi wisata alam yang ingin di ubah. Jika terdapat form yang harus diisi, namun tidak di *input*-kan maka akan menampilkan pesan kesalahan pada halaman tersebut. Berikut adalah Diagram mengubah lokasi wisata :



Gambar 3.6.Activity Diagram edit lokasi

* + - * 1. **Activity Diagram Hapus Lokasi**

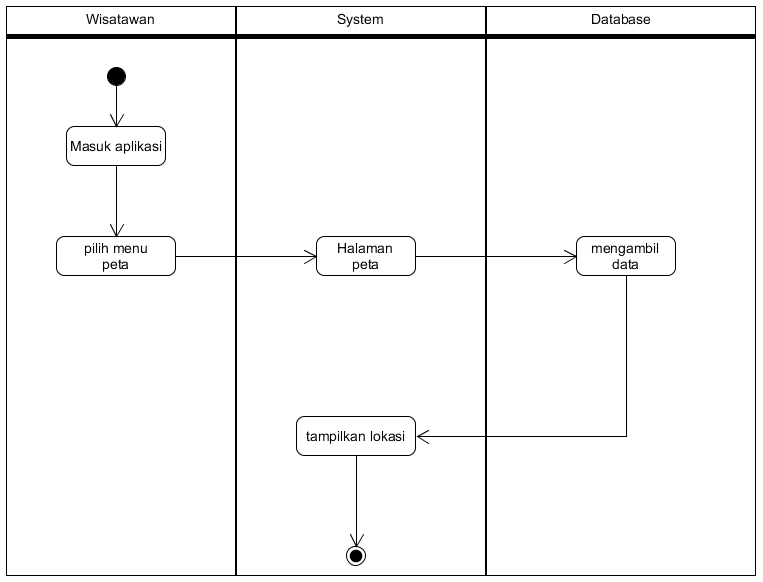
Seperti halnya menambah dan edit lokasi, sistem hapus lokasi pada aplikasi pemetaan lokasi wisata ini bertujuan untuk menghapus lokasi wisata dari database. Ketika berhasil *login*, maka admin akan diarahkan ke halaman utama, dan pada halaman utama tersebut daftar lokasi wisata. Kemudian, untuk menghapus lokasi admin harus menekan lama pada lokasi yang ingin di dihapus, maka akan menampilkan pilihan edit, lihat, dan hapus. Selanjutnya admin memilih hapus tuntuk menghapus lokasi dari database. Berikut adalah Diagram menghapus lokasi wisata :



Gambar 3.7.Activity Diagram hapus lokasi

* + - * 1. **Activity Diagram Melihat Lokasi di peta**

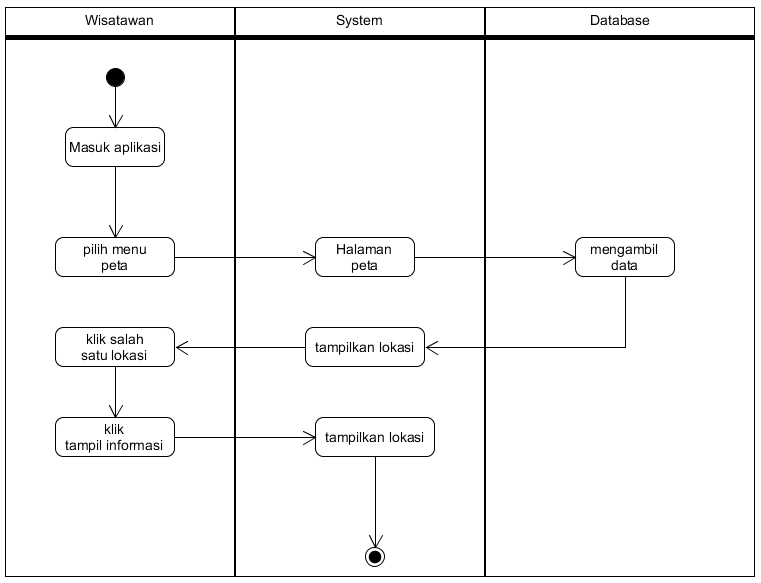
Pada Gambar 3.8. Sistem melihat lokasi di peta merupakan salah satu fitur pada *user*. Dimana *user* dapat melihat seluruh lokasi wisata yang ada di Pulau Lombok ketika aplikasi pertama kali dibuka. Lokasi yang ditampilkan berupa icon pin di setiap wilayahnya, ketika *user* menekan pin, maka akan menampilkan nama dari lokasi wisata.



Gambar 3.8.Activity Diagram melihat lokasi di peta

* + - * 1. **Activity Diagram Lihat Informasi Lokasi**

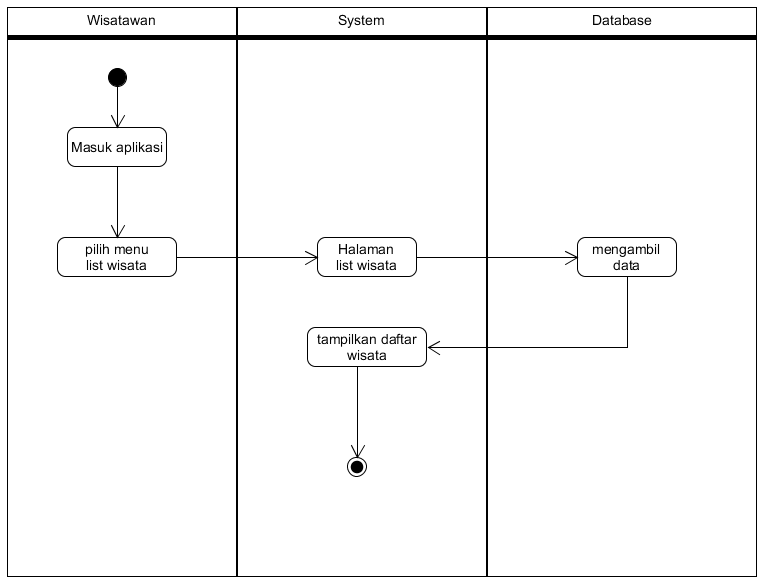
Pada Gambar 3.9. melihat informasi lokasi di peta merupakan salah satu fitur pada *user*. Dimana *user* dapat melihat seluruh lokasi wisata yang ada di Pulau Lombok ketika aplikasi pertama kali dibuka. Lokasi yang ditampilkan berupa icon pin di setiap wilayahnya, ketika *user* menekan pin, maka akan menampilkan nama dari lokasi wisata tersebut. Kemudian *user* dapat melihat detail informasi dengan klik lihat informasi.



Gambar 3.9. Activity Diagram lihat informasi lokasi

* + - * 1. **Activity Diagram Melihat *list* wisata**

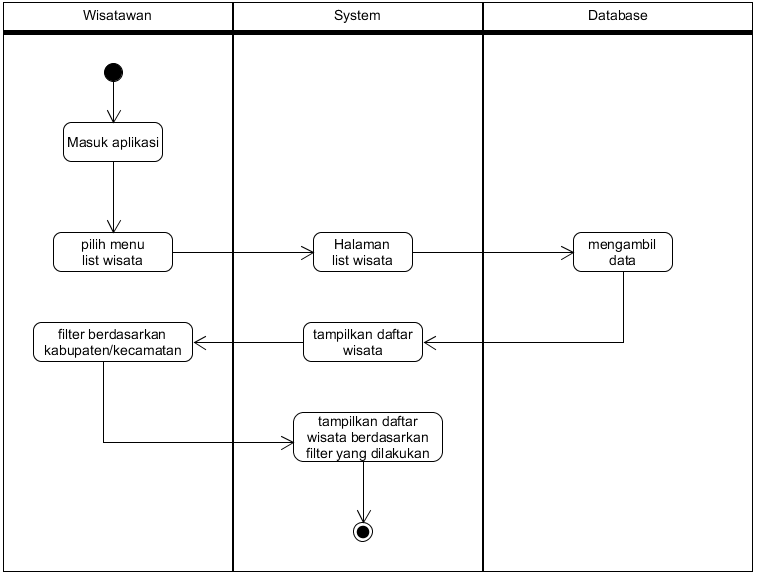
Pada Gambar 3.10. Activity diagram ini wisatawan dapat melihat *list* daftar wisata yang ada di Pulau Lombok. Dimana wisatawan dapat memilih *menu* *list* wisata Ketika masuk ke aplikasi, maka aplikasi akan mengarahkan wisatawan ke halaman *list* wisata.



Gambar 3.10.Activity Diagram list wisata

* + - * 1. **Activity Diagram Filter *list* wisata**

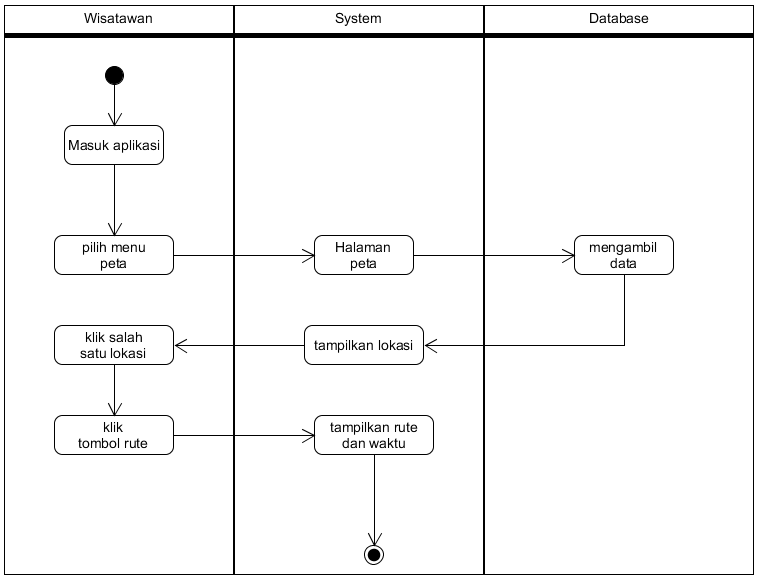
Pada halaman *list* wisata wisatawan dapat melakukan filter berdasarkan kabupaten/kecamatan yang ada di Pulau Lombok. Dengan wisatawan hanya memilih *menu* dropdown yang ada, kemudian data akan menyesuaikan sesuai dengan apa yang di pilih oleh wisatawan seperti Gambar 3.11.



Gambar 3.11. Activity Diagram filter list wisata

* + - * 1. **Activity Diagram Melihat Jarak dan waktu ke lokasi**

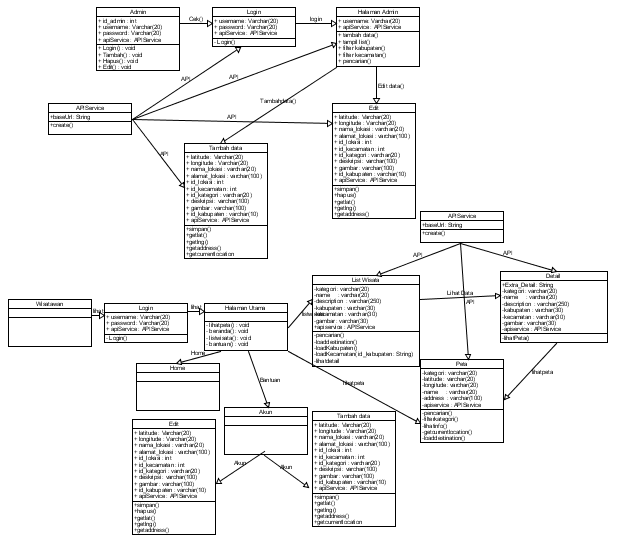
Sistem melihat jarak lokasi di peta merupakan salah satu fitur pada *user*. Dimana *user* dapat melihat seluruh lokasi wisata yang ada di Pulau Lombok ketika aplikasi pertama kali dibuka. Lokasi yang ditampilkan berupa icon pin di setiap wilayahnya, ketika *user* menekan pin, maka akan menampilkan nama dari lokasi wisata tersebut. Kemudian *user* dapat mengetahui jarak dan waktu ke lokasi wisata dari lokasi *user* dengan klik Lihat Jarak. Berikut adalah Diagram Activity Melihat jarak lokasi di peta seperti Gambar 3.12.:



Gambar 3.12.Activity Diagram lihat jarak dan waktu

* + - 1. ***Class Diagram***

Pada Gambar 3.13 merupakan *class Diagram* pada sistem informasi geografis pemetaan lokasi wisata alam di Pulau Lombok. Terdapat *class* Admin, login, halaman admin, tambah, edit. Sedangkan untuk wisatawan dapat melihat peta, beranda, *list*/daftar wisata dan bantuan



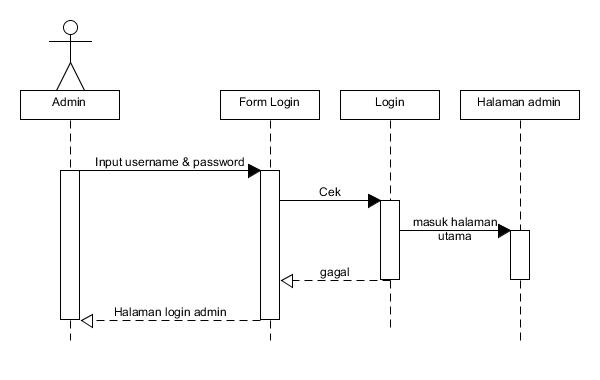
Gambar 3.13.Class Diagram

* + - 1. **Sequence Diagram**

Berikut adalah Sequence Diagram pada Sistem Informasi Geografis untuk pemetaan lokasi wisata alam :

**1. Pengguna Admin**

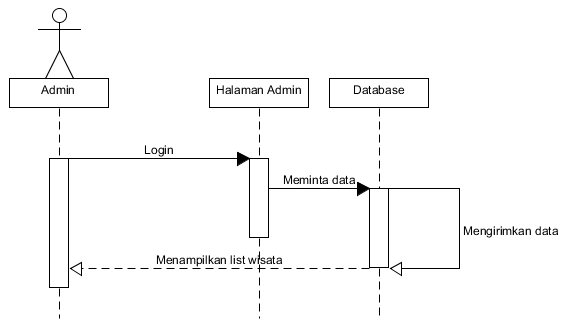
* 1. **Proses Login Admin**



Gambar 3.14.Proses Login Admin pada Sequence Diagram

Pada Gambar 3.14. Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam ini, terdapat proses *login* yang hanya terdapat pada admin saja, dimana admin harus memasukkan *username* dan *password* pada form *login*. Kemudian dilakukan pengecek-kan apakah *username* dan *password* sudah benar atau tidak, jika tidak maka akan diarahkan kembali ke halaman *login*. Namun, jika *username* dan *password* sudah benar, maka akan diarahkan ke halaman utama dari admin.

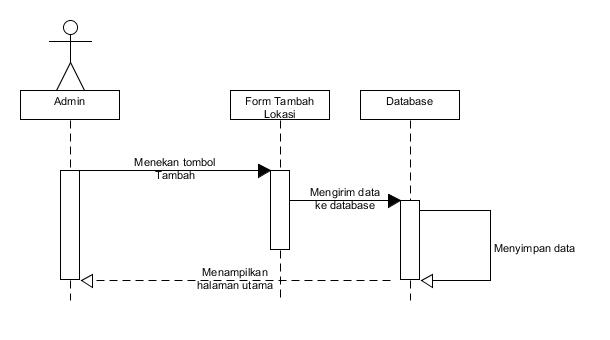
* 1. **Proses Lihat Lokasi**



Gambar 3.15.Proses Lihat lokasi wisata

Pada Gambar 3.15. Sistem Informasi Geografis untuk Pemetaaan Lokasi Wisata Alam terdapat fitur lihat lokasi wisata pada halaman admin, *list* wisata akan tampil setiap admin masuk ke halaman utama admin. Untuk dapat mengakses halaman utama, admin harus *login* terlebih dahulu, kemudian halaman tersebut akan meminta *list* wisata yang terdapat pada database, kemudian database menampilkan seluruh daftar wisata yang telah di publish.

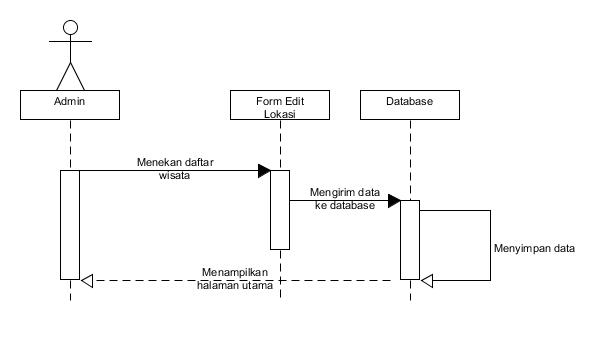
* 1. **Proses Tambah Lokasi**



Gambar 3.16.Proses Menambah lokasi wisata

Pada Gambar 3.16 merupakan proses menambah lokasi wisata oleh admin. Dimana dimulai pada halaman utama, untuk menambahkan lokasi admin harus menekan tombol tambah yang nantinya diarahkan ke sebuah form untuk menambah lokasi. Setelah pengisian selesai dan admin menekan submit, maka data yang sudah diisi sebelumnya dikirimkan ke database, selanjutnya datanya disimpan dan ditampilkan pada halaman utama admin.

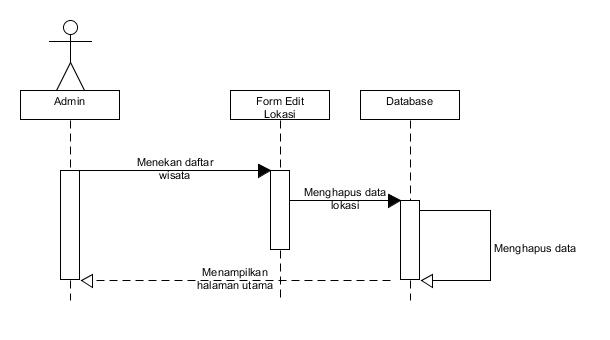
* 1. **Proses Edit Lokasi**



Gambar 3.17.Proses Edit Lokasi Wisata pada *Sequence* Diagram

Pada Gambar 3.17 merupakan proses melakukan *update* atau edit lokasi wisata yang dilakukan oleh admin. Untuk melakukan update admin menekan salah satu dari daftar lokasi wisata yang ada, maka akan diarahkan ke halaman edit lokasi wisata. Setelah update selesai dilakukan, admin dapat menekan tombol submit, maka data yang telah diisi akan dikirimkan ke data dan database akan menyimpan data yang telah dikirim. Selanjutnya data yang telah di edit akan ditampilkan pada halaman utama admin.

* 1. **Proses Hapus Lokasi**



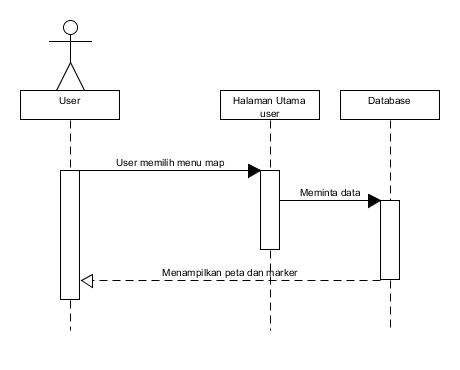
Gambar 3.18.Proses Hapus Lokasi

Pada Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam, terdapat proses hapus lokasi wisata yang dimulai ketika admin menekan daftar wisata. Selanjutnya admin dapat menekan tombol hapus untuk menghapus lokasi wisata. Ketika lokasi wisata berhasil dihapus, maka akan diarahkan ke halaman utama.

**2. Pengguna Wisatawan**

**a. Tampil Map**

Pada Gambar 3.19 merupakan sequence diagram ketika user pertama kali masuk ke aplikasi dan memilih atau menekan *menu* “map”. Maka aplikasi akan menampilkan peta digital dan seluruh lokasi wisata alam yang berada di Pulau Lombok.



Gambar 3.19.Tampil halaman peta

* 1. **Tampil Detail Lokasi**

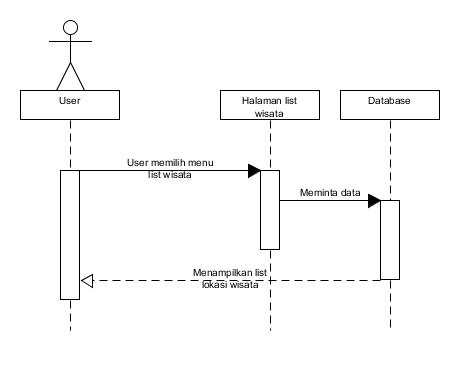
Pada sistem Informasi Geografis terdapat fitur detail lokasi yang sequence diagramnya seperti Gambar 3.20. Dimana untuk melihat halaman detail dari lokasi wisata, user dapat terlebih dahulu memilih lokasi wisata yang ingin dilihat detailnya. Jika sudah, maka user akan diarahkan ke halaman detail lokasi yang akana menampilkan informasi dari lokasi tersebut.



Gambar 3.20. Detail lokasi

**c. Tampil *List* Lokasi**

Pada Gambar 3.21. yang merupakan sequence diagram dari *list* lokasi, user terlebih dahulu harus memilih pada *menu* *list* destinasi. Jika user telah memilih *menu* tersebut, maka akan diarahkan ke halaman *list* destinasi. Alurnya dapat dilihat pada gambar berikut:



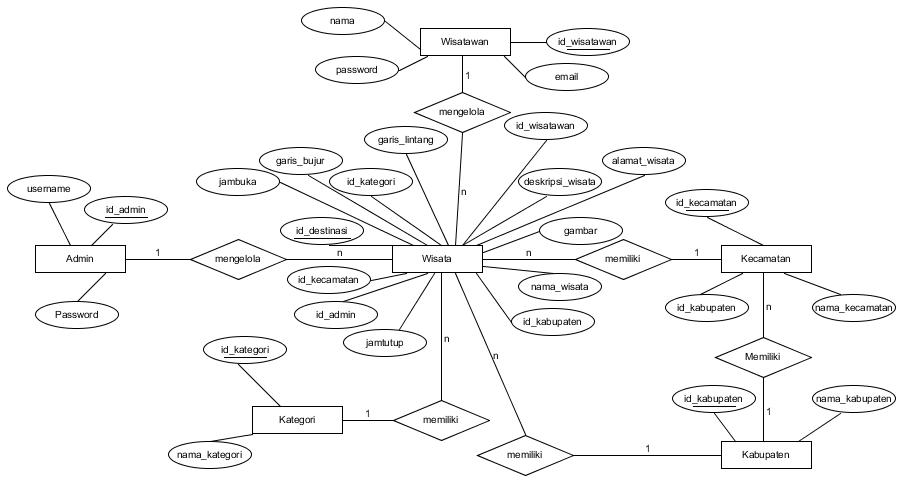
Gambar 3.21. *List* lokasi

* + - 1. **Desain Database**

Berikut adalah beberapa Langkah yang dilakukan dalam melakukan desain database :

* + - 1. **Perancangan Database**

Gambar dibawah ini merupakan Entity Relationship Diagram (ERD) dari Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam, dapat dilihat pada Gambar 3.22 :



Gambar 3.22**.** Entity Relationship Diagram

Pada Gambar 3.22 adalah gambar ERD pada Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam, dimana terdapat 4 entitas yaitu Admin, Kabupaten, Kecamatan dan Wisata. Dimana himpunan entitas saling terhubung dengan himpunan relasi, yang berarti terdapat adanya relasi. Pada setiap entitas memiliki *primary key*, seperti id\_admin untuk entitas admin, id\_kabupaten untuk entitas kabupaten, id\_kecamatan untuk entitas kecamatan dan, id\_wisata untuk entitas kecamatan.

* + - 1. **Struktur Tabel**

Berikut adalah struktur tabel yang ada pada Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam :

Tabel 3.1. Tabel Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama Field | Type | Size | Keterangan |
| 1. | Id\_Admin | Varchar(4) |  | Primary Key pada table admin dengan auto increment |
| 2. | *Username* | Varchar(10) |  |  |
| 3. | *Password* | Varchar(10) |  |  |

Tabel 3.2.Tabel Kabupaten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama Field | Type | Size | Keterangan |
| 1. | Id\_kabupaten | Varchar(5) |  | Primary Key pada table kabupaten |
| 2. | Nama\_kabupaten | Varchar(25) |  |  |

Tabel 3.3.Tabel Kecamatan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama Field | Type | Size | Keterangan |
| 1. | Id\_kecamatan | Varchar(5) |  | Primary Key pada table kecamatan |
| 2. | Id\_kabupaten | Varchar(5) |  | Foreign key |
| 3. | Nama\_kecamatan | Varchar(25) |  |  |

Tabel 3.4.Tabel Destinasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama Field | Type | Size | Keterangan |
| 1. | Id\_Destinasi | Int |  | Primary Key pada table lokasi dengan auto increment |
| 2. | Id\_Admin | Varchar(5) |  | Foreign key |
| 3. | Id\_kecamatan | Varchar(5) |  | Foreign Key |
| 4. | Nama\_wisata | Varchar(25) |  |  |
| 5. | Alamat\_wisata | Varchar(200) |  |  |
| 6. | Gambar | Varchar(100) |  |  |
| 7. | Deskripsi\_wisata | Varchar(250) |  |  |
| 8. | Garis\_lintang | Varchar(30) |  |  |
| 9. | Garis\_bujur | Varchar(30) |  |  |
| 10. | Id\_kategori | Varchar(20) |  | Foreign Key |
| 11. | Id\_Kabupaten | Varchar(20) |  | Foreign Key |

Tabel 3.5.Tabel Kecamatan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama Field | Type | Size | Keterangan |
| 1. | Id\_Kategori | Varchar(5) |  | Primary Key pada table kategori |
| 2. | Nama\_Kategori | Varchar(25) |  |  |

Tabel 3.6.Tabel Wisatawan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama Field | Type | Size | Keterangan |
| 1. | Id\_wisatawan | Varchar(5) |  | Primary Key pada table kategori |
| 2. | email | Varchar(50) |  |  |
| 3. | nama | Varchar(50) |  |  |
| 4. | Password | Varchar(15) |  |  |

### Pembuatan Prototype

Berikut adalah *prototype* dari Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam yang akan dibuat. Terdapat beberapa halaman yang ada pada Sistem Informasi ini, yaitu halaman untuk admin, dan halaman untuk masyarakat. Berikut adalah perancangan mockup/interface dari prototypenya :

* + - 1. **Halaman Login**

Pada Gambar 3.23. Halaman *login* digunakan untuk pengguna mengelola seluruh destinasi wisata yang nantinya akan di publikasikan. Pada halaman *login* ini, terdapat form untuk mengisi *username* dan *password* yang telah terdaftar di dalam database. Jika, *username* dan *password* sudah benar maka akan diarahkan ke halaman dashboard. Namun, jika tidak, akan tetap dihalaman *login* tersebut.



Gambar 3.23.Halaman *login* admin

* + - 1. **Halaman Dashboard**

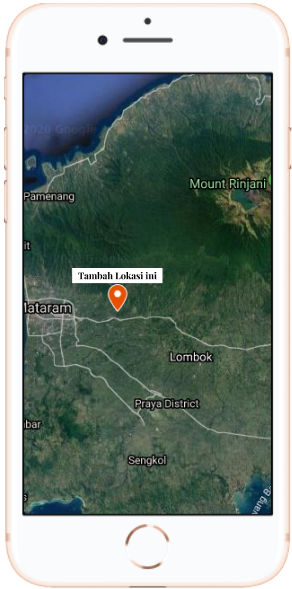
Pada Gambar 3.24.Halaman dashboard admin merupakan halaman yang pertama kali ditampilkan setelah admin melakukan *login*. Pada halaman ini akan menampilkan beberapa fitur seperti akan menampilkan seluruh wisata *list* wisata yang sudah di publikasi. Terdapat fitur filter berdasarkan kabupaten/kecamatan. Selain itu juga, nantinya wisatawan dapat mengetahui berapa banyak daftar wisata pada sebuah kabupaten/kecamatan.



Gambar 3.24.Halaman Dashboard admin

* + - 1. **Halaman Tambah**

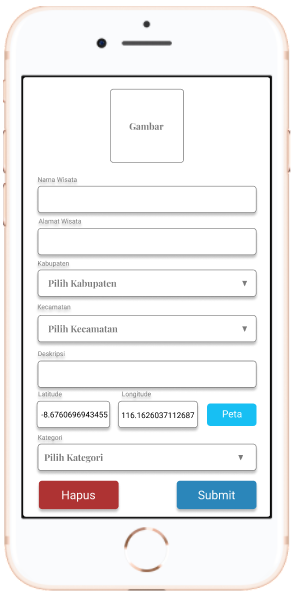
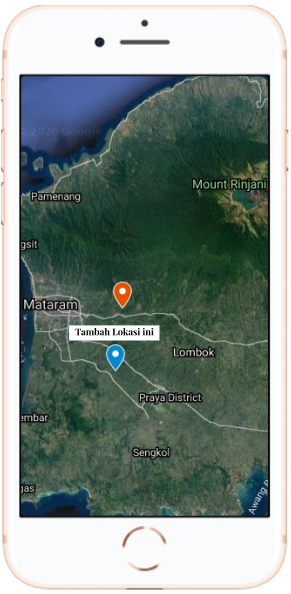
Terdapat 2 langkah dalam melakukan tambah data yaitu Ketika admin meng-klik tombok tambah pada halaman list, maka admin akan diarahkan ke halaman maps tambah, seperti Gambar 3.25. untuk melakukan pin terhadap lokasi yang ingin ditambahkan. Setelah melakukan pin, selanjutnya klik info “Tambah Lokasi ini”. Maka, Admin akan diarahkan ke halaman form tambah yang mana latitude dan longitudenya terisi otomatis setelah meng-klik info “Tambah lokasi ini” seperti Gambar 3.26.

Gambar 3.25.Halaman Maps Tambah Gambar 3.26.Halaman Form Tambah

* + - 1. **Halaman Edit**

Pada halaman Edit admin ini bertujuan bagi admin untuk melakukan perubahan data pada lokasi wisata yang ingin diubah. Berbeda dengan halaman tambah yang akan diberikan maps terlebih dahulu, halaman edit ini akan menampilkan form terlebih dahulu kepada admin. Namun, Ketika admin ingin melakukan perubahan lokasi maka admin dapat meng-klik tombol peta pada Gambar 3.27. Selanjutnya admin akan diarahkan ke halaman Maps Edit dan melakukan perubahan titik lokasi. Pin biru menandakan bahwa lokasi tersebut adalah lokasi baru dan orange lokasi sebelumnya. Ketika admin meng-klik “Tambah Lokasi ini” seperti Gambar 3.28. maka akan diarahkan ke halaman *edit* *form* lagi dan latitude, longitudenya akan berganti dengan latitude, longitude yang baru.

Gambar 3.27.Halaman Form Edit Gambar 3.28.Halaman Maps Edit

* + - 1. **Halaman Dashboard Wisatawan**

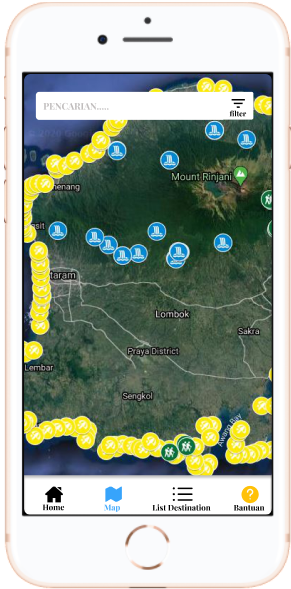
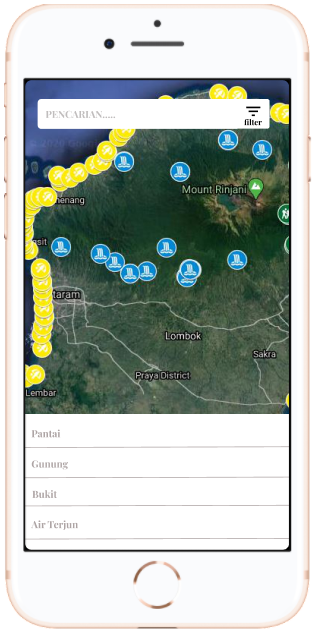
Pada Gambar 3.29. Halaman dashboard. Wisatawan pertama kali akan diberikan tampilan logo dari aplikasi. Halaman ini berguna untuk memberikan informasi ke wisatawan terkait dengan informasi secara singkat seperti logo aplikasi dan nama dari aplikasi tersebut.



Gambar 3.29.*Home* *User*

* + - 1. **Halaman Peta Wisata**

Pada Halaman Peta wisata, aplikasi akan menampilkan seluruh wisata alam yang ada di Pulau Lombok dalam bentuk peta dan pin. Terdapat fitur pencarian yang akan menampilkan saran destinasi yang dituliskan pada kolom pencarian seperti Gambar 3.30, dan fitur filter bagi *user* untuk melihat daftar wisata berdasarkan kategorinya seperti pantai, bukit, gunung, dan air terjun ketika tombol filter di tekan seperti pada Gambar 3.31.

Gambar 3.30.Halaman maps *user*  Gambar 3.31.Fitur Filter Kategori

* + - 1. **Halaman Detail Wisata**

Pada Gambar 3.32 Halaman Detail Wisata berisikan detail dari wisata yang telah dipilih. Dimana terdapat beberapa informasi yang akan ditampilkan pada halaman ini, seperti nama wisata, jalan wista, deskripsi wisata, kategori, dan gambar dari wisata tersebut. Ketika wisatawan menekan tombol lihat di map, maka akan diarahkan ke halaman map dan menampilkan lokasi yang dituju. Berikut adalah tampilannya :



Gambar 3.32.Halaman Detail

* + - 1. **Halaman List Wisata**

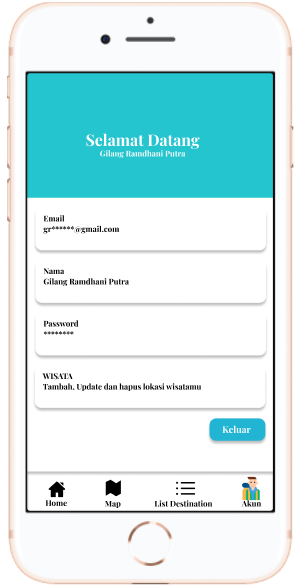
Pada Gambar 3.33. Halaman List Wisata, aplikasi akan menampilkan seluruh wisata alam yang ada di Pulau Lombok dalam bentuk *list*. Terdapat juga fitur pencarian dan filter bagi *user* yang ingin menlihat daftar wista di kabupaten/kecamatan yang terdaftar. Selain itu wisatawan juga dapat melihat total wisata yang ada pada kabupaten/kecamatan yang dipilih.



Gambar 3.33.*List* Wisata *User*

* + - 1. **Halaman Akun**

Pada Gambar 3.34. Halaman Akun ini terdapat beberapa pilihan menu seperti email, nama, password, wisata dan keluar. Dimana fungsi dari menu email/nama/password di klik akan diarahkan untuk mengganti email/nama/password. Sedangkan jika wisata di klik maka akan diarahkan ke halaman tambah data seperti dengan yang dilakukan oleh admin. Terakhir untuk button keluar ada untuk logout dari akun dan diarahkan ke halaman Home.



Gambar 3.34.Halaman Bantuan

### Evaluasi Prototype

Evaluasi *prototype* dilakukan dengan melakukan komunikasi kepada pegawai dinas pariwisata tentang kelayakan dan kesesuaian fungsi-fungsi pada *prototype* yang telah dibuat. Jika pada *prototype* sudah sesuai dengan yang diinginkan *client*. Maka, tahap pengembangan dapat dilanjutkan ke pembuatan sistem yang nyata. Namun, jika belum sesuai maka kembali pada tahap-tahap sebelumnya yaitu analisis kebutuhan dan memulai kembali pembuatan *prototype*.

### Pembuatan Produk Sebenarnya

Setelah dilakukannya tahap evaluasi *prototype*, maka dapat dimulai pengembangan sistem ke tahap pembuatan produk. Dalam tahap ini, digunakan kotlin dan MySQL untuk bahasa pemrogramannya, sedangkan teknologi tambahan yang digunakan adalah *google maps* API. Kotlin adalah bahasa pemrograman yang digunakan untuk pengembangan aplikasi android, bahasa ini merupakan pengembangan dari bahasa java yang lebih mudah digunakan dan dipelajari, sedangkan MySQL digunakan untuk pengelolaan data di database. *Google maps* API digunakan untuk membuat peta, lokasi saat ini, mendapatkan jarak dan waktu ke lokasi yang dituju.

### Pengujian Produk

Setelah sistem yang dibuat telah selesai, maka langkah selanjutnya yang dilakukan adalah dengan menguji sistemnya. Pengujian di lakukan dengan menggunakan *blackbox testing* yang berfungsi untuk menguji sistem yang berjalan. Serta selanjutnya merupakan uji kelayakan oleh pengguna, baik itu pegawai dinas pariwisata atau masyarakat terkait dengan sistem yang telah jadi dengan menggunakan metode *Black Box* ( oleh keryawan dinas pariwisata ) dan *Mean Opinion Score* (MOS) oleh dinas pariwisata dan wisatawan.

### Implementasi Sistem

Langkah terakhir yang dilakukan setelah dilakukan pengujian, selanjutnya pengimplementasian sistem dalam kehidupan sehari-hari. Namun, aplikasi tetap harus ada pendampingan jika terdapat pembaharuan atau pengembangan.

Pada Penelitian ini tidak sampai pada tahap implementasi, melainkan hanya sampai pada tahap pengujian saja.

# BAB IV

# HASIL DAN PEMBAHASAN

## Evaluasi Prototype

Sebelum dilakukan evaluasi, telah dilakukan beberapa tahap seperti pengumpulan kebutuhan, desain sistem, sampai membangun sebuah prototype. Kemudian yang akan dilakukan saat ini adalah evaluasi dari prototype yang telah dibangun dengan melakukan konsultasi ke dinas pariwisata provinsi Nusa Tenggara Barat. Dari melakukan konsultasi di dinas pariwisata, terdapat berbagai masukkan pada kebutuhan sistem yang mempengaruhi prototype yang telah dibuat pada tahap membangun sebuah prototype.

Pada tahap evaluasi prototype ini penulis melakukan kunjungan ke Dinas Pariwisata Provinsi NTB untuk melakukan evaluasi bersama. Dimana penulis mendemokan bentuk aplikasi yang telah di hosting dengan hosting 000webhost dan API (Application Programming Interface) sesuai dengan prototype yang telah dibuat pada tahap membangun sebuah prototype. Mendemokan aplikasi ini dilakukan bersama dengan salah satu pegawai dari Dinas Pariwisata Wisata di bidang destinasi. Fitur yang di demokan antara lain halaman beranda, peta, daftar wisata, detail wisata dan bantuan. Sehingga, terdapat beberapa penambahan kebutuhan pada sistem seperti penambahan kebutuhan informasi jam buka dan jam tutup, perubahan pada kategori gunung menjadi desa wisata, perubahan nama pantai yang sebelumnya pada tahap pengumpulan data bernama pantai kuta menjadi pantai mandalika, dan diberikannya data destinasi wisata alam yang telah terdaftar pada Kawasan Strategis Pariwisata Daerah oleh Dinas Pariwisata. Berikut adalah penjelasan dari setiap kebutuhan yang telah diberikan:

### Pengumpulan Kebutuhan

Terdapat beberapa evaluasi yang dilakukan pada aplikasi sehingga mempengaruhi kebutuhan yang sudah dilakukan sebelumnya. Berikut adalah daftar kebutuhan setelah dilakukan evaluasi :

1. **Jam Buka dan Jam Tutup**

Dilakukan penambahan jam buka dan jam tutup pada halaman tambah data nantinya berguna untuk memberikan informasi kepada wisatawan terkait waktu untuk berkunjung yang tepat ke wisata alam yang dituju. Proses penambahan jam buka dan jam tutup menggunakan plugin date picker, sehingga admin tidak perlu nenambahkan waktu dengan menulis. Namun, jika memang admin ingin menambahkan waktunya dengan menulis sendiri, tidak dipermasalahkan.

1. **Kategori Gunung menjadi Desa Wisata**

Terdapat perubahan kategori wisata yaitu yang semula terdapat destinasi gunung, maka diubah menjadi destinasi Desa Wisata. Tujuan diubahnya menjadi destinasi Desa Wisata untuk mendukung SK Gubernur Nusa Tenggara Barat Tahun 2019 tentang Penetapan 99 Lokasi Desa Wisata di Provinsi Nusa Tenggara Barat Tahun 2019-2023. Namun, nantinya di aplikasi ini hanya menggunakan data Desa Wisata untuk Pulau Lombok saja sebanyak 59 Desa Wisata.

1. **Penambahan Kategori Gili**

Penambahan kategori gili dilakukan karena pada prototype belum terdapat kategori gili.

1. **Perubahan Nama Pantai Kuta menjadi Pantai Mandalika**

Perubahan nama pada data wisata alam yang semulanya pantai kuta menjadi pantai mandalika.

1. **Kategori Bukit diganti dengan kategori Taman Wisata Alam**

Digantinya kategori bukit menjadi kategori Taman Wisata Alam, dikarenakan pada data yang diberikan oleh dinas pariwisata tidak terdapat kategori bukit yang terdaftar di Kawasan Strategis Pariwisata Daerah.

1. **Pencocokan data dari google maps dan Dinas Pariwisata**

Seperti pada penjelasan di latar belakang dan bab 3, bahwa didapatkan daftar wisata alam di google maps sebanyak 140 pantai, 28 air terjun, dan 15 bukit. Namun, setelah dilakukan pencocokan data wisata alam yang ada di Dinas Pariwisata hasilnya hanya 35 destinasi wisata alam yang tervalidasi dan terdaftar pada Kawasan Strategis Pariwisata Daerah ( KPSD ). Selain 35 destinasi wisata alam yang berhasil di validasi, terdapat tambahan wisata alam sebanyak 81 destinasi yang nantinya akan digunakan di aplikasi yang dibuat.

1. **Penambahan Fitur Wisatawan dapat menambah lokasi wisata**

Terdapat evaluasi dari prototype yang telah dikonsultasikan oleh pengguna masyarakat, fitur tambahan tersebut adalah wisatawan dapat menambahkan lokasi wisata alam atau desa wisata, yang nantinya lokasi yang telah ditambahkan akan di validasi terlebih dahulu oleh dinas pariwisata untuk mendapatkan status “sudah tervalidasi” pada wisata tersebut. Fitur ini juga telah dikonsultasikan ke dinas pariwisata dan telah mendapat persetujuan.

### Pembuatan Prototype Hasil Evaluasi

Berikut adalah *prototype* dari Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam yang akan dibuat setelah dilakukan evaluasi. Terdapat beberapa halaman yang mendapat penambahan pada Sistem Informasi ini, yaitu di halaman tambah dan edit untuk admin, halaman detail pada aplikasi masyarakat, fitur wisatawan dapat menambahkan lokasi wisata, dan fitur validasi lokasi oleh dinas pariwisata. Berikut adalah perancangan *mockup/interface* dari prototypenya :

* + - 1. **Halaman Tambah**

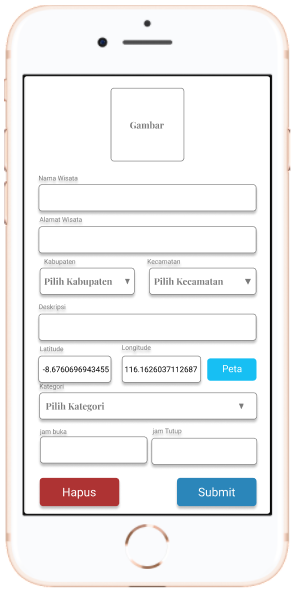
Dapat dilihat pada gambar 4.1. merupakan halaman tambah yang mana terdapat penambahan 2 kolom baru yaitu form jam buka dan jam tutup. Tujuannya untuk memberikan informasi terkait jam buka dan jam tutup pada sebuah lokasi wisata.



Gambar 4.1.Halaman Tambah Setelah Evaluasi

* + - 1. **Halaman Edit**

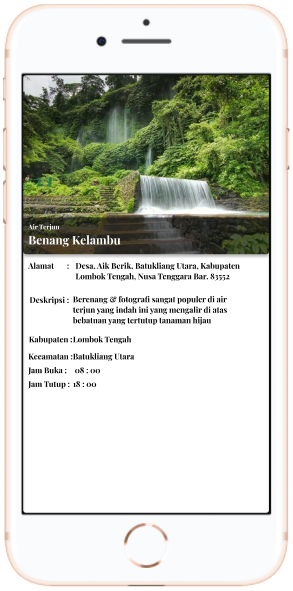
Dapat dilihat pada gambar 4.2. merupakan halaman edit yang mana terdapat penambahan 2 kolom baru yaitu form jam buka dan jam tutup. Tujuannya untuk memberikan informasi terkait jam buka dan jam tutup pada sebuah lokasi wisata.



Gambar 4.2.Halaman Edit Setelah Evaluasi

* + - 1. **Halaman Detail**

Pada Gambar 4.3. Halaman Detail Wisata berisikan detail dari wisata yang telah dipilih. Dimana terdapat beberapa informasi yang akan ditampilkan pada halaman ini, seperti nama wisata, jalan wisata, deskripsi wisata, kategori, gambar, jam buka dan jam tutup dari wisata tersebut. Ketika wisatawan menekan tombol lihat di map, maka akan diarahkan ke halaman map dan menampilkan lokasi yang dituju. Berikut adalah tampilannya :



Gambar 4.3.Halaman Detail Setelah Evaluasi

* + - 1. **Halaman Validasi Wisata**

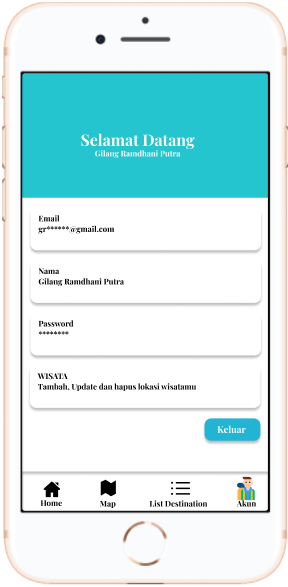
Pada Gambar 4.4. Halaman Validasi Wisata oleh dinas pariwisata ini, ditampilkan daftar destinasi wisata alam yang telah ditambahkan oleh wistawan. Nantinya dinas pariwisata hanya melakukan konfirmasi sebagai tanda bahwa lokasi tersebut telah divalidasi.



Gambar 4.4.Halaman Validasi Wisata

* + - 1. **Halaman Akun Bagi Wisatawan**

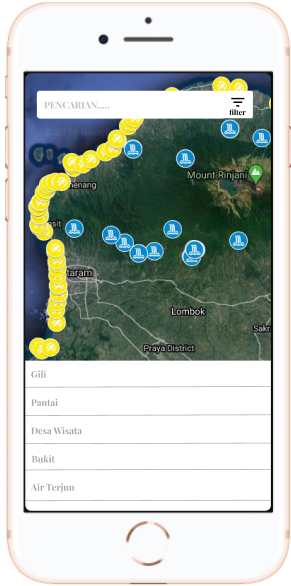
Pada Halaman akun Gambar 4.5., wisatawan dapat melakukan berbagai aktivitas seperti menggubah email, nama, password. Selain itu, wisatawan juga dapat menambahkan lokasi wisata yang belum terdaftar di dinas pariwisata, melakukan *update*, dan *delete* sesuai dengan lokasi yang ditambahkan.



Gambar 4.5.Halaman Akun Wisata

* + - 1. **Halaman Peta**

Pada Halaman Peta wisata Gambar 4.6, aplikasi akan menampilkan seluruh wisata alam yang ada di pulau Lombok dalam bentuk peta dan pin. Terdapat fitur pencarian yang akan menampilkan saran destinasi yang dituliskan pada kolom pencarian, dan fitur filter bagi *user* untuk melihat daftar wisata berdasarkan kategorinya seperti pantai, bukit, desa wisata, dan air terjun ketika tombol filter di tekan.



Gambar 4.6.Halaman Peta Setelah Evaluasi

## Pembuatan Aplikasi Sebenarnya

Setelah lolos evaluasi *prototype*, selanjutnya pembuatan aplikasi dapat mulai dilakukan. Pada tahap ini, akan dilakukan pembuatan aplikasi secara utuh agar aplikasi dapat digunakan pada segi fungsionalitasnya. Beberapa hal yang akan dibahas pada tahap ini, antara lain : pembuatan database, pembuatan API, dan membangun program secara menyeluruh.

### Pembuatan database

Gambar dibawah ini merupakan Struktur database dari Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam setelah terdapat tambahan kebutuhan dalam sistem, dapat dilihat pada Gambar 4.7 :



Gambar 4.7.Struktur Database

Pada Gambar 4.7. merupakan struktur database pada aplikasi Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam setelah dilakukan evaluasi, dimana terdapat tabel admin, destination, kabupaten, kategori dan kecamatan. Dimana setiap tabelnya akan saling terhubung satu dengan yang lain.

Berikut adalah struktur tabel yang ada pada Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam :



Gambar 4.8.Tabel Admin

Gambar 4.8. merupakan tabel admin pada database “siwita”. Terdapat field id\_admin, username dari admin, dan password admin yang nantinya akan digunakan untuk login ke aplikasi adminnya.



Gambar 4.9.Tabel Kabupaten

Gambar 4.9. merupakan struktur tabel dari tabel kabupaten. Terdapat field id\_kabupaten dan name\_kabupaten dengan tipe data varchar 4 dan 20. Dapat dilihat pada tabel bahwa id\_kabupaten merupakan primary key dari tabel kabupaten tersebut.



Gambar 4.10.Tabel Kecamatan

Gambar 4.10. merupakan struktur tabel dari tabel kecamatan. Terdapat field id\_kecamatan, name\_kecamatan dan id\_kabupaten dengan tipe data varchar 4,20, dan 4. Dapat dilihat juga pada tabel bahwa id\_kecamatan merupakan primary key dari tabel kecamatan dan id\_kabupaten merupakan foreign key dari tabel kabupaten.



Gambar 4.11.Tabel Destinasi

Pada gambar 4.11. yang merupakan tabel destinasi, dapat dilihat terdapat beberapa field seperti id\_destination dengan tipe int yang merupakan primary key dan auto increment, name\_destination, lat\_destination, lng\_destination, address\_destination, desc\_destination, id\_kecamatan, img\_destination, id\_kategori, id\_kabupaten, jambuka, dan jamtutup memiliki tipe data varchar. Seperti yang dapat dilihat pada tabel 4.4. terdapat id\_kecamatan yang merupakan foreign key dari tabel kecamatan, id\_kategori dari tabel kategori, dan id\_kabupaten dari tabel kabupaten.



Gambar 4.12.Tabel Kategori

Selanjutnya gambar 4.12. yang merupakan tabel dari kategori yang memiliki id\_kategori dan name\_kategori, keduanya memiliki tipe data varchar dan id\_kategori merupakan primary key dari tabel kategori tersebut.

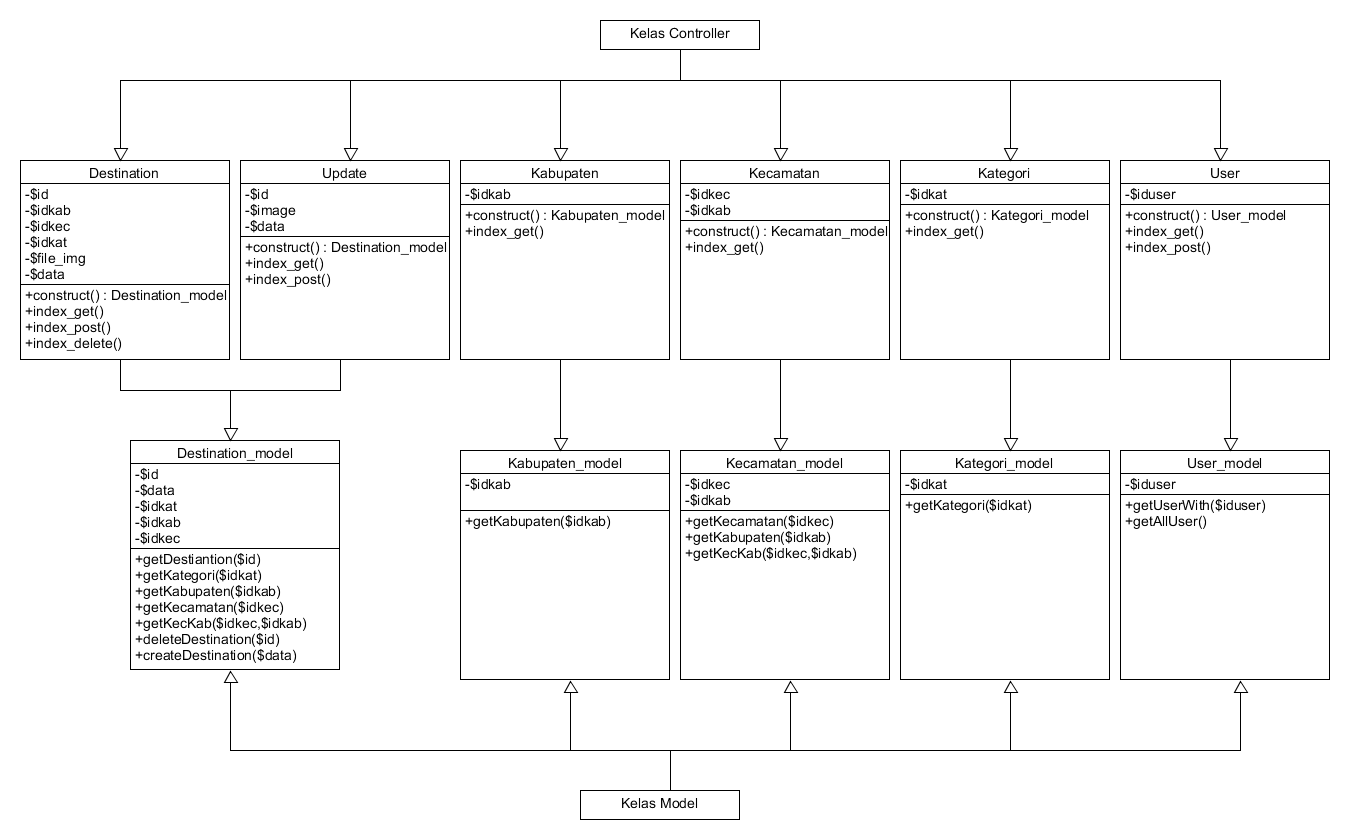


Gambar 4.13.Tabel Wisatawan

Selanjutnya gambar 4.13. yang merupakan tabel dari wisatawan yang memiliki id\_wisatawan, email, nama, dan password. Email, nama dan password memiliki tipe data varchar dan id\_wisatawan merupakan primary key dari tabel wisatawan dengan tipe data Int dan *auto increment*.

### Pembuatan API

Langkah selanjutnya adalah pembuatan API menggunakan Codeigniter yang nantinya berguna sebagai penghubung antara database dengan aplikasi android. Berikut adalah class diagram dari pembuatan API Sistem Informasi Pemetaan Lokasi Wisata Alama berbasis mobile :



Gambar 4.14.Kelas Diagram API

Pada Gambar 4.14. yang merupakan kelas diagram dari API pada Sistem Informasi Geografis Pemetaan Lokasi Wisata Alam. Dapat dilihat pada gambar bahwa terdapat 2 kelas utama pada pembuatan API kali ini yaitu Kelas Model dan Kelas Controller. Pada Kelas Controller terdapat beberapa kelas yaitu Destination, Update, Kabupaten, Kecamatan dan User. Sedangkan pada Kelas Model yaitu Destination\_model, Kabupaten\_model, Kecamatan\_model, Kategori\_model dan User\_model.

* + - 1. **Api Destinasi**

Berikut adalah cara untuk membuat API destinasi, yang mana pada api destinasi ini digunakan Controller dan Model dalam pembuatannya :

* + - * 1. **Controller Destinasi**

Destinasi ini merupakan sebuah kelas yang meng-extends Controller sehingga nantinya controller ini dapat terhubung ke Destinasi\_model. Tujuan dihubungkan ke Destinasi\_model agar dapat mengambil data pada database.

* + - 1. **Fungsi Get**

|  |  |
| --- | --- |
| public function index\_get()  {  $id = $this->get('id\_destination');  $idkat = $this->get('id\_kategori');  $idkab = $this->get('id\_kabupaten');  $idkec = $this->get('id\_kecamatan');  if($id !== null){  $destination = $this->destination->getDestination($id);  }else if($idkat !==null){  $destination = $this->destination->getKategori($idkat);  }else if($idkab !==null){  $destination = $this->destination->getKabupaten($idkab);  }else if($idkec !==null){ | $destination = $this->destination>getKecamatan($idkec);  }else if($idkab && $idkec !==null){  $destination = $this->destination->getKecKab($id, $idkec);  }else{  $destination = $this->destination->getDestination();  }  if($destination){  $this->response([  'status' => true,  'data' => $destination  ], REST\_Controller::HTTP\_OK);  }else{  $this->response([  'status' => false,  'message' => 'id tidak ditemukan'  ], REST\_Controller::HTTP\_NOT\_FOUND);  } } |

*Source code* 4.1. Fungsi get destinasi

Pada *source code* 4.1. merupakan fungsi get dalama pembuatan api pada tabel destinasi. Dimana dapat dilihat dalam funsinya terdapat variable $id, $idkat, $idkab, $idkec yang menyimpan nilai id\_destination, id\_kategori, id\_kabupaten dan id\_kecamatan. Kegunaan dari id-id tersebut untuk nantinya digunakan Ketika ingin melakukan filter get berdasarkan destinasi, kategori, kabupaten dan kecamatan pada tabel destination. Dapat dilihat juga kondisi Ketika setiap idnya tidak sama dengan null, maka datanya akan ditampilkan sesuai dengan id yang ditulis pada API. Jika tidak ada id yang ditulis pada idnya maka datanya dari destinasi akan ditampilkan semuanya.

* + - 1. **Fungsi Hapus**

|  |  |
| --- | --- |
| public function index\_delete(){  $id = $this->delete('id\_destination');  if($id ===null){  $this->response([  'status' => false,  'message' => 'Provide an id'  ], REST\_Controller::HTTP\_BAD\_REQUEST);  }else{  if($this->destination->deleteDestination($id)>0){ | //ok  $this->response([  'status' => true,  'data' => $id,  'message' => 'berhasil di delete'  ], REST\_Controller::HTTP\_NO\_CONTENT);  }else{  //id not found  $this->response([  'status' => false,  'message' => 'id not found'  ], REST\_Controller::HTTP\_BAD\_REQUEST);  }  } } |

*Source code* 4.2. Fungsi Hapus destinasi

Pada *source code* 4.2. merupakan fungsi hapus dalam pembuatan API pada tabel destinasi. Dimana dapat dilihat dalam funsinya terdapat variable $id yang menyimpan nilai id\_destination. Kegunaan dari id tersebut untuk nantinya digunakan Ketika ingin melakukan hapus data berdasarkan id\_destination pada tabel destination. Dapat dilihat juga kondisi Ketika setiap idnya sama dengan kosong maka di API-nya akan menampilkan status dan messagenya. Jika terdapat id yang ditulis maka datanya dari destinasi akan dihapus berdasarkan id yang dipilih.

* + - 1. **Fungsi Tambah**

|  |  |  |
| --- | --- | --- |
| public function index\_post()  {  if($\_FILES["img\_destination"]['name']=='')  {  $file\_img = " ";  }  else  {  $file\_img = time().$\_FILES["img\_destination"]['name'];  $config['upload\_path'] = "./assets/foto/";  $config['allowed\_types'] = 'gif|jpg|png|jpeg';  $config['file\_name'] = $file\_img;  $this->load->library('upload', $config);  if ( ! $this->upload->do\_upload('img\_destination')) {  $this->upload->display\_errors();  }  else  {  $coba = $this->upload->data();  } | $data = [  'name\_destination' => $this->post('name\_destination'),  'lat\_destination' => $this->post('lat\_destination'),  'lng\_destination' => $this->post('lng\_destination'),  'address\_destination' => $this->post('address\_destination'),  'desc\_destination' => $this->post('desc\_destination'),  'id\_kecamatan' => $this->post('id\_kecamatan'),  'img\_destination' => $file\_img,  'id\_kategori' => $this->post('id\_kategori'),  'id\_kabupaten' => $this->post('id\_kabupaten'),  'jambuka' => $this->post('jambuka'),  'jamtutup' => $this->post('jamtutup'),  'id\_admin' => $this->post('id\_admin'),  'status' => $this->post('status'),  'id\_wisatawan' => $this->post('id\_wisatawan')    ]; | if($this->destination->createDestination($data) > 0){  $this->response([  'status' => true,  'message' => 'berhasil di ditambahkan'  ], REST\_Controller::HTTP\_CREATED);  }else{  //id not found  $this->response([  'status' => false,  'message' => 'gagal menambah data'  ], REST\_Controller::  HTTP\_BAD\_REQUEST);  }  }  } |

*Source code* 4.3. Fungsi Tambah destinasi

Pada S*ource code* 4.3. merupakan fungsi tambah destinasi pada controller Destinasi. Seperti yang dapat dilihat terdapat $\_FILES yang merupakan cara untuk melakukan input berupa file atau gambar dengan mengambil nama field dari tabel destinasi yaitu img\_destination. Dapat dilihat juga dibuat sebuah array yang akan menampung semua data yang diinputkan oleh admin/wisatawan.

* + - 1. **Destinasi\_model**

Destination model merupakan sebuah kelas yang mengextends dari CI\_Model, dimana kegunaan dari destinasi\_model ini adalah untuk melakukan komunikasi dengan database.

* + - 1. **getDestination**

|  |  |
| --- | --- |
| public function getDestination($id = null){  if($id === null){  return $this->db->get('destination')->result\_array(); | }else{  return $this->db->get\_where('destination', ['id\_destination' => $id])->result\_array();  }  } |

*Source code* 4.4. fungsi getDestiantion

Fungsi getDestination digunakan untuk mengambil seluruh atau berdasarkan id pada database. Dapat dilihat jika $id sama dengan kosong, maka semua data pada tabel destination akan ditampilkan, namun jika tidak maka ditampilkan berdasarkan id\_destinationnya.

* + - 1. **deleteDestination**

|  |
| --- |
| public function deleteDestination($id){  $this->db->delete('destination', ['id\_destination' => $id]);  return $this->db->affected\_rows();  } |

*Source code* 4.5. fungsi deleteDestination

Fungsi deleteDestination digunakan untuk menghapus data pada tabel destination berdasarkan idnya.

1. **createDestination**

|  |
| --- |
| public function createDestination($data){  $this->db->insert('destination', $data);  return $this->db->affected\_rows();  } |

*Source code* 4.6. fungsi createDestination

Fungsi createDestination digunakan untuk melakukan penambahan data pada tabel destination, query yang digunakana adalah insert yang parameternya berisikan nama tabel dan data yang akan di-input-kan.

1. **updateDestination**

|  |
| --- |
| public function updateDestination($data, $id){  $this->db->update('destination', $data, ['id\_destination' => $id]);  return $this->db->affected\_rows();  } |

*Source code* 4.7. Fungsi updateDestination

Fungsi updateDestination digunakan untuk melakukan perubahan data pada tabel destination, dapat dilihat query yang digunakan adalah update yang parameternya tabel destination, data dan id yang akan dilakukan perubahan data.

* + - 1. **API Kategori**

Berikut adalah cara untuk membuat API kategori, yang mana pada API destinasi ini digunakan Controller dan Model dalam pembuatannya :

* + - * 1. **Controller Kategori**

Kategori ini merupakan sebuah kelas yang meng-extends Controller sehingga nantinya controller ini dapat terhubung ke Kategori\_model. Tujuan dihubungkan ke Kategori\_model agar dapat mengambil data pada database.

|  |  |
| --- | --- |
| public function index\_get()  {  $id = $this->get('id\_kategori');  if($id === null){  $kategori = $this->kategori->getKategori();  }else{  $kategori = $this->kategori->getKategori($id);  } | if($kategori){  $this->response([  'status' => true,  'data' => $kategori  ], REST\_Controller::HTTP\_OK);  }else{  $this->response([  'status' => false,  'message' => 'id kategori tidak ditemukan'  ], REST\_Controller::HTTP\_NOT\_FOUND); }} |

*Source code* 4.8. Fungsi Get Kategori

Pada Controller untuk API Kategori hanya terdapat fungsi get kategori, karena pada API kategori hanya untuk mengambil kategori yang sudah ada pada database, tidak untuk menambah, update bahkan menghapus. Dapat dilihat pada s*ource code* 4.5. terdapat $id yang berguna untuk menyimpan get id\_kategori. Kemudian dilakukan pengencekan, jika pada APInya tidak terdapat id, maka seluruh kategori akan ditampilkan dengan menghubungkannya ke fungsi getKategori pada model. Namun, jika $idnya tidak sama dengan null maka ditampilkan kategori berdasarkan id yang ditulis dan memanggil fungsi getKategori($id) pada model Kategori.

* + - * 1. **Kategori\_model**

|  |
| --- |
| public function getKategori($id = null){  if($id === null){  return $this->db->get('kategori')->result\_array();  }else{  return $this->db->get\_where('kategori', ['id\_kategori' => $id])->result\_array(); }  } |

*Source code* 4.9. Model kategori

Pada S*ource code* 4.6. yang merupakan model dari API Kategori. Dapat dilihat terdapat fungsi getKategori yang parameternya berisikan $id = null. getKategori() ini nantinya digunakan oleh controller Kategori untuk digunakan dalam memanggil data pada database. Seperti pada sourcenya jika $idnya null, maka memanggil fungsi get pada tabel kategori, kemudian ditampilkan dalam bentuk array. Selanjutnya jika tidak kosong, makan memangil fungsi get\_where pada tabel kategori dan id\_kategori berdasarkan id yang dimasukkan, kemudian ditampilkan datanya dalam bentuk array.

* + - 1. **Api Kabupaten**

Berikut adalah cara untuk membuat API kabupaten, yang mana pada API kabupaten ini digunakan Controller dan Model dalam pembuatannya :

* + - * 1. **Controller Kabupaten**

Kabupaten ini merupakan sebuah kelas yang meng-extends Controller sehingga nantinya controller ini dapat terhubung ke Kabupaten \_model. Tujuan dihubungkan ke Kabupaten\_model agar dapat mengambil data pada database.

|  |  |
| --- | --- |
| public function index\_get()  {  $id = $this->get('id\_kabupaten');  if($id === null){  $kabupaten = $this->kabupaten->getKabupaten();  }else{  $kabupaten = $this->kabupaten->getKabupaten($id);  } | if($kabupaten){  $this->response([  'status' => true,  'data' => $kabupaten  ], REST\_Controller::HTTP\_OK);  }else{  $this->response([  'status' => false,  'message' => 'id kabupaten tidak ditemukan'  ], REST\_Controller::HTTP\_NOT\_FOUND);  } } |

*Source code* 4.10. Controller kabupaten

Pada Controller untuk API Kabupaten hanya terdapat fungsi get kabupaten, karena pada API kabupaten hanya untuk mengambil kabupaten yang sudah ada pada database, tidak untuk menambah, update bahkan menghapus. Dapat dilihat pada *source code* 4.10. terdapat $id yang berguna untuk menyimpan get id\_kabupaten. Kemudian dilakukan pengencekan, jika pada APInya tidak terdapat id, maka seluruh kabupaten akan ditampilkan dengan menghubungkannya ke fungsi getKabupaten pada model. Namun, jika $idnya tidak sama dengan null maka ditampilkan kategori berdasarkan id yang ditulis dan memanggil fungsi getKabupaten($id) pada model Kabupaten.

* + - * 1. **Kabupaten\_model**

|  |
| --- |
| public function getKabupaten($id = null){  if($id === null){  return $this->db->get('kabupaten')->result\_array();  }else{  return $this->db->get\_where('kabupaten', ['id\_kabupaten' => $id])->result\_array();  } } |

*Source code* 4.11. Model kabupaten

Pada S*ource code* 4.11. yang merupakan model dari API Kabupaten. Dapat dilihat terdapat fungsi getKategori yang parameternya berisikan $id = null. getKabupaten() ini nantinya digunakan oleh controller Kategori untuk digunakan dalam memanggil data pada database. Seperti pada sourcenya jika $idnya null, maka memanggil fungsi get pada tabel kabupaten, kemudian ditampilkan dalam bentuk array. Selanjutnya jika tidak kosong, makan memangil fungsi get\_where pada tabel kabupaten dan id\_kabupaten berdasarkan id yang dimasukkan, kemudian ditampilkan datanya dalam bentuk array.

* + - 1. **API Kecamatan**

Berikut adalah cara untuk membuat API kecamatan, yang mana pada API kecamatan ini digunakan Controller dan Model dalam pembuatannya :

* + - * 1. **Controller Kecamatan**

Kecamatan ini merupakan sebuah kelas yang meng-extends Controller sehingga nantinya controller ini dapat terhubung ke Kecamatan \_model. Tujuan dihubungkan ke Kecamatan \_model agar dapat mengambil data pada database.

|  |  |
| --- | --- |
| public function index\_get()  {  $id = $this->get('id\_kabupaten');  $idkec = $this->get('name\_kecamatan');  if($id !== null){  $kecamatan = $this->kecamatan->getKabupaten($id);  }else if ($idkec !==null){  $kecamatan = $this->kecamatan->getKecamatan($idkec);  }else if($id && $idkec !==null){  $destination = $this->destination->getKecKab($id, $idkec); | }else{  $kecamatan = $this->kecamatan->getKecamatan();  } if($kecamatan){  $this->response([  'status' => true,  'data' => $kecamatan  ], REST\_Controller::HTTP\_OK);  }else{$this->response([  'status' => false,  'message' => 'id id\_kecamatan tidak ditemukan'  ], REST\_Controller::HTTP\_NOT\_FOUND);} } |

*Source code* 4.12. Controller kecamatan

Pada Controller untuk API Kecamatan hanya terdapat fungsi get kecamatan, karena pada API kabupaten hanya untuk mengambil kecamatan yang sudah ada pada database, tidak untuk menambah, update bahkan menghapus. Dapat dilihat pada *source code* 4.12. terdapat $id yang berguna untuk menyimpan get id\_kabupaten, dan $idkec berguna untuk menyimpan get id\_ id\_kecamatan. Kemudian dilakukan pengencekan, jika pada API id untuk kabupaten tidak sama dengan null, maka seluruh kabupaten akan ditampilkan berdasarkan id dari kabupaten dan memanggil getKabupaten($id) dari Model Kecamatan. Begitu juga dengan id\_kecamatan, jika $idkec tidak sama dengan null maka menampilkan daftar kecamatan dengan memanggil menggunakan method getKecamatan($id) dari Model Kecamatan. Namun, jika $id dan $idkecnya sama dengan null maka semua list kecamatan akan ditampilkan.

* + - * 1. **Kecamatan\_model**

|  |  |
| --- | --- |
| public function getKabupaten($id = null){  if($id === null){  return $this->db->get('kecamatan')->result\_array();  }else{  return $this->db->get\_where('kecamatan', ['id\_kabupaten' => $id])->result\_array();  }  } | public function getKecamatan($id = null){  if($id === null){  return $this->db->get('kecamatan')->result\_array();  }else{  return $this->db->get\_where('kecamatan', ['id\_kecamatan' => $id])->result\_array();  }  } |

*Source code* 4.13. Model kecamatan

Pada *source code* 4.13. yang merupakan model dari API Kecamatan. Dapat dilihat terdapat fungsi getKategori yang parameternya berisikan $id = null. getKecamatan() ini nantinya digunakan oleh controller Kecamatan untuk digunakan dalam memanggil data pada database. Seperti pada sourcenya jika $idnya null, maka memanggil fungsi get pada tabel kecamatan, kemudian ditampilkan dalam bentuk array. Selanjutnya jika tidak kosong, makan memangil fungsi get\_where pada tabel kecamatan dan id\_kecamatan berdasarkan id yang dimasukkan, kemudian ditampilkan datanya dalam bentuk array.

* + - 1. **API** **User**

Berikut adalah cara untuk membuat API admin, yang mana pada API admin ini digunakan Controller dan Model dalam pembuatannya :

* + - * 1. **Controller User**

User ini merupakan sebuah kelas yang meng-extends Controller sehingga nantinya controller ini dapat terhubung ke User\_model. Tujuan dihubungkan ke User\_model agar dapat mengambil data pada database.

**Fungsi Get**

|  |  |
| --- | --- |
| public function index\_get(){  $id = $this->get('id\_admin');  if($id === null){  $user = $this->user->getAllUser();  }else{  $user = $this->user->getUserWith($id);  } | if($user){  $this->response([  'status' => true,  'data' => $user  ], 200);  }else{  $this->response([  'status' => false,  'message' => 'Oops! Data tidak ditemukan :('], 404);} } |

*Source code* 4.14. Fungsi get user

Pada Controller untuk API User terdapat fungsi get yang berguna untuk mengambil data keseluruhan atau berdasarkan id\_admin-nya. Dapat dilihat pada *source code* 4.14. terdapat $id yang berguna untuk menyimpan get id\_admin. Kemudian dilakukan pengencekan, jika pada APInya tidak terdapat id, maka seluruh user akan ditampilkan dengan menghubungkannya ke fungsi getAllUser pada model. Namun, jika $idnya tidak sama dengan null maka ditampilkan kategori berdasarkan id yang ditulis dan memanggil fungsi getUserWith($id) pada model User.

**Fungsi Post**

|  |  |
| --- | --- |
| public function index\_post(){  $user = $this->user->getAllUser();    $username = $this->post('username');  $password = $this->post('password');  foreach ($user as $data) { | if($data['username'] === $username  && $data['password'] === $password  ){    $tmp\_user = $this->user->getUserWith($data['id\_admin']);  $this->response([  'status' => true,  'token' => $tmp\_user  ], 200);  }  }  } |

*Source code* 4.15. Fungsi post user

Pada Controller untuk API User terdapat fungsi post yang berguna untuk pos data ke tabel Admin. Dapat dilihat pada *source code* 4.15. terdapat $user mengambil semua akun yang terdapat pada tabel Admin di database. Kemudian $username untuk menampung input-kan username, dan password untuk menampung password yang di input-kan. Dilakukan perulangan untuk mengecek apakah data yang ada pada database sama dengan data yang di input-kan user, maka user dapat login dan semua data terkait dengan akun tersebut akan diambil berdasarkan idnya.

* + - * 1. **User\_model**

|  |  |
| --- | --- |
| public function getAllUser(){  $query = $this->db->query('SELECT \* FROM admin ');  return $query->result\_array();  } | public function getUserWith($id){  if($id === null){  return $this->db->get('admin')->result\_array();  }else{  return $this->db->get\_where('admin', ['id\_admin' => $id] )->result\_array();  } } |

*Source code* 4.16. Model User

Pada *source code* 4.16. terdapat 2 fungsi yang nantinya dapat digunakan pada controller User yaitu getAllUser dan getUserWith($id). getAllUser berguna untuk mengambil data yang ada pada tabel admin di database. Kemudian untuk getUserWith($id) mengambil data pada tabel admin berdasarkan id\_adminya.

* + - 1. **API Wisatawan**

Berikut adalah cara untuk membuat API wisatawan, yang mana pada API wisatawan ini digunakan Controller dalam pembuatannya :

* + - * 1. **Controller Wisatawan**

Kecamatan ini merupakan sebuah kelas yang meng-extends Controller sehingga nantinya controller ini dapat terhubung ke User\_model. Tujuan dihubungkan ke User \_model agar dapat mengambil data pada database.

* + - 1. **Fungsi Get**

|  |  |
| --- | --- |
| public function index\_get(){  $id = $this->get('id\_wisatawan');  if($id === null){  $user = $this->user->getAllWisatawan();  }else{  $user = $this->user->getWisatawanWith($id);  } | if($user){  $this->response([  'status' => true,  'data' => $user  ], 200);  }else{  $this->response([  'status' => false,  'message' => 'Oops! Data tidak ditemukan :('  ], 404);}} |

*Source code* 4.17. Fungsi get wistawan

Pada Controller untuk API Wisatawan terdapat fungsi get yang berguna untuk mengambil data keseluruhan atau berdasarkan id\_wisatawan-nya. Dapat dilihat pada s*ource code* 4.17. terdapat $id yang berguna untuk menyimpan get id\_wisatawan. Kemudian dilakukan pengencekan, jika pada APInya tidak terdapat id, maka seluruh user akan ditampilkan dengan menghubungkannya ke fungsi getAllWisatawan pada model. Namun, jika $idnya tidak sama dengan null maka ditampilkan kategori berdasarkan id yang ditulis dan memanggil fungsi getWisatawanWith($id) pada model User.

* + - 1. **Fungsi Post**

|  |  |
| --- | --- |
| public function index\_post(){          $pengguna = $this->user->getAllWisatawan();            $username = $this->post('email');          $password= $this->post('password'); | foreach ($pengguna as $data) {            if($data['email'] === $username               && $data['password'] === $password             ){          $tmp\_user = $this->user->getWisatawanWith($data['id\_wisatawan']);              $this->response([                  'status' => true,                  'data' => $data              ], 200);}}} |

*Source code* 4.18. Fungsi post wistawan

Pada Controller untuk API User terdapat fungsi post yang berguna untuk pos data ke tabel Wisatawan. Dapat dilihat pada *source code* 4.18. terdapat $user mengambil semua akun yang terdapat pada tabel Wisatawan di database. Kemudian $email untuk menampung input-kan email, dan password untuk menampung password yang di input-kan. Dilakukan perulangan untuk mengecek apakah data yang ada pada database sama dengan data yang di input-kan wisatawan, maka user dapat login dan semua data terkait dengan akun tersebut akan diambil berdasarkan idnya.

* + - 1. **Fungsi Put**

|  |  |
| --- | --- |
| public function index\_put(){          $id = $this->put('id\_wisatawan');          $data = [              'email' => $this->put('email'),              'nama' => $this->put('nama'),              'password' => $this->put('password'),          ]; | if($this->user->updateWisatawan($data, $id) > 0){              $this->response([                  'status' => true,                  'message' => 'berhasil di update'              ], REST\_Controller::HTTP\_NO\_CONTENT);          }else{              //id not found              $this->response([                  'status' => false,                  'message' => 'gagal update'              ], REST\_Controller::HTTP\_BAD\_REQUEST);          }      } |

*Source code* 4.19. Fungsi put wistawan

Pada s*ource code* 4.19 merupakan fungsi untuk melakukan update pada data wisatawan. Dapat dilihat terdapat id\_wisatawan, email, nama dan password dengan tipe methodnya adalah put yaitu untuk melakukan update. Selanjutnya datanya akan dikirim ke method updateWisatawan dengan parameter $data dan $id ke User\_model untuk dilakukan update di database.

* + - 1. **API** **Registration**

Berikut adalah cara untuk membuat API registration, yang mana pada API registration ini digunakan Controller dalam pembuatannya :

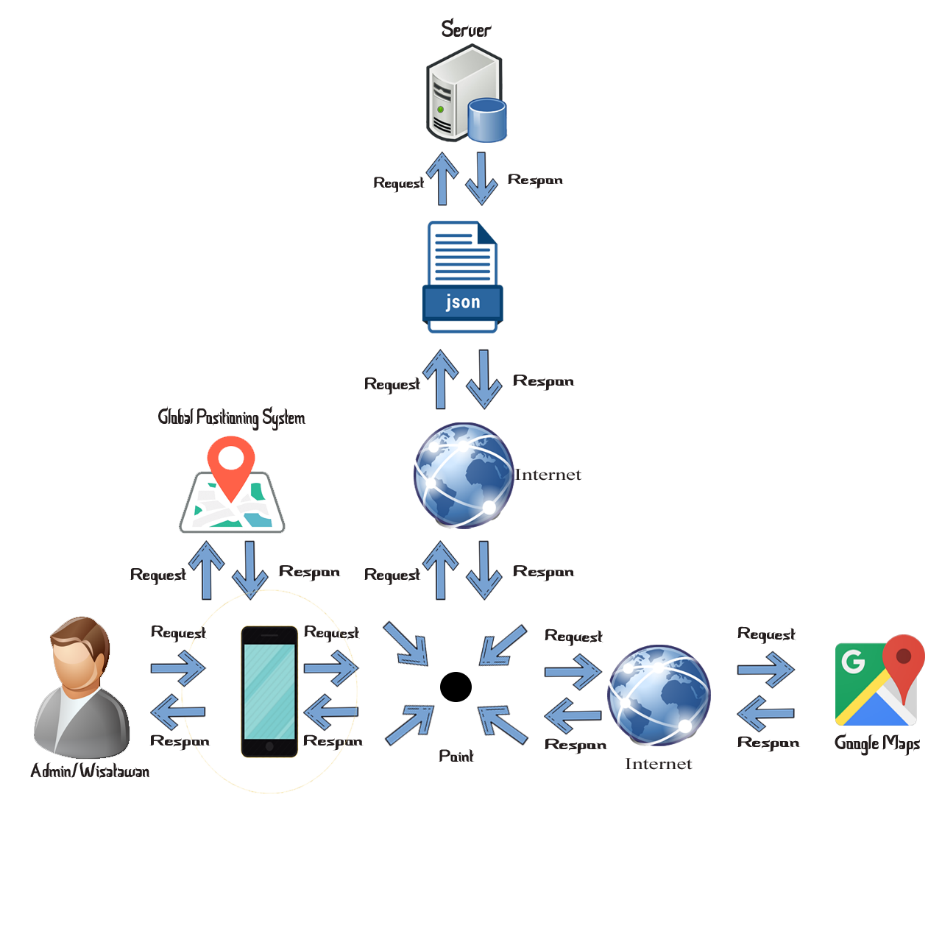
* + - * 1. **Controller Registration**

|  |  |
| --- | --- |
| public function index\_post(){            $email = $this->post('email');          $nama= $this->post('nama');          $password= $this->post('password');          $exists = $this->user->cekemail($email);          $count = count($exists);            if(empty($count)){              $databaru = [                  'email' => $email,                  'nama' => $nama,                  'password' => $password,              ];              $tambah = $this->user->createNewUser($databaru); | if($tambah > 0){                  $this->response([                      'status' => true,                      'message' => 'berhasil di ditambahkan'                  ], REST\_Controller::HTTP\_CREATED);              }else{                  //id not found                  $this->response([                      'status' => false,                      'message' => 'gagal menambah data'                  ], REST\_Controller::HTTP\_BAD\_REQUEST);              }            }else{              $this->response([                  'status' => false,                  'message' => 'Email telah terdaftar'                  ], REST\_Controller::HTTP\_BAD\_REQUEST);            }  } |

*Source code* 4.20. Fungsi post registration

Fungsi dari controller registration ini adalah untuk melakukan pendaftaran bagi wisatawan yang ingin mendaftar di aplikasi. Seperti pada s*ource code* 4. 20. dapat dilihat email terlebih dahulu di cek dengan menggunakan method cekemail($email) yang ada pada User\_model. Kemudian cek apakah emailnya sudah terdaftar atau tidak. Jika email sudah terdaftar, maka data yang diinputkan tidak dapat di kirim ke database. Sedangkan, jika emailnya belum terdaftar, maka field email, nama dan password akan terisi sesuai dengan inputan wisatawan.

### Arsitektur Aplikasi setelah diimplementasi

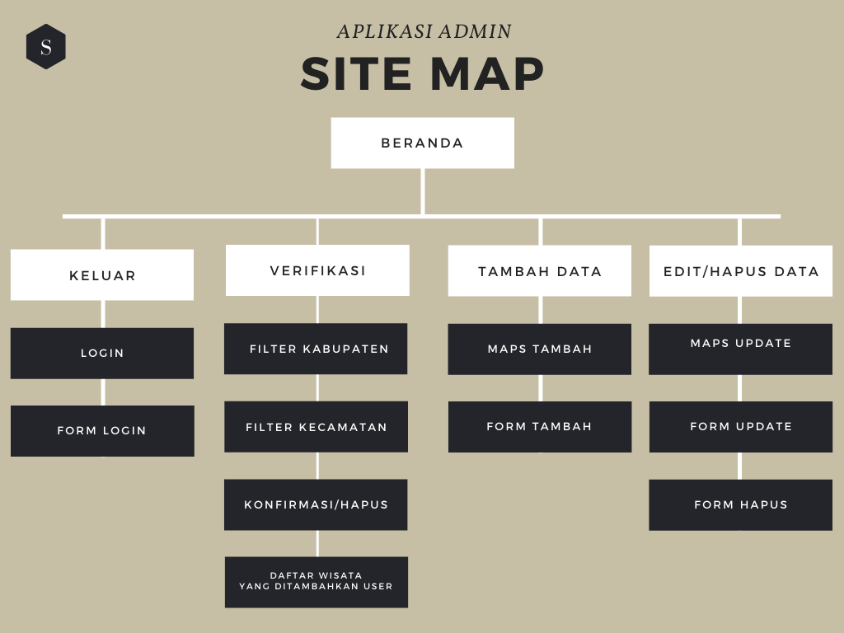
****

Gambar 4.15.Arsitektur Aplikasi

Arsitektur Aplikasi adalah cara untuk mendeskripsikan teknologi apa saja yang digunakan pada pembuatan aplikasi. Seperti yang dapat dilihat pada gambar 4.15. terdapat penggunaan google maps, GPS, android, internet, json dan server. Untuk membuat aplikasi ini dapat diimplementasikan adalah *handphone* yang digunakan terlebih dahulu harus tersambung dengan internet yang gunanya untuk mengambil data yang ada di server, mendapatkan kordinat, dan penggunaan rute yang nantinya dihubungkan ke google maps. Penggunaan json agar request dan respon dari client ke server agar lebih mudah untuk digunakan. Karena, dari json sendiri digunakan untuk melakukan perturakan dan penyimpanan data, json juga sangat ideal dengan Bahasa pemrograman yang digunakan yaitu kotlin.

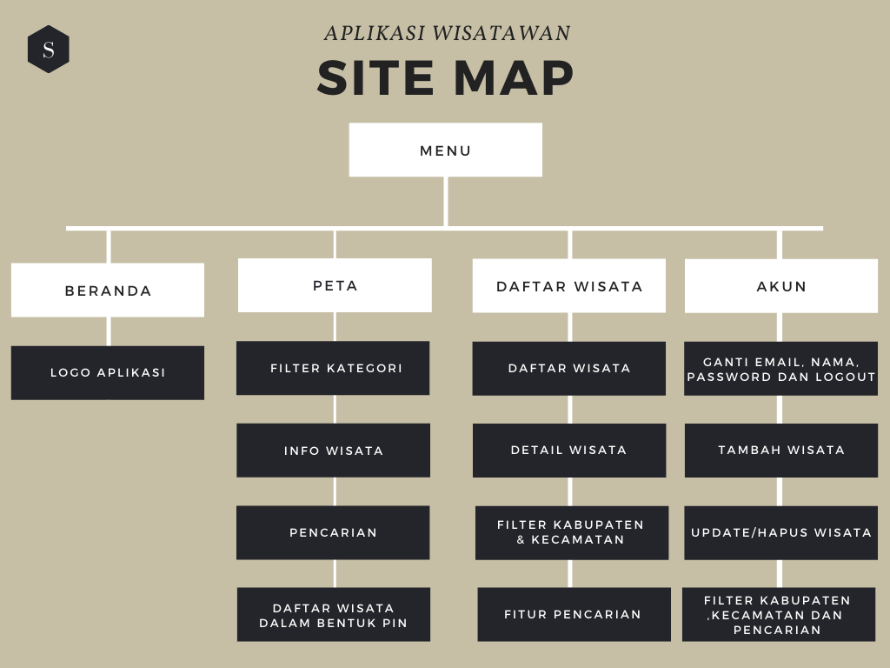
### Pembuatan SIG Pemetaan Lokasi Wisata Alam

Pada Tahap ini merupakan pembuatan kelas SIG Pemetaan Lokasi Wisata Alam pada IDE Android Studio, yang nantinya berfungsi untuk menjalankan aplikasi sebagaimana mestinya. Terdapat 2 aplikasi yaitu Aplikasi untuk Wisatawan dan admin . Penjelasan terkait dengan sitemap dan kelas-kelasnya dijelaskan pada setiap aplikasi :



Gambar 4.16.Sitemap aplikasi untuk admin

Pada Gambar 4.16. merupakan sitemap aplikasi untuk admin, dimana pada halaman beranda terdapat fitur keluar, verifikasi, tambah data, edit dan hapus data. Tentunya untuk dapat mengakses aplikasi ini harus login terlebih dahulu



Gambar 4.17.Sitemap aplikasi untuk wisatawan

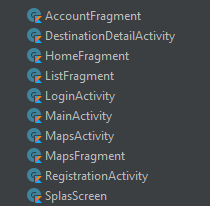
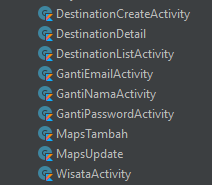
Seperti pada Gambar 4.17 yang merupakan sitemap aplikasi untuk wisatawan, terdapat beberapa fitur utama pada aplikasi ini seperti beranda, peta, daftar wisata dan akun. Untuk mengakses beranda, peta, dan daftar wisata tidak perlu melakukan login kecuali untuk mengakses wisatawan dapat melakukan tambah lokasi diperlukan login.

* + - 1. **Aplikasi Untuk Wisatawan**

Pembuatan kelas pada aplikasi untuk wisatawan berguna untuk membuat aplikasi ini dapat berjalan sebagaimana mestinya dan menerapkan UI/UX dari prototype yang telah dibuat pada Langkah sebelumnya ke aplikasi sesungguhnya.

* + - * 1. **Activity (Kotlin)**

Package Activity pada Gambar 4.18. merupakan sebuah kelas yang nantinya terhubung ke Layout (XML) atau tampilan dari aplikasi. Activity ini nantinya tempat dilakukan koding untuk membuat fungsi-fungsi yang akan ditampilkan pada sebuah halaman.

Gambar 4.18.Kelas Activity

* + - * 1. **Kelas AccountFragment**

Kelas AccountFragment merupakan halaman yang digunakan untuk melakukan login jika wisatawa ingin menambahkan data. Nantinya, Ketika wisatawan sudah login, maka akan menampilkan akun seperti nama, email, password, button wisata dan tombol keluar.

* + - * 1. **DestinationDetailActivity**

Kelas DestinationDetailActivity merupakan sebuah kelas yang akan menampilkan informasi detail terkait dengan lokasi wisata yang terdapat pada list maupun peta. Pada kelas ini juga berguna untuk mengambil informasi yang ada pada database untuk di tampilkan pada halaman Detail dari Wisata yang di klik pada halaman list. Terdapat gambar, nama destinasi, kategori, alamat, deskripsi, kabupaten, kecamatan, jam buka, jam tutup serta tombol bagikan. Kelas DestinationDetailActivity ini terhubung dengan layout xml activity\_destiny\_detail.xml.

* + - * 1. **HomeFragment**

Kelas HomeFragment merupakana kelas yang berguna untuk menampilkan logo dari aplikasi Siwita, kelas ini terhubung dengan layout XML fragment\_home.xml.

* + - * 1. **ListFragment**

Kelas ListFragment merupakan kelas yang berguna untuk menampilkan Daftar destinasi yang sudah di publikasi. Di kelas fragment ini terdapat beberapa fungsi seperti menampilkan menu dropdown (spinner) kabupaten, menampilkan menu dropdown (spinner) kecamatan, fungsi pencarian dan menampilkan daftar destinasi. Kelas ini terhubung dengan layout xml fragment\_list.xml.

* + - * 1. **MainActivity**

Kelas MainActivity merupakan kelas utama pada aplikasi ini, kegunaan dari MainActivity ini unttuk melakukan handle terhadap ListFragment, MapsFragment, BantuanFragment, dan HomeFragment. Pada Kelas ini juga digunakan untuk membuat menu pada bagian bawah aplikasi seperti home, map, list wisata dan bantuan. Dapat dikatakan bahwa kelas MainActivity ini sebagai penghubung antara setiap Fragment yang sudah dibuat. Kelas ini terhubung dengan layout xml activity\_main.xml.

* + - * 1. **MapsFragment**

Kelas MapsFragment merupakan kelas yang berguna untuk menampilkan peta pada aplikasi. Selain peta, pada kelas ini juga memiliki banyak fungsi seperti fungsi menampilkan daftar destinasi dalam bentuk pin, fungsi pencarian yang berguna untuk mencari lokasi wisata yang ada pada aplikasi, fungsi filter yang berguna untuk melakukan filter berdasarkan kategori, menampilkan informasi berdasarkan wisata yang dipilih. Kelas ini juga terhubung pada google maps. Jadi, Ketika pengguna melakukan klik pada salah satu daftar wisata, maka nantinya akan menampilkan tombol untuk melihat rute dan diarahkan langsung ke google maps. Kelas ini terhubung dengan layout XML fragment\_maps.xml.

* + - * 1. **SplashScreen**

Kelas SplashScreen merupakan kelas yang pertama kali ditampilkan kepada user berupa logo dari aplikasi, dimana kelas ini akan tampil selama 5 detik yang kemudian diarahkan ke kelas ListFragment. Kelas ini terhubung ke activity\_splash\_screen.xml

* + - * 1. **RegistrationActivity**

Kelas RegistrationActivity merupakan kelas yang digunakan wisatawan untuk mendaftarkan diri ke aplikasi, terdapat form untuk wisatawan, yang berisikan email, nama dan password. Jika Email belum terdaftar maka berhasil terdaftar, namun jika email sudah terdaftar akan menampilkan pesan bahwa email sudah terdaftar.

* + - * 1. **GantiEmailActivity**

Kelas GantiEmailAcivity digunakan Ketika di halaman akun user melakukan klik pada menu email. Pada halaman ini akan menampilkan sebuah form yang berisikan email lama dan email baru. Jika, email lama yang diinputkan benar maka dapat melakukan perubahan email. Namun, jika email lama salah maka akan menampilkan pesan error email lama salah.

* + - * 1. **GantiNamaActivity**

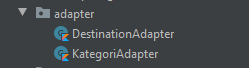
Kelas GantiNamActivity digunakan Ketika di halaman akun user melakukan klik pada menu nama. Pada halaman ini akan menampilkan sebuah form yang berisikan nama. Jika, wisatawan hanya melakukan perubahan pada nama yang ada pada form tersebut. Maka jika sudah di klik tombol update, maka nama akan berubah.

* + - * 1. **GantiPasswordActivity**

Kelas GantiPasswordActivity digunakan Ketika di halaman akun user melakukan klik pada menu password. Pada halaman ini akan menampilkan sebuah form yang berisikan password lama dan password baru. Jika, password lama yang diinputkan benar maka dapat melakukan perubahan password. Namun, jika password lama salah maka akan menampilkan pesan error password lama salah.

* + - 1. **Adapter (Kotlin)**

Berbeda dengan package Activity, package adapter pada Gambar 4.19. ini merupakan kelas yang menghubungkan antara data yang ada pada database ke sebuah listview/recycleview pada aplikasi, data yang biasanya akan ditampilkan ini berupa data dalam bentuk arraylist. Jadi pada setiap adapter akan menerima ArrayList, kemudian ditampilkan pada sebuah listview/recycleview.



Gambar 4.19.Adapter

* + - * 1. **DestinationAdapter**

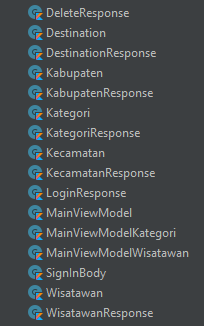
Kelas DestinationAdapter merupakan sebuah kelas ArrayAdapter yang berguna untuk menampilkan data pada tabel destination. Data pada tabel destination tersebut nantinya berupa ArrayList, ArrayList ini didapatkan pada kelas model MainViewModel yang dikirimkan ke DestinationAdapter kemudian ditampilkan pada sebuah layout yang telah dibuat pada layout list\_item.xml. DestinationAdapter ini terhubung dengan kelas ListFragment, dimana pada halaman ListFragment sendiri dibuat sebuah object dari DestinationAdapter untuk memanggil list wisata yang sudah dibuat pada Recycleview untuk ditampilkan pada halaman ListFragment.

* + - * 1. **KategoriAdapter**

Kelas KategoriAdapter merupakan kelas ArrayAdapter yang berguna untuk menampilkan data pada tabel kategori. Data kategori ini berupa Arraylist pada MainViewModelKategori yang akan diletakkan pada sebuah layout yang dibuat pada sebuah xml bernama item\_row\_category.xml. DestinationAdapter ini terhubung dengan kelas ListFragment, dimana pada halaman MapsFragment sendiri dibuat sebuah object dari KategoriAdapter untuk memanggil daftar kategori yang sudah dibuat pada Recycleview untuk ditampilkan Ketika tombol filter kategori di klik dan akan menampilkan nama-nama kategori.

* + - 1. **Model (Kotlin)**

Package model ini merupakan pengimplementasian pada sebuah tabel di database. Dimana dapat dilihat nama setiap kelas yang dibuat merupakan nama yang ada pada tabel di database. Namun, pada package model ini juga nantinya digunakan untuk mengambil data pada database berupa arraylist seperti MainViewModel dan MainViewModelKategori untuk mengambil array pada database seperti pada Gambar 4.20.



Gambar 4.20.Kelas Model

* + - * 1. **Destination**

Kelas Destination merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel destination seperti id\_destination, name\_desination, lat\_destination, lng\_destination, address\_destination, desc\_destination, id\_kategori, img\_destination, id\_kabupaten, id\_kecamatan, jambuka dan jamtutup. Kelas ini nantinya akan banyak digunakan Ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **DestinationResponse**

Kelas DestinationResponse merupakan sebuah data class yang terhubung dengan kelas Destination, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API Destination, jika statusnya true maka akan menampilkan destinasi dalam bentuk array, namun jika false akan menampilkan pesan error.

* + - * 1. **Kabupaten**

Kelas Destination merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel kabupaten seperti id\_kabupaten, dan name\_kabupaten. Kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **KabupatenResponse**

Kelas KabupatenResponse merupakan sebuah data class yang terhubung dengan kelas Kabupaten, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API Kabupaten, jika statusnya true maka akan menampilkan kategori dalam bentuk array, namun jika false akan menampilkan pesan error di Array APInya.

* + - * 1. **Kategori**

Kelas Kategori merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel kategori seperti id\_kategori, dan name\_kategori. Kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **KategoriResponse**

Kelas Kategori Response merupakan sebuah data class yang terhubung dengan kelas Kategori, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API Kategori, jika statusnya true maka akan menampilkan kategori dalam bentuk array, namun jika false akan menampilkan pesan error di Array APInya.

* + - * 1. **Kecamatan**

Kelas Kecamatan merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel kecamtan seperti id\_kecamatan, name\_kecamatan, dan id\_kabupaten. Kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **KecamatanResponse**

Kelas KecamatanResponse merupakan sebuah data class yang terhubung dengan kelas Kecamatan, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API Kecamatan, jika statusnya true maka akan menampilkan kecamatan dalam bentuk array, namun jika false akan menampilkan pesan error di Array APInya.

* + - * 1. **MainViewModel**

Kelas MainViewModel merupakan sebuah kelas yang digunakan untuk mengambil data pada database dalam bentuk ArrayList. Dimana pada kelas ini mengimplementasikan Kelas Model Destination dan DestinationResponse untuk terhubung dengan APInya terkait dengan data dan status dari API Destination. Terdapat fungsi setDestination yang berguna untuk mengambil semua data yang ada pada API. Ada juga fungsi getDestination yang berguna untuk melakukan return dari data destination yang sudah diambil dari ArrayList Destination.

* + - * 1. **MainViewModelWisatawan**

Kelas MainViewModelWisatawan merupakan sebuah kelas yang digunakan untuk mengambil data pada database dalam bentuk ArrayList. Dimana pada kelas ini mengimplementasikan Kelas Model wisatawan dan WisatawanResponse untuk terhubung dengan APInya terkait dengan data dan status dari API wisatawan. Terdapat fungsi setKategori yang berguna untuk mengambil semua data yang ada pada API. Ada juga fungsi getWisatawan yang berguna untuk melakukan return dari data kategori yang sudah diambil dari ArrayList Destination .

* + - * 1. **LoginResponse**

Kelas LoginResponse merupakan sebuah data class yang terhubung dengan kelas SingInBody, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API admin, jika statusnya true maka akan menampilkan admin dalam bentuk array, namun jika false akan menampilkan pesan error di Array APInya.

* + - * 1. **SignInBody**

Kelas SignInBody merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel admin seperti id\_admin, email, dan password. Kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **MainViewModelKategori**

Kelas MainViewModelKategori merupakan sebuah kelas yang digunakan untuk mengambil data pada database dalam bentuk ArrayList. Dimana pada kelas ini mengimplementasikan Kelas Model kategori dan KategoriResponse untuk terhubung dengan APInya terkait dengan data dan status dari API Kategori. Terdapat fungsi setKategori yang berguna untuk mengambil semua data yang ada pada API. Ada juga fungsi getDestination yang berguna untuk melakukan return dari data kategori yang sudah diambil dari ArrayList Destination .

* + - * 1. **Wisatawan**

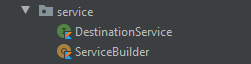
Kelas wisatawan merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel kecamtan seperti id\_wisatawan, email, nama, password. Kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **WisatawanResponse**

Kelas WisatawanResponse merupakan sebuah data class yang terhubung dengan kelas wisatawan, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API wisatawan, jika statusnya true maka akan menampilkan kecamatan dalam bentuk array, namun jika false akan menampilkan pesan error di Array APInya.

* + - 1. **Service (Kotlin)**

Gambar 4.21. Package Service merupakan cara yang digunakan untuk terhubung dengan API yang sudah dibuat sebelumnya. Package ini nantinya terbagi menjadi 2 yaitu SericeBuilder dan DestinationService.



Gambar 4.21.Package Service

* + - * 1. **ServiceBuilder**

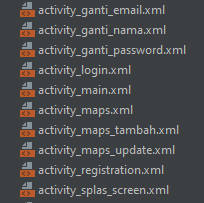
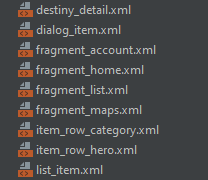
Kelas Object ServiceBuilder merupakan kelas utama yang nantinya menghubungkan aplikasi dengan API. Dimana pada kelas ini juga mengimplementasikan Gson untuk menjadi data yang semulanya Array menjadi Gson. Kemudian pada kelas ini juga menghubungkan aplikasi dengan Base url dari APInya. Terakhir, kelas ini juga terhubung dengan kelas interface DestinationService.

* + - * 1. **DestinationService**

Berbeda dengan SericeBuilder yang mengambil base url dari API, kelas interface DestinationService ini merupakan kelas yang mengimplementasi endpoint dari base url yang telah dibuat pada API. Nantinya pada kelas ini terdapat fungsi GET, POST, dan DELETE. Dimana disetiap fungsi yang dibuat mengimplementasikan kelas Response pada package Model.

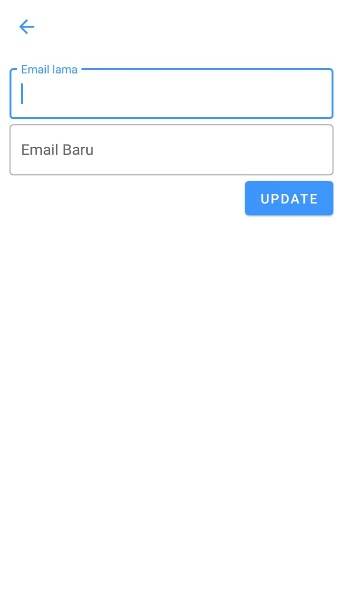
* + - 1. **Layout (XML)**

Gambar 4.22. Layout Wisatawan merupakan sebuah package yang berguna untuk tempat membuat sebuah interface yang nantinya akan dilihat oleh pengguna.

Gambar 4.22.Layout Wisatawan

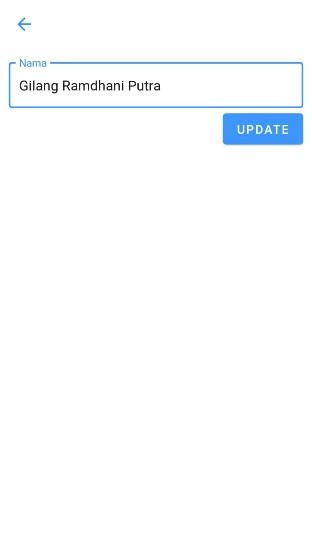
* + - * 1. **Activity\_ganti\_email**



Gambar 4.23.Layout Ganti Email

Gambar 4.23. Layout ganti email ini adalah merupakan tempat untuk membuat tampilan untuk kelas GantiEmailActivity, dimana terdapat 2 edittext menerima inputan berupa email lama dan email baru, kemudian juga terdapat button update untuk melakukan update password.

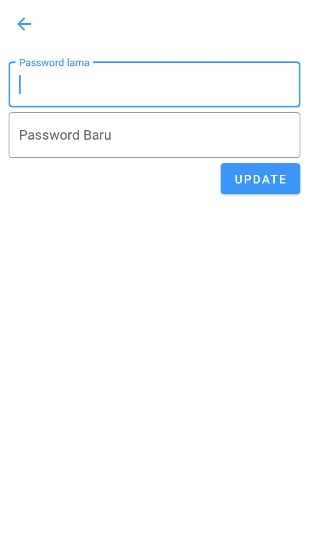
* + - * 1. **Actvitiy\_ganti\_nama**



Gambar 4.24.Layout Ganti Nama

Gambar 4.24. Layout ganti nama ini adalah merupakan tempat untuk membuat tampilan untuk kelas GantiNamaActivity, dimana terdapat 1 inputan berupa nama dan juga terdapat button update untuk melakukan update nama.

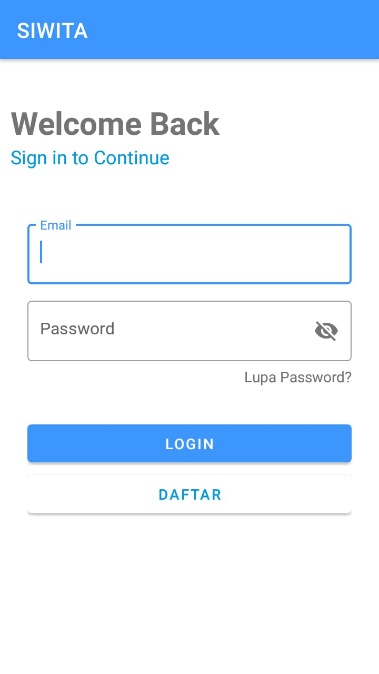
* + - * 1. **Activity\_ganti\_password**



Gambar 4.25.Layout Ganti Password

Gambar 4.25 Layout ganti password ini adalah merupakan tempat untuk membuat tampilan untuk kelas GantiPasswordActivity, dimana terdapat 2 edittext untuk menerima input-an berupa password lama dan password baru dan juga terdapat button update untuk melakukan update password.

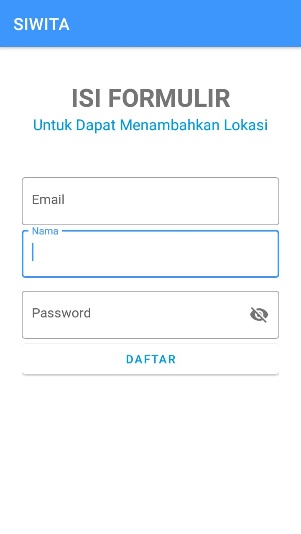
* + - * 1. **Activity\_login**



Gambar 4.26.Layout login

Layout login merupakan tempat untuk membuat tampilan login bagi kelas LoginActivity, dimana terdapat 2 edittext untuk menampung email dan menampung password. Kemudian juga terdapat button Login dan Button Daftar.

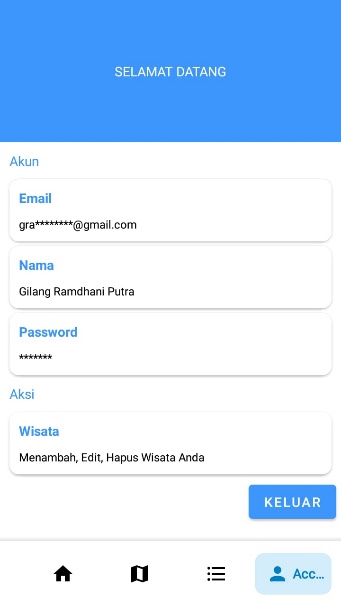
* + - * 1. **Activity\_registration**



Gambar 4.27.Layout daftar

Layout activity registration merupakan tampilan untuk kelas RegistrationActivity, dimana terdapat 3 edittext untuk menampung email, nama dan password. Kemudian terdapat juga button daftar yang berguna untuk mengirim semua inputan ke database.

* + - * 1. **Fragment\_account**

Gambar 4.28.Sebelum login Gambar 4.29.Setelah login

Layout fragment\_account merupakan layout untuk menampilkan tampilan pada kelas AccountFragment, dimana pada kelas ini di atur Ketika wisatawan belum login maka akan menampilkan button login seperti Gambar 4.28. Namun, jika sudah login akan menampilkan tampilan berupa menu akun dan menu wisata serta button keluar seperti pada Gambar 4.29..

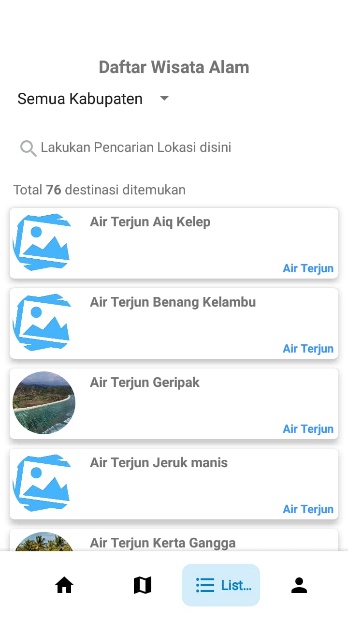
* + - * 1. **Fragment\_home**



Gambar 4.30.Layout home

Gambar 4.30. fragment\_home merupakan tampilan yang ditampilkan setelah dari kelas SplasScreen, pada kelas ini akan menampilkan logo dari aplikasi.

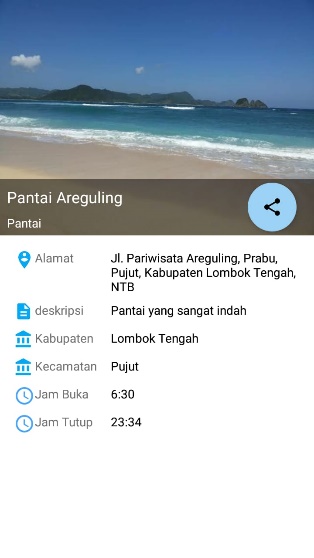
* + - * 1. **Fragment\_list**



Gambar 4.31.Layout list

Gambar 4.31. Layout fragment\_list merupakan tampilan untuk menampilkan list wisata dari database. Terdapat juga menu dropdown untuk melakukan filter berdasarkan kategori dan kecamatan, begitu juga terdapat menu pencarian pada halaman ini.

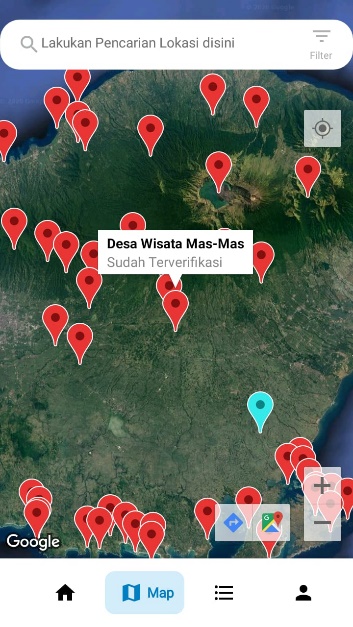
* + - * 1. **Activity\_detail**



Gambar 4.32.layout\_detail

Layout ini merupakan tampilan untuk menampilkan detail dari lokasi wisata Ketika salah satu list wisata pada fragment\_list di klik. Maka nantinya akan menampilkan gambar seperti gambar 4.32 yaitu terdapat informasi berupa nama, kategori, alamat, deskripsi, kabupaten, kecamatan, jam buka dan jam tutup.

* + - * 1. **Fragment\_maps**



Gambar 4.33.layout maps

Gambar 4.33 Layout fragment\_maps merupakan tampilan untuk menampilkan peta wisata dengan data berupa pin atau marker. Terdapat beberapa menu pada halaman ini seperti lokasi saat ini, filter kategori, pencarian, dan rute.

* + - * 1. **Item\_row\_category**

Layout item\_row\_category merupakan tampilan untuk menampilkan daftar kategori Ketika filter kategori pada halaman fragment\_maps di klik.

* + - * 1. **List\_item**

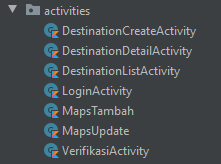
Layout list\_item merupakan tampilan untuk menampilkan list item yang nantinya digunakan untuk menampilkan daftar wisata pada halaman fragment\_list.

* + - 1. **Aplikasi Untuk Admin**

Pembuatan kelas pada aplikasi untuk admin berguna untuk membuat aplikasi ini dapat berjalan sebagaimana mestinya dan menerapkan UI/UX dari prototype yang telah dibuat pada Langkah sebelumnya ke aplikasi sesungguhnya.

* + - * 1. **Activity ( Kotlin )**

Package Activity seperti Gambar 4.34 merupakan sebuah kelas yang nantinya terhubung ke Layout (XML) atau tampilan dari aplikasi. Activity ini nantinya tempat dilakukan koding untuk membuat fungsi-fungsi yang akan ditampilkan pada sebuah halaman. Pada package activities aplikasi admin terdapat beberapa kelas Activity seperti berikut :



Gambar 4.34.Package Activity

* + - * 1. **DestinationCreateActivity**

Kelas Activity Destination Create ini merupakan sebuah kelas yang berguna untuk melakukan penambahan lokasi wisata. Pada kelas ini terdapat fungsi tambah yang dapat digunakan untuk mengirimkan data yang sudah di *input* admin ke dalam server. Selain fungsi tambah data, terdapat juga fungsi untuk mengambil gambar pada galeri yang nantinya digunakan untuk mengambil gambar dan dikirimkan ke server. Kelas ini terhubung dengan layout activity\_create\_destination.xml.

* + - * 1. **DestinationDetailActivity**

Kelas Activity Destination Detail merupakan sebuah kelas yang digunakan untuk menampilkan halaman detail dari list destinasi yang ditampilkan pada kelas Destination List. Dimana pada halaman detail destination ini juga merupakan kelas untuk melakukan hapus atau update informasi lokasi. Fungsi yang ada pada kelas ini seperti fungsi update, delete, dan ambil gambar di galeri. Kelas ini terhubung dengan layout activity\_detail\_destination.xml.

* + - * 1. **DestinationListActivity**

Kelas Activity Destination List adalah kelas yang pertama kali ditampilkan Ketika admin berhasil login ke dalam aplikasi. Dimana pada kelas ini nantinya akan menampilkan daftar wisata yang telah di tambahkan pada kelas DestinationCreateActivity. Pada kelas ini terdapat beberapa fungsi seperti fungsi load destination untuk menampilkan daftar wisata, load kabupaten untuk menampilkan daftar kabupaten, load kecamatan untuk menampilkan daftar kecamatan dan fungsi pencarian untuk melakukan pencarian berdasrakan nama lokasi. Selain itu, pada kelas ini juga terdapat tombol tambah yang nantinya Ketika di klik akan diarahkan ke hlamana MapsTambah untuk melakukan tambah lokasi berdasrkan lokasi yang diklik pada peta terlebih dahulu.

* + - * 1. **LoginActivity**

Kelas LoginActivity merupakan kelas yang digunakan untuk menampilkan halaman login untuk admin sebelum masuk ke dalam aplikasi. Dimana pada halaman login ini admin harus terlebih dahulu melakukan input email dan password untuk dapat login. Fungsi pada kelas ini adalah fungsi login. Pada LoginActivity ini nantinya juga akan menyimpan informasi terkait dengan data dari admin dan akan disimpan pada session. Kelas ini terhubung dengan layout activity\_login.xml.

* + - * 1. **MapsTambah**

Kelas MapsTambah merupakan kelas yang dijalankan Ketika admin melakukan klik pada tombol tambah di halaman Destination List. Kelas ini bertujuan untuk memilih lokasi berdasarkan lokasi yang di klik pada peta. Nantinya Ketika peta di klik, maka akan menampilkan marker yang sebagai penanda bahwa lokasi sudah dipilih. Jika sudah memilih lokasi, maka akan diarahkan ke halaman DestinationCreate untuk dilakukan penambahan informasi lainnya.

* + - * 1. **MapsUpdate**

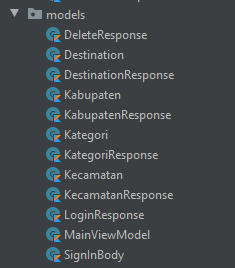
Kelas MapsTambah merupakan kelas yang dijalankan Ketika admin melakukan klik pada salah tombol pada di halaman Destination Detail. Kelas ini bertujuan untuk memilih lokasi berdasarkan lokasi yang di klik pada peta. Nantinya Ketika peta di klik, maka akan menampilkan marker yang sebagai penanda bahwa lokasi sudah dipilih. Jika sudah memilih lokasi, maka akan diarahkan ke halaman Destination Detail untuk dilakukan perubahan lokasi peta dan informasi lainnya.

* + - * 1. **VerifikasiActivity**

Kelas VerifikasiActivity merupakan kelas yang berisikan daftar wisata yang telah di ditambahkan oleh wisatawan untuk dilakukan validasi. Admin dapat melakukan pengecekkan informasi wisata kemudian meng-klik tombol update untuk membuat lokasi tersebut tervalidasi.

* + - 1. **Model (Kotlin)**

Package model Gambar 4.35. merupakan pengimplementasian pada sebuah tabel di database. Dimana dapat dilihat nama setiap kelas yang dibuat merupakan nama yang ada pada tabel di database. Namun, pada package model ini juga nantinya digunakan untuk mengambil data pada database berupa arraylist seperti MainViewModel untuk mengambil array pada database.



Gambar 4.35.Package Model

* + - * 1. **Destination**

Kelas Destination merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel destination seperti id\_destination, name\_desination, lat\_destination, lng\_destination, address\_destination, desc\_destination, id\_kategori, img\_destination, id\_kabupaten, id\_kecamatan, jambuka dan jamtutup. Kelas ini nantinya akan banyak digunakan Ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **DestinationResponse**

Kelas DestinationResponse merupakan sebuah data class yang terhubung dengan kelas Destination, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API Destination, jika statusnya true maka akan menampilkan destinasi dalam bentuk array, namun jika false akan menampilkan pesan error.

* + - * 1. **Kabupaten**

Kelas Destination merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel kabupaten seperti id\_kabupaten, dan name\_kabupaten. Kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **KabupatenResponse**

Kelas KabupatenResponse merupakan sebuah data class yang terhubung dengan kelas Kabupaten, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API Kabupaten, jika statusnya true maka akan menampilkan kategori dalam bentuk array, namun jika false akan menampilkan pesan error di Array APInya.

* + - * 1. **Kategori**

Kelas Kategori merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel kategori seperti id\_kategori, dan name\_kategori. Kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **KategoriResponse**

Kelas Kategori Response merupakan sebuah data class yang terhubung dengan kelas Kategori, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API Kategori, jika statusnya true maka akan menampilkan kategori dalam bentuk array, namun jika false akan menampilkan pesan error di Array APInya.

* + - * 1. **Kecamatan**

Kelas Kecamatan merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel kecamtan seperti id\_kecamatan, name\_kecamatan, dan id\_kabupaten. Kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

* + - * 1. **KecamatanResponse**

Kelas KecamatanResponse merupakan sebuah data class yang terhubung dengan kelas Kecamatan, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API Kecamatan, jika statusnya true maka akan menampilkan kecamatan dalam bentuk array, namun jika false akan menampilkan pesan error di Array APInya.

* + - * 1. **MainViewModel**

Kelas MainViewModel merupakan sebuah kelas yang digunakan untuk mengambil data pada database dalam bentuk ArrayList. Dimana pada kelas ini mengimplementasikan Kelas Model Destination dan DestinationResponse untuk terhubung dengan APInya terkait dengan data dan status dari API Destination. Terdapat fungsi setDestination yang berguna untuk mengambil semua data yang ada pada API. Ada juga fungsi getDestination yang berguna untuk melakukan return dari data destination yang sudah diambil dari ArrayList Destination.

* + - * 1. **LoginResponse**

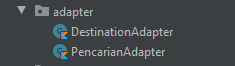
Kelas LoginResponse merupakan sebuah data class yang terhubung dengan kelas SingInBody, dimana kelas ini berguna untuk memberikan informasi terkait dengan status dan data yang digunakan pada API admin, jika statusnya true maka akan menampilkan admin dalam bentuk array, namun jika false akan menampilkan pesan error di Array APInya.

* + - * 1. **SignInBody**

Kelas SignInBody merupakan sebuah data class yang berguna untuk mengimplementasikan entitas apa saja yang ada pada tabel admin seperti id\_admin, email, dan password. Kelas ini nantinya akan banyak digunakan ketika ingin melakukan interaksi dengan data pada database.

* + - 1. **Adapter(Kotlin)**

Berbeda dengan package Activity, package adapter pada gambar 4.36. ini merupakan kelas yang menghubungkan antara data yang ada pada database ke sebuah listview/recycleview pada aplikasi, data yang biasanya akan ditampilkan ini berupa data dalam bentuk arraylist. Jadi pada setiap adapter akan menerima ArrayList, kemudian ditampilkan pada sebuah listview/recycleview.



Gambar 4.36.Package Adapter

* + - * 1. **DestinationAdapter**

Kelas DestinationAdapter merupakan sebuah kelas ArrayAdapter yang berguna untuk menampilkan data pada tabel destination. Data pada tabel destination tersebut nantinya berupa ArrayList, ArrayList ini didapatkan pada kelas model MainViewModel yang dikirimkan ke DestinationAdapter kemudian ditampilkan pada sebuah layout yang telah dibuat pada layout list\_item.xml. DestinationAdapter ini terhubung dengan kelas DestinationListActivity, dimana pada halaman DestinationListActivity sendiri dibuat sebuah object dari DestinationAdapter untuk memanggil list wisata yang sudah dibuat pada Recycleview untuk ditampilkan pada halaman DestinationListActivity.

* + - * 1. **PencarianAdapter**

Kelas PencairanAdapter merupakan kelas ArrayAdapter yang berguna untuk menampilkan data pada tabel destination. Data destination ini berupa Arraylist pada MainViewModel yang akan diletakkan pada sebuah layout yang dibuat pada sebuah xml bernama item\_list.xml. DestinationAdapter ini terhubung dengan kelas DestinationListActivity, dimana pada halaman DestinationListActivity sendiri dibuat sebuah object dari KategoriAdapter untuk memanggil daftar wisata yang sudah dibuat pada Recycleview untuk ditampilkan Ketika tombol pencarian di klik dan dituliskan nama destinasi yang diinginkan, maka nantinya akan diberikan rekomendasi nama wisata.

* + - 1. **SessionManager (Kotlin)**

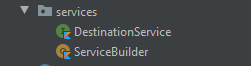
Gambar 4.37. SessionManager merupakan sebuah kelas yang digunakan untuk menyimpan data dari admin Ketika berhasil melakukan login seperti id, email, password, dan status dari admin apakah sedang login atau tidak. Jika admin sudah login dan keluar dari aplikasi, maka aplikasi nantinya akan menyimpan data dari admin, sehingga admin tidak perlu melakukan login lagi. Pada kelas ini juga terdapat fungsi clearsession yang berguna Ketika admin melakukan logout, maka semua data admin Ketika login akan di hapus sehingga tidak dapat masuk ke dalam halaman beranda admin jika belum login.



Gambar 4.37.Kelas SessionManager

* + - 1. **Services (Kotlin)**

Package Service merupakan cara yang digunakan untuk terhubung dengan API yang sudah dibuat sebelumnya. Package ini nantinya terbagi menjadi 2 yaitu SericeBuilder dan DestinationService seperti yang dapat dilihat pada Gambar 4.38.



Gambar 4.38.Package Services

* + - * 1. **ServiceBuilder**

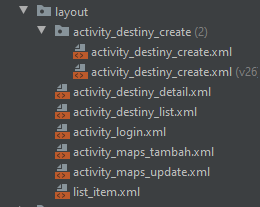
Kelas Object ServiceBuilder merupakan kelas utama yang nantinya menghubungkan aplikasi dengan API. Dimana pada kelas ini juga mengimplementasikan Gson untuk menjadi data yang semulanya Array menjadi Gson. Kemudian pada kelas ini juga menghubungkan aplikasi dengan Base url dari APInya. Terakhir, kelas ini juga terhubung dengan kelas interface DestinationService.

* + - * 1. **DestinationService**

Berbeda dengan SericeBuilder yang mengambil base url dari API, kelas interface DestinationService ini merupakan kelas yang mengimplementasi endpoint dari base url yang telah dibuat pada API. Nantinya pada kelas ini terdapat fungsi GET, POST, dan DELETE. Dimana disetiap fungsi yang dibuat mengimplementasikan kelas Response pada package Model.

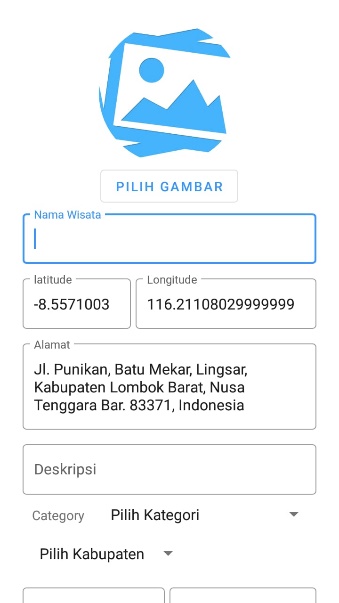
* + - 1. **Layout ( XML )**

Gambar 4.39 merupakan sebuah package yang berguna untuk tempat membuat sebuah interface yang nantinya akan dilihat oleh pengguna. Seperti yang dapat dilihat terdapat beberapa xml yang nantinya akan digunakan Ketika kelas Activity dijalankan.



Gambar 4.39.Package layout

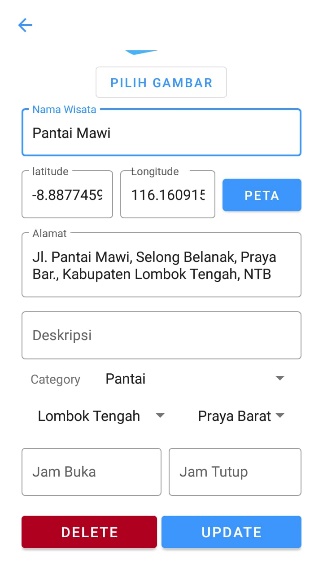
* + - * 1. **Activity\_destiny\_create**



Gambar 4.40.Layout tambah

Gambar 4.40. merupakan sebuah dokumen XML yang berguna untuk membuat tampilan dari halaman Destinasi Create atau halaman tambah dari halaman admin. Pada halaman ini digunakan beberapa view seperti edittext, button, imageview. Sedangkan untuk ViewGroupnya digunakan ScrollView, ConstraintLayout dan Linearlayout. Layout ini terhubung dengan kelas DestinationCreateActivity.

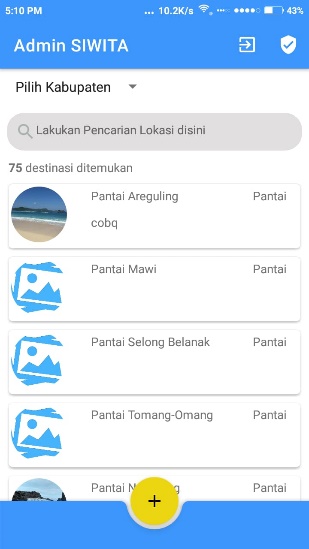
* + - * 1. **Activity\_destiny\_detail**



Gambar 4.41.Layout detail

Gambar 4.41. merupakan sebuah dokumen XML yang berguna untuk membuat tampilan dari halaman Destinasi Detail atau halaman update dari halaman admin. Pada halaman ini digunakan beberapa view seperti edittext, button, imageview. Sedangkan untuk ViewGroupnya digunakan ScrollView, ConstraintLayout dan Linearlayout. Layout ini terhubung dengan kelas DestinationDetailActivity.

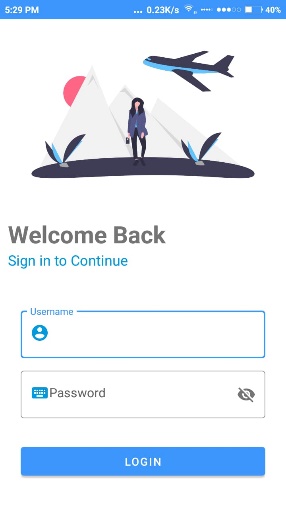
* + - * 1. **Activity\_destiny\_list**



Gambar 4.42.Layout list

Gambar 4.42. merupakan sebuah dokumen XML yang berguna untuk membuat tampilan dari halaman Destinasi List atau halaman list dari halaman admin. Pada halaman ini digunakan beberapa view seperti spinner(dropdown), autocomplete(pencarian), Textview. Sedangkan untuk ViewGroupnya digunakan ScrollView, dan RecyleView. Layout ini terhubung dengan kelas DestinationListActivity.

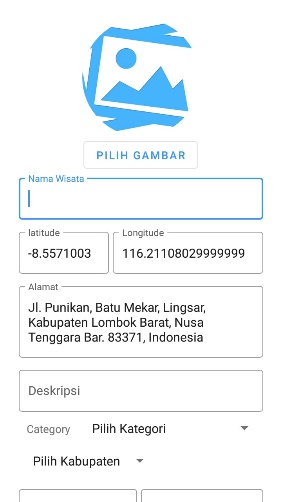
* + - * 1. **Activity\_login**



Gambar 4.43.Layout login

Gambar 4.43. merupakan sebuah dokumen XML yang berguna untuk membuat tampilan dari halaman Login Activity atau login dari aplikasi admin. Pada halaman ini digunakan beberapa view seperti edittext, button, textview. Sedangkan untuk ViewGroupnya digunakan ScrollView, dan Linearlayout. Layout ini terhubung dengan kelas LoginActivity.

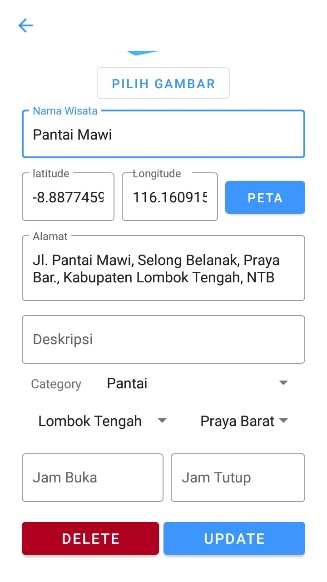
* + - * 1. **Activity\_maps\_tambah**



Gambar 4.44.Layout tambah

Gambar 4.44. merupakan sebuah dokumen XML yang berguna untuk membuat tampilan dari halaman Maps Tambah atau halaman Ketika admin klik tombol tambah di halaman destination list activity. Pada halaman ini digunakan fragment yang melakukan implement maps. Sedangkan untuk ViewGroupnya Linearlayout. Layout ini terhubung dengan kelas MapsTambah.

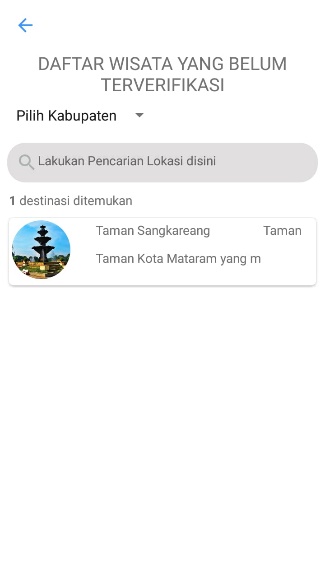
* + - * 1. **Activity\_maps\_update**



Gambar 4.45.Layout update

Gambar 4.45. merupakan sebuah dokumen XML yang berguna untuk membuat tampilan dari halaman Maps Update atau halaman Ketika admin klik tombol peta di halaman Destination Detail Activity. Pada halaman ini digunakan fragment yang melakukan implement maps. Sedangkan untuk ViewGroupnya Linearlayout. Layout ini terhubung dengan kelas MapsUpdate.

* + - * 1. **Activity\_Verifikasi**

****

Gambar 4.46.Halaman Verifikasi

Pada Gambar 4.46. merupakan halaman verifikasi pada aplikasi admin. Halaman ini berguna untuk melakukan verifikasi terhadap lokasi wisata yang telah ditambahkan wisatawan. Pada halaman ini digunakan beberapa view seperti spinner(dropdown), autocomplete(pencarian), Textview. Sedangkan untuk ViewGroupnya digunakan ScrollView, dan RecyleView. Layout ini terhubung dengan kelas VerifikasiActivity.

* + - * 1. **List\_item**

Layout ini merupakan sebuah dokumen XML yang berguna untuk membuat tampilan dari item yang akan ditampilkan pada halaman List Wisata. Pada halaman ini digunakan view yaitu textview dan imageview. Sedangkan untuk ViewGroupnya Linearlayout. Layout ini terhubung dengan kelas DestinationAdapter.

## Pengujian Aplikasi

Pengujian pada aplikasi ini dilakukan dengan menggunakan metode *blackbox testing* dan MOS. Digunakannya metode *blackbox testing* yaitu dengan menguji masing-masing fungsi yang sudah dibuat dalam aplikasi ini. Hal tersebut bermaksud untuk menemukan kesalahan atau bug pada sistem yang sudah dibangun sehingga pada saat aplikasi ini digunakan nantinya dapat berjalan sesuai dengan apa yang diinginkan dan diharapkan oleh pengguna. Pengujian *blackbox testing* sendiri adalah pengujian langsung pada fungsi-fungsi aplikasi tanpa memperhatikan alur eksekusi program. Pengujian ini dilakukan dengan memperhatikan apakah fungsi yang dibuat sudah sesuai atau tidak dengan yang diharapkan.

Selanjutnya, pada pengujuan metode MOS akan dilakukan penyebaran kuesioner kepada 38 responden yang terdiri dari seorang karyawan Dinas Pariwisata Provinsi Nusa Tenggara Barat, dan 37 orang wisatawan untuk melakukan pengujian terhadap sistem yang telah dibuat. Pengujian dilakukan dengan menunjukkan sistem kepada responden untuk dijalankan. Setelah responden menjalankan sistem, diberikan kuesioner yang berisi pertanyaan dan jawaban. Tujuan dari pengujian ini untuk mengetahui kualitas sistem yang dilihat dari sisi pengguna. Metode yang digunakan dalam pengujian kuesioner adalah metode kuantitatif, yaitu hasil pengujian ditentukan dalam suatu nilai angka.

### 4.3.1. Hasil pengujian *blackbox testing*

Metode pengujian *black box* merupakan metode pengujian yang menguji fungsi-fungsi di dalam sistem untuk menentukan apakah fungsi-fungsi tersebut sudah berjalan sesuai harapan atau tidak. Berikut merupakan jabaran fungsi-fungsi di dalam Sistem Informasi Pemetaan Lokasi Wisata Alam yang akan di uji menggunakan metode *black box*:

* + - 1. **Fungsi Pada Aplikasi Admin**

**Fungsi Login**

Fungsi *login* admin dilakukan untuk menguji apakah fungsi yang berjalan pada halaman *login* admin sudah berjalan dengan baik dan benar. Pengujian fungsi *log* *in* admin akan dijelaskan pada Tabel 4.1.

Tabel 4.1.Pengujian Black Box pada Fungsi Login

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Form username dan password disin dengan data yang tidak benar | Login gagal dan akan dirujuk Kembali ke halaman login | Sesuai | Valid |
| Form Username dan password di isi dengan data yang benar | Proses login berhasil kemudian admin diarahkan ke halaman dashboard | Sesuai | Valid |

Pada pengujian pertama terdapat beberapa variasi dalam melakukan login. Pertama apabila form username diisi dan form password kosong, kedua apabila form password diisi dan form username di kosongkan dan yang ketiga apabila form username dan password dikosongkan. Hasil dari kondisi tersebut login tidak berhasil. Jika yang diisi sesuai dengan data yang ada di database maka proses login berhasil.

**Fungsi Tambah**

Pengujian fungsi pencarian rute dilakukan untuk menguji apakah fungsi yang berjalan pada fitur tersebut sudah berjalan dengan baik dan benar. Pengujian fungsi menampilkan data transaksi akan dijelaskan pada Tabel 4.2.

Tabel 4.2.Pengujian Black Box pada Fungsi Tambah

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Memasukkan data dan menekan tombol tambah | Data yang di inputkan berhasil dikirim ke database | Sesuai | Valid |

Pada pengujian fungsi untuk menambahkan data wisata, dilakukan dengan cara menekan tombol tambah pada halaman tambah. Hasil dari pengujian tersebut adalah data yang diinputkan pada halaman tambah berhasil ditambahkan ke database dan ditampilkan di dashboard admin.

**Fungsi Update**

Pengujian fungsi update data wisata dilakukan untuk menguji apakah fungsi yang terdapat pada proses tersebut sudah berjalan dengan baik dan benar. Pengujian fungsi update data wisata dijelaskan pada Tabel 4.3.

Tabel 4.3.Pengujian Black Box pada Fungsi Update

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Melakukan update pada data dan menekan tombol update | Data yang di inputkan berhasil diupdate dan dikirim ke database | Sesuai | Valid |

Pada pengujian fungsi untuk merubah data wisata, dilakukan dengan cara menekan tombol ubah pada halaman detail. Hasil dari pengujian tersebut adalah data yang diinputkan pada halaman detail berhasil diubah lalu dikirim ke database dan ditampilkan di dashboard admin.

* + - 1. **Fungsi Delete**

Pengujian fungsi hapus data wisata dilakukan untuk menguji apakah fungsi yang terdapat pada proses tersebut sudah berjalan dengan baik dan benar. Pengujian fungsi hapus data wisata dijelaskan pada Tabel 4.4.

Tabel 4.4.Pengujian Black Box pada Fungsi Delete

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan tombol hapus pada halaman detail | Data yang dipilih akan terhapus | Sesuai | Valid |

Pada pengujian fungsi untuk hapus data wisata, dilakukan dengan cara menekan tombol hapus pada halaman detail. Hasil dari pengujian tersebut adalah data yang dipilih berhasil dihapus.

* + - 1. **Fungsi Pada Aplikasi Wisatawan**

**Fungsi Menampilkan Lokasi Wisata**

Pengujian fungsi menampilkan lokasi wisata dilakukan untuk menguji apakah fungsi yang terdapat pada proses tersebut sudah berjalan dengan baik dan benar. Pengujian fungsi menampilkan lokasi wisata dijelaskan pada Tabel 4.5.

Tabel 4.5.Pengujian Black Box pada Menampilkan wisata

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Menekan Halaman Peta | Data wisata berhasil ditampilkan | Sesuai | Valid |

Pada pengujian fungsi untuk menampilkan data wisata, dilakukan dengan cara menekan tombol halaman peta pada bagian menu. Hasil dari pengujian tersebut adalah data wisata alam berhasil ditampilkan pada halaman peta.

**Fungsi Pencarian**

Pengujian fungsi pencarian wisata berdasarkan apa yang ditulis oleh wisatawan dilakukan untuk menguji apakah fungsi yang terdapat pada proses tersebut sudah berjalan dengan baik dan benar. Pengujian fungsi pencarian wisata tersebut dijelaskan pada Tabel 4.6.

Tabel 4.6.Pengujian Black Box pada Menampilkan wisata

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Wisatawan menuliskan wisata yang diinginkan pada kolom pencarian | Data wisata berhasil ditampilkan berdasarkan apa yang ditulis oleh wisatawan | Sesuai | Valid |

Pada pengujian fungsi untuk melakukan pencarian wisata wisata, dilakukan dengan cara menuliskan nama wisata yang ingin dicari pada kolom pencarian. Hasil dari pengujian tersebut adalah data wisata alam berhasil ditampilkan berdasarkan apa yang dituliskan oleh wisatawan.

**Fungsi Filter Kategori Wisata**

Pengujian fungsi filter kategori wisata berdasarkan apa yang dipilih oleh wisatawan dilakukan untuk menguji apakah fungsi yang terdapat pada proses tersebut sudah berjalan dengan baik dan benar. Pengujian fungsi filter kategori wisata tersebut dijelaskan pada Tabel 4.7.

Tabel 4.7.Pengujian Black Box pada filter kategori wisata

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Wisatawan menekan tombol filter kategori, kemudian memilih kategori yang diinginkan | Data wisata berhasil ditampilkan berdasarkan kategori yang dipilih | Sesuai | Valid |

Pada pengujian fungsi untuk melakukan filter kategori wisata, dilakukan dengan cara menekan tombol filter kategori pada halaman peta, kemudian memilih kategori yang diinginkan. Hasil dari pengujian tersebut adalah data wisata alam berhasil ditampilkan berdasarkan apa yang dipilih oleh wisatawan.

**Fungsi Filter Kabupaten**

Pengujian fungsi filter kabupaten berdasarkan apa yang ditulis oleh wisatawan dilakukan untuk menguji apakah fungsi yang terdapat pada proses tersebut sudah berjalan dengan baik dan benar. Pengujian fungsi filter kabupaten tersebut dijelaskan pada Tabel 4.8.

Tabel 4.8.Pengujian Black Box pada filter kabupaten

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Wisatawan menekan menu dropdown kabupaten, kemudian memilih kabupaten yang diinginkan | Data wisata berhasil ditampilkan berdasarkan kabupaten yang dipilih | Sesuai | Valid |

Pada pengujian fungsi untuk melakukan filter kabupaten, dilakukan dengan cara menekan tombol filter kabupaten pada halaman daftar wisata, kemudian memilih kabupaten yang diinginkan. Hasil dari pengujian tersebut adalah data wisata alam berhasil ditampilkan berdasarkan kabupaten yang dipilih oleh wisatawan.

* + - 1. **Fungsi Filter Kecamatan**

Pengujian fungsi filter kecamatan berdasarkan apa yang dipilih oleh wisatawan dilakukan untuk menguji apakah fungsi yang terdapat pada proses tersebut sudah berjalan dengan baik dan benar. Pengujian fungsi filter kecamatan tersebut dijelaskan pada Tabel 4.9.

Tabel 4.9.Pengujian Black Box pada filter kecamatan

|  |  |  |  |
| --- | --- | --- | --- |
| Skenario Pengujian | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
| Wisatawan menekan menu dropdown kabupaten, kemudian memilih kabupaten yang diinginkan, lalu memilih kecamatan | Data wisata berhasil ditampilkan berdasarkan kabupaten dan kecamatan yang dipilih | Sesuai | Valid |

Pada pengujian fungsi untuk melakukan filter kecamatan, dilakukan dengan cara menekan tombol filter kabupaten pada halaman daftar wisata dan memilih kabupaten yang diinginkan, kemudian memilih kecamatan. Hasil dari pengujian tersebut adalah data wisata alam berhasil ditampilkan berdasarkan kecamatan yang dipilih oleh wisatawan.

1. **Hasil pengujian MOS**

Setelah dilakukan pengujian MOS berbentuk kuisioner, hasil jawaban yang didapatkan dihitung untuk mendapatkan kesimpulan seberapa layak sistem dapat digunakan dan dipasarkan. Terdapat dua jenis kuisioner yang akan dibagikan, yaitu kuisioner untuk admin (pihak Dinas Pariwisata Provinsi Nusa Tenggara Barat bidang Destinasi) dan calon pengguna aplikasi wisatawan. Kuisioner yang diberikan masing-masing terdiri dari enam pertanyaan. Berikut merupakan isi pertanyaan kedua jenis kuisioner :

1. Admin
2. Apakah tampilan aplikasi mudah dipahami?
3. Apakah aplikasi mudah digunakan?
4. Apakah aplikasi ini dapat membantu pihak dalam mengelola wisata di aplikasi?
5. Apakah aplikasi sudah sesuai dengan kebutuhan pengguna?
6. Apakah aplikasi berjalan dengan baik?
7. Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?
8. Wisatawan
9. Apakah tampilan aplikasi mudah dipahami?
10. Apakah aplikasi mudah digunakan?
11. Apakah aplikasi ini dapat membantu wisatawan dalam mencari lokasi wisata alam yang diinginkan?
12. Apakah aplikasi sudah sesuai dengan kebutuhan pengguna?
13. Apakah aplikasi berjalan dengan baik?
14. Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?

Responden diminta untuk menjawab kuesioner tersebut dengan pilihan jawaban sebagai berikut:

1 = Tidak setuju.

2 = Kurang setuju.

3 = Cukup.

4 = Setuju.

5 = Sangat setuju

Berdasarkan jawaban tersebut, dilakukan perhitungan persentase dari setiap jawaban yang diberikan oleh pengguna di setiap pertanyaannya. Rumus untuk menghitung persentase jawaban adalah sebagai berikut:

Persentase = J / N \*100 % Keterangan:

J : Total nilai jawaban yang diberikan pengguna pada setiap nomor.

N : Jumlah pengguna.

Selanjutnya dilakukan proses perhitungan persentase rata-rata dari setiap poin jawaban, perhitungan yang digunakan adalah menggunakan rumus sebagai berikut:

|  |
| --- |
| Rata-rata = (Persentase ke-1 + ….. + Persentase ke-6) / 6 |

Persentase jawaban responden terhadap kuesioner yaitu, tidak setuju, kurang setuju, cukup, setuju dan sangat setuju dapat dihitung secara keseluruhan dan didefinisikan sebagai nilai rata-rata. Dengan menggunakan rumus perhitungan diatas, perhitungan untuk hasil jawaban pengguna pada setiap pertanyaan pada kuesioner adalah sebagai berikut:

1. Admin

Pada Tabel 4.10 merupakan hasil pengujian kuisioner yang telah dilakukan terhadap 37 wisatawan dari berbagai daerah seperti lombok, sumbawa, taliwang. Sehingga didapatkan hasil bahwa rata-rata yang menjawab sangat setuju adalah 33.3%, setuju 66.6%, Cukup 0%, Kurang setuju 0% dan Tidak Setuju adalah 0%

Keterangan :

Pertanyaan 1 = Apakah tampilan sistem informasi mudah dipahami?

Pertanyaan 2 = Apakah sistem mudah digunakan?

Pertanyaan 3 = Apakah aplikasi ini dapat membantu pihak dalam mengelola wisata di aplikasi?

Pertanyaan 4 = Apakah aplikasi sudah sesuai dengan kebutuhan anda?

Pertanyaan 5 = Apakah aplikasi berjalan dengan baik?

Pertanyaan 6 = Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?

SS = Sangat setuju, S = Setuju, C = Cukup, KS = Kurang Setuju, Ts = Tidak Setuju

Tabel 4.10.Hasil Pengujian Kuisoner Dengan Pegawai Dinas Pariwisata NTB

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pertanyaan | SS | | S | | C | | KS | | TS | |
| total | % | total | % | total | % | total | % | total | % |
| 1 | 0 | 0 | 1 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 1 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| Rata-rata | 33.3% | | 66.6% | | 0 | | 0 | | 0 | |

1. Wisatawan

Pada Tabel 4.16 merupakan hasil pengujian kuisioner yang telah dilakukan terhadap 37 wisatawan dari berbagai daerah seperti lombok, sumbawa, taliwang. Sehingga didapatkan hasil bahwa rata-rata yang menjawab sangat setuju adalah 24.3%, setuju 51.3%, Cukup 22.5%, Kurang setuju 1.35% dan Tidak Setuju adalah 0.45%

Keterangan :

Pertanyaan 1 = Apakah tampilan sistem informasi mudah dipahami?

Pertanyaan 2 = Apakah sistem mudah digunakan?

Pertanyaan 3 = Apakah aplikasi ini dapat membantu wisatawan menemukan wisata alam yang diinginkan dengan mudah?

Pertanyaan 4 = Apakah aplikasi sudah sesuai dengan kebutuhan anda?

Pertanyaan 5 = Apakah aplikasi berjalan dengan baik?

Pertanyaan 6 = Apakah fitur yang disediakan dapat mempermudah pengaksesan aplikasi?

SS = Sangat setuju, S = Setuju, C = Cukup, KS = Kurang Setuju, Ts = Tidak Setuju

Tabel 4.11. Hasil Pengujian Kuisioner dengan Wisatawan

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pertanyaan | SS | | S | | C | | KS | | TS | |
| total | % | total | % | total | % | total | % | total | % |
| 1 | 11 | 29.7 | 18 | 48.6 | 8 | 21.6 | 0 | 0 | 0 | 0 |
| 2 | 11 | 29.7 | 18 | 48.6 | 8 | 21.6 | 0 | 0 | 0 | 0 |
| 3 | 9 | 24.3 | 22 | 59.5 | 6 | 16.2 | 0 | 0 | 0 | 0 |
| 4 | 7 | 18.9 | 17 | 45.9 | 11 | 29.7 | 1 | 2.7 | 1 | 2.7 |
| 5 | 9 | 24.3 | 18 | 48.6 | 9 | 24.3 | 1 | 2.7 | 0 | 0 |
| 6 | 7 | 18.9 | 21 | 56.8 | 8 | 21.6 | 1 | 2.7 | 0 | 0 |
| Rata-rata | 24.3% | | 51.3% | | 22.5% | | 1.35% | | 0.45% | |

Dapat dilihat pada Gambar 4.47 yang merupakan grafik persentase rata-rata responden yang telah menggunakan Aplikasi SIWITA, dapat disimpulkan bahwa 51.30% setuju, 24.30% sangat setuju terkait dengan kemudahan dalam mengakses aplikasi, tampilan mudah di fahami, aplikasi berjalan dengan baik, membantu wisatawan menemukan wisata dengan mudah.

Gambar 4. 47. Grafik Perentase rata-rata responden pengguna aplikasi SIWITA

# BAB V

# PENUTUP

## Kesimpulan

Berdasarkan uraian dan pembahasan yang telah disajikan sebelumnya, berikut ini kesimpulan yang dapat ditarik :

1. Perancangan Sistem Informasi Pemetaan Lokasi Wisata Alam pada tugas akhir ini dapat memberikan informasi yang valid terkait dengan lokasi wisata alam yang ada di pulau Lombok, dapat dibuktikan bahwa dari 140 pantai, 23 air terjun, dan 8 bukit hanya 35 lokasi wisata alam saja yang sudah tervalidasi di Dinas Pariwisata.
2. Terdapat beberapa fitur pendukung yang dapat memudahkan wisatawan mendapatkan lokasi wisata alam seperti fitur pencarian, fitur filter berdasarkan kategori wisata ( gili, desa wisata, pantai, air terjun dan bukit ), fitur rute menuju lokasi wisata alam yang diinginkan, filter kabupaten dan kecamatan. Dapat dibuktikan dari hasil kuisioner bahwa 51.30% setuju dan 24.30% sangat setuju dengan fitur dan kemudahan dari aplikasi SIWITA.
3. Dengan adanya aplikasi ini dapat membantu Dinas Pariwisata dalam mengelola dan memberikan informasi tentang wisata alam di pulau lombok, dapat dibuktikan dengan hasil kuisioner yang diberikan ke pegawai Dinas Pariwisata yaitu 66.6% setuju, dan 33.3% sangat setuju.

## Saran

Adapun saran yang diberikan agar sistem yang telah dibangun menjadi lebih baik lagi yaitu sebagai berikut :

1. Kedepannya aplikasi ini dapat menjangkau seluruh Destinasi Wisata Alam se Nusa Tenggara Barat.
2. Penambahan fitur Bahasa inggris untuk menjangkau wisatawan mancanegara.
3. Aplikasi yang dibuat dapat dilanjutkan sebagai penelitian atau pengabdian masyarakat dengan menghosting secara professional, kemudian pendampingan untuk implementasinya ke masyarakat dan pemerintah daerah.

# DAFTAR PUSTAKA

1. Dinas Pariwisata Provinsi NTB. Angka Kunjungan Wisatawan Ke Nusa Tenggara Barat diakses dalam : <http://www.disbudpar.ntbprov.go.id/angka-kunjungan-wisatawan-ke-ntb/> (10/12/2019. 21.22 WITA)
2. Irfan, Pahrul dkk. Analisa Strategi Pengembangan E-Tourism sebagai Promosi Pariwisata Di Pulau Lombok. ILKOM. Volume 9 Nomor 3. 325-330. 2017.
3. Dinas Pariwisata Provinsi Aceh. Imti Rilis 10 Destinasi Wisata Halal Terbaik Di Indonesia. Di akses dalam : <http://disbudpar.acehprov.go.id/imti-rilis-10-destinasi-wisata-halal-terbaik-di-indonesia/> ( 01/23/2020. 23.00 WITA )
4. Rahmayani, Indah. 2015. Indonesia Raksasa Teknologi Digital Asia. Diakses dalam : <https://kominfo.go.id/content/detail/6095/indonesia-raksasa-teknologi-digital-asia/0/sorotan_media>. (15/12/2019. 13:30 WITA)
5. Putra, Alfian Adestya. Pemanfaatan Aplikasi Google Maps Pada Smartphone Android Sebagai Sarana Belajar Navigasi Mahasiswa Fakultas Ilmu Sosial Universitas Negeri Semarang. Semarang : Fakultas Ilmu Sosial Universitas Negeri Semarang. 2016.
6. Abas, Mohamad Ilyas dkk. Sistem Informasi Geografis Tempat Wisata. Malang : UNMER Malang. 2017.
7. Tanjung, Muhammad Rusdi dkk. Sistem Informasi Geografis Objek Wisata Alam Di Provinsi Sumatera Utara Berbasis Mobile Android. Medan : Universitas Potensi Utama
8. Yuwono, Bambang Dkk. Sistem Informasi Geografis Berbasis Android Untuk Pariwisata Di Daerah Magelang. Yogyakarta: Upn “Veteran” Yogyakarta. 2015.
9. Kusuma, Indra Jati. Sistem Informasi Geografis Parwisata Pulau Lombok Berbasis Android. Yogyakarta : Universitas Islam Negeri Sunan Kalijaga Yogyakarta.2013.
10. Agrarian, Rizki Puta dkk. Pembuatan Aplikasi Mobile GIS Berbasis Android Untuk Informasi Pariwisata Di Kabupaten Gunung Kidul. Semarang : Universitas Diponegoro. 2015.
11. Sahmat, dkk. Sistem Informasi Pemetaan Objek Wisata Lombok Berbasis Web. STMIK Mataram Volume 7 No 1. 2017.
12. Aronoff, S."Geographic Information System a Management Perspective" (Terjemahan). WDL Publication, Ottawa-Canad. 1989.
13. Nandi. Memaksimalkan Potensi Wisata Alam di Jawa Barat. Manajemen Resort dan Leisure. Vol 1, No. 1. 2005.
14. Medina, Dahniar. Aplikasi Tour & Travel Haji dan Umroh di Al-Falah Berbasis Android. J-TIIES. Vol. 1, No. 1. 2017.
15. Budiman. Perancangan Aplikasi Mobile Campus ( m-Campus) (Studi Kasus: AMIK HASS ). Cyber Hass Informatika dan Komputer. Vol 2, No. 2. Hal 25-51. 2018.
16. Elian, A., Shiddiqi, A.M., dan Studiawan, H. LayananInformasi Kereta Api Menggunakan GPS, Google Maps, dan Android. Fakultas Teknologi Informasi Institut TeknologiSepuluh November, Surabaya. 2012.
17. Mengenal Android Studio. [Online]. Available: <https://developer.android.com/studio/intro?hl=id>. Diakses pada (15 Desember 2019)
18. Juansyah, Andi. Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( A- GPS) Dengan Platform Android. Jurnal Ilmiah Komputer Informatika. Edisi 1, Volume 1. 2015.
19. Rahmatullah Pratama, Aditya. “Belajar *Unified Modeling Language*,” Codepolitan, 2019.
20. Nababan Darsono and Windy Nugraha, "*Object Oriented Analysis & desain"* . Tangerang: Universitas Pelita Harapan, 2014.
21. Purnomo, Dwi. "Model Prototyping Pada Pengembangan Sistem Informasi*"* Jurnal Informatika Merdeka Pasuruan. Vol 2. No. 2. 2017.
22. Y. Kustiahningsih, D. Rosa, “Pemrograman Basis Data Berbasis WEB Menggunakan PHP dan Mysql”, Yogyakarta: Graha Ilmu, 2011.
23. Afuan, Lasmedi. Pemanfaatan Framework Codeigniterdalam Pengembangan Sistem Informasi Pendataan Laporan Kerja Praktek Mahasiswa Program Studi Teknik Informatika Unsoed(Codeigniter Framework Used in Information System Development for Student’s Report Data Collection Practices Program in Informatics Engineering Program Study of Unsoed). JUITA Vol 1 Nomor 2 Hal 39-44. 2010.
24. Sibarani, Niko Sumanda, dkk. Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin. Politeknik Negeri Bandung: Bandung. 2018.