

BAB II

TINJAUAN PUSTAKA

2.1 Destinasi Wisata

Menurut Undang-Undang No. 10 Tahun 2009, Bab 1, Pasal 1 tentang Kepariwisata, destinasi wisata adalah kawasan geografis yang berada dalam satu atau lebih wilayah administratif yang di dalamnya terdapat daya tarik wisata, fasilitas umum, fasilitas pariwisata, aksesibilitas, serta masyarakat yang saling terkait dan melengkapi terwujudnya kepariwisataan

(Hary Hermawan, 2017) menjelaskan bahwa destinasi wisata adalah tempat yang dikunjungi lebih lama oleh seseorang daripada tempat-tempat lain selama perjalanan liburan mereka

2.2 Android

Android adalah sistem operasi yang dirancang untuk perangkat mobile berbasis Linux, yang mencakup sistem operasi, *middleware*, dan aplikasi. Android memberikan platform terbuka yang memungkinkan para pengembang untuk menciptakan aplikasi yang inovatif. Sistem operasi Android telah menjadi dasar bagi lebih dari satu miliar *smartphone* dan tablet yang ada di pasaran (Putra, Nugroho & Puspitarini, 2016).

Android menawarkan pendekatan yang komprehensif dalam pembuatan aplikasi. Ini berarti bahwa satu aplikasi Android yang dibangun dapat berfungsi pada berbagai perangkat yang menjalankan sistem operasi Android, termasuk *smartphone*, *smartwatch*, tablet, dan perangkat lainnya. Kemajuan teknologi Android yang cepat juga tidak bisa dipisahkan dari kontribusi *AOSP (Android Open Source Project)* yang bertanggung jawab atas pengembangan sistem operasi Android dan dipimpin secara langsung oleh Google (Irvansyah, Setiawansyah & Muhaqiqin, 2020).

2.3 Algoritma haversine Formula

Algoritma Haversine Formula adalah metode matematis yang digunakan untuk menghitung jarak antara dua titik pada permukaan bumi dengan menggunakan data koordinat geografis garis bujur (*longitude*) dan garis lintang (*latitude*) (Rohmatulloh, Herlambang, & Wibowo, 2022).

Formula Haversine didasarkan pada prinsip trigonometris yang merupakan bagian dari geometri, dan nama "Haversine" berasal dari fungsi trigonometri haversine (*hav*) yang digunakan dalam rumus ini. Meskipun cukup akurat untuk sebagian besar perhitungan, rumus ini tidak memperhitungkan perbedaan ketinggian dan kedalaman di permukaan bumi (Nisar, Wasilah, Arfida & Almani, 2020).

Rumus Haversine Formula adalah sebagai berikut:

$$a : \sin^2(\Delta\text{lat}/2) + \cos(\text{lat1}) * \cos(\text{lat2}) * \sin^2(\Delta\text{long}/2)$$

$$c : 2 * a \tan^2(\sqrt{a}, \sqrt{1-a})$$

$$d : R * c$$

Keterangan:

- lat1 : Garis lintang pertama
- lat2 : Garis lintang kedua
- long1 : Garis bujur pertama
- long2 : Garis bujur kedua
- Δlat : Perbedaan antara garis lintang (*latitude*) kedua dan pertama
- Δlong : Perbedaan antara garis bujur (*longitude*) kedua dan pertama
- R : Radius rata rata bumi (6371) km
- d : jarak

2.4 Flutter

Flutter merupakan sebuah *framework* sumber terbuka yang bersifat lintas platform yang diciptakan oleh Google. *framework* ini digunakan untuk

pengembangan aplikasi *mobile* dengan fokus pada kinerja tinggi. Flutter memungkinkan pembuatan aplikasi yang dapat dihadirkan pada berbagai platform, termasuk Android, iOS, Web, dan Desktop, semuanya dari satu kode sumber tunggal. Pada dasarnya, Flutter menggunakan bahasa pemrograman Dart, yang memiliki keunggulan dalam kemudahan pembelajaran jika sudah terbiasa dengan bahasa pemrograman Java atau JavaScript (Sari & Rahmayuda, 2022)

Antarmuka pengguna dalam Flutter diwujudkan melalui penggunaan *widget*. *Widget* merupakan elemen dasar dalam pembangunan aplikasi Flutter. Secara sederhana, *widget* dapat dianggap sebagai petunjuk untuk berbagai komponen antarmuka pengguna. Ketika *widget-widget* ditempatkan bersama, mereka membentuk struktur berhirarki yang disebut sebagai pohon *widget*. Dalam Flutter, setiap elemen dibangun dengan menggunakan *widget* sebagai konfigurasi, sehingga setiap elemen pada layar pada dasarnya adalah *widget* yang dirender (Nata, Suyadnya, & Khrisne 2022).

2.5 Laravel

Laravel adalah sebuah *framework open-source* dalam bahasa PHP yang dirancang oleh Taylor Otwell. *framework* ini mengadopsi desain arsitektur *model-view-controller* (MVC), yang memberikan kemudahan bagi pengembang dalam membuat aplikasi web, baik yang memiliki skala besar maupun yang lebih kecil. Selain itu, Laravel memiliki baris perintah (CLI) yang disebut Artisan, yang mempermudah pengembang dalam menyederhanakan, mempermudah, dan mempercepat proses pengembangan. Artisan dapat digunakan untuk mengelola migrasi data, basis data, model, controller, dan banyak tugas lainnya (Nata, Suyadnya, & Khrisne 2022).

Laravel menyediakan beragam fitur tambahan yang mendukung pengembangan aplikasi web, seperti sistem otentikasi yang siap digunakan,

routing yang bersih, sistem templating yang kuat dengan Blade, serta ORM yang andal dengan Eloquent.

2.6 Mysql

MySQL adalah perangkat lunak sistem manajemen basis data relasional (RDBMS) yang multi utas dan multi pengguna. MySQL merupakan turunan dari konsep utama dalam basis data, yaitu pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dilakukan dengan mudah dan otomatis. MySQL diciptakan oleh Michael "Monty" Widenius pada tahun 1979, seorang programmer komputer asal Swedia. Widenius mengembangkan sistem basis data sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing (Sadali, & Putra, 2020).

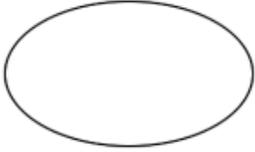
2.7 Use Case Diagram

Use case diagram adalah diagram yang menggambarkan fungsi-fungsi yang dapat dilakukan oleh suatu perangkat lunak. *Use case diagram* juga dapat digunakan untuk memahami bagaimana sistem tersebut bekerja (Fauzan, Siahaan, Rochimah, & Triandini, 2021).

Berikut adalah komponen komponen yang terdapat pada *use case diagram*.

Tabel 2.1 Komponen *Use Case Diagram*

Simbol	Nama	Deskripsi
	<i>Actors</i>	Mewakili peran orang/perangkat yang menggunakan sistem yang bertindak atau menggunakan fungsi sesuai dengan yang dideskripsikan.

	<i>Use Case</i>	Gambaran fungsional pada sistem yang dapat digunakan oleh aktor.
	<i>Association</i>	Menunjukkan hubungan antara aktor dan <i>use case</i> .
	<i>Include</i>	Menunjukkan bahwa sebuah <i>use case</i> merupakan suatu fungsionalitas dari <i>use case</i> lainnya.
	<i>Extend</i>	Menunjukkan bahwa sebuah <i>use case</i> merupakan suatu tambahan fungsionalitas dari <i>use case</i> lainnya.

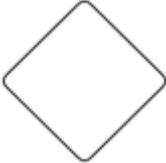
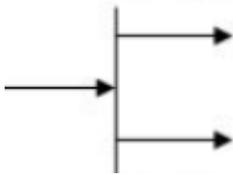
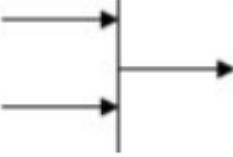
2.8 *Activity Diagram*

Activity diagram adalah diagram yang menggambarkan aliran kerja dari suatu sistem atau proses. *Activity diagram* dapat digunakan untuk menggambarkan logika internal dari operasi yang kompleks (Sulaiman, Ahmad, & Ahmad, 2019).

Berikut adalah komponen komponen yang terdapat pada *Activity diagram*.

Tabel 2.2 Komponen *Activity diagram*

Simbol	Nama	Deskripsi
	<i>Initial State</i>	Melambangkan awal dimulainya suatu aliran kerja dari sistem.

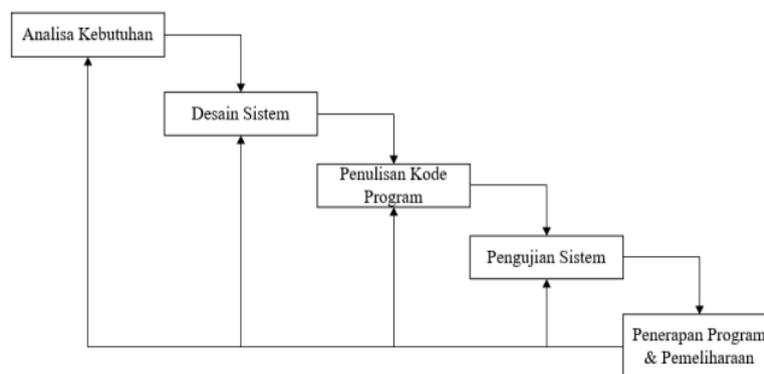
	<i>Final State</i>	Melambangkan bagian akhir dari suatu aliran kerja dari sistem.
	<i>Swimlane</i>	Melambangkan pemisahan organisasi bisnis yang bertanggung jawab pada suatu aktivitas.
	<i>Activity</i>	Melambangkan suatu aktivitas yang dilakukan oleh sistem
	<i>Decision</i>	Melambangkan pilihan kondisi atau percabangan dimana terdapat lebih dari satu pilihan aktivitas.
	<i>Transition</i>	Melambangkan hubungan aktivitas selanjutnya setelah aktivitas sebelumnya.
	<i>Synchronization Fork</i>	Melambangkan pemecahan tingkah laku menjadi aktivitas paralel.
	<i>Synchronization Join</i>	Melambangkan penggabungan kembali aktivitas yang paralel.

2.9 Entity Relationship Diagram

Entity relationship diagram (ERD) merupakan suatu diagram yang dimanfaatkan untuk merancang hubungan antar tabel dalam sebuah basis data. ERD juga dapat digunakan untuk menggambarkan kebutuhan data dari sebuah organisasi.

2.10 Waterfall

Ada beberapa model SDLC (*Software Development Life Cycle*). Salah satunya adalah model *waterfall*, Metode *waterfall* adalah model pengembangan perangkat lunak klasik yang mengikuti pendekatan sistematis dan berurutan, *waterfall merupakan metode yang sangat populer dan sering digunakan, proses pengembangan sistem dibagi menjadi beberapa langkah. Dalam proyek besar, setiap langkah dijalankan oleh tim yang berbeda (Kurniawan, Apriliah, Kurniawan, & Firmansyah, 2020).*



Gambar 2.1 Metode *Waterfall*

Berdasarkan gambar 2.1 terdapat penjelasan dari setiap fase proses dari metode *waterfall* sebagai berikut:

1. Analisa kebutuhan: Tahap ini bertujuan untuk mengumpulkan dan mendefinisikan kebutuhan pengguna dari sistem perangkat lunak yang akan dikembangkan.
2. Desain Sistem: Tahap ini bertujuan untuk mendesain sistem perangkat lunak yang akan dikembangkan.
3. Penulisan Code Program: Tahap ini bertujuan untuk membangun sistem perangkat lunak berdasarkan desain yang telah dibuat.
4. Pengujian Sistem: Tahap ini bertujuan untuk memastikan bahwa sistem perangkat lunak memenuhi kebutuhan pengguna.

5. Pemeliharaan: Tahap ini bertujuan untuk memperbaiki kesalahan dan meningkatkan sistem perangkat lunak setelah sistem tersebut dirilis ke pengguna.

2.11 Black Box Testing

Black box testing merupakan metode pengujian yang hanya memeriksa hasil eksekusi perangkat lunak melalui penggunaan data uji, serta memeriksa fungsionalitasnya. Dalam analogi ini, mirip dengan melihat suatu kotak hitam, di mana kita hanya dapat melihat penampilan luar kotak tersebut tanpa mengetahui apa yang ada di dalamnya. Sama halnya dengan pengujian black box, penilaian dilakukan hanya terhadap tampilan luar (antarmuka) tanpa memperhatikan rincian internal yang terjadi (Astuti, 2018).

2.12 Penelitian Terkait

Dalam pembuatan penelitian ini, penulis merujuk pada sejumlah penelitian terkait yang telah menginspirasi pemikiran penulis mengenai masalah yang sedang diteliti. Penelitian terkait dapat dilihat pada Tabel 2.3

Tabel 2.3 Penelitian terkait

Judul	Penulis, Tahun	Metode	Hasil
SISTEM INFORMASI GEOGRAFIS RUMAH KOST BERBASIS WEB DENGAN MENGGUNAKAN METODE HAVERSINE FORMULA	Muh. Andi Saputra. (2020)	<i>Waterfall</i>	Aplikasi sistem informasi geografis berbasis web yang menampilkan data rumah kost terdekat dari kampus yang sudah ditentukan lokasinya

Rancang Bangun Sistem Informasi Pariwisata Berbasis Web Sebagai Media Promosi Pada Kabupaten Lampung Barat	Kania Latifah, Eka Nisatul Mukaroh. (2021)	<i>Prototype</i>	Aplikasi sistem informasi pariwisata berbasis web untuk mempromosikan Kabupaten Lampung Barat. dengan fitur-fitur seperti berita, informasi Dinas Pemuda dan Olahraga, profil, galeri, dan informasi tempat wisata.
APLIKASI RESTFULL PADA SISTEM INFORMASI GEOGRAFIS PARIWISATA KOTA BANDAR LAMPUNG	Melvitasari Tinambunan, Sanriomi Sintaro. (2021)	<i>Waterfall</i>	Aplikasi Sistem Informasi Geografis (SIG) Zona Wisata Kota Bandar Lampung Berbasis Android, yang memuat data lokasi wisata yang ada di kota bandar lampung.
Sistem Informasi Geografis Tempat Penting Pantura Tegal Semarang (SIG TEPI Pantura Terang) Berbasis Web	M. Dwi Setyo, S.Kom., Setyoningsih Wibowo, S.T., M.kom. (2018)	<i>Waterfall</i>	Aplikasi Sistem Informasi Geografis (SIG) Pantura, Tegal, Semarang bertujuan untuk memetakan lokasi penting seperti SPBU, Tambal Ban, Bengkel, Rest Area, Kantor dan Pos Polisi, Puskesmas, serta Rumah Sakit.

PENERAPAN FORMULA HAVERSINE PADA SISTEM INFORMASI GEOGRAFIS PENCARIAN JARAK TERDEKAT LOKASI LAPANGAN FUTSAL	Yulianto, Ramadiani , Awang Harsa Kridalaksana. (2018)		Aplikasi Sistem Informasi Geografis (SIG) berbasis web yang memungkinkan pencarian lokasi lapangan futsal terdekat dengan mudah dan akurat.
--	--	--	---