

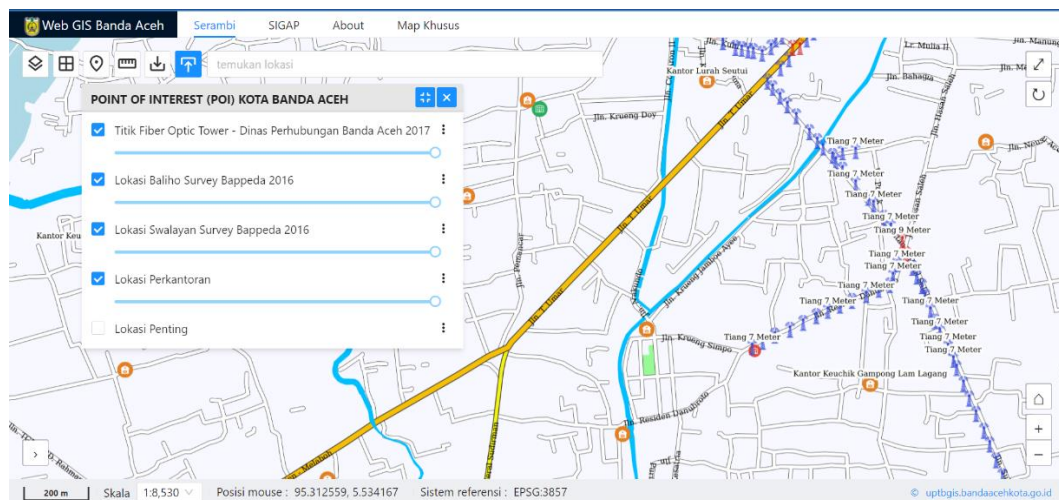
## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

##### **2.1.1 Sistem Informasi Geografis (SIG)**

Sistem Informasi Geografis (SIG) merupakan suatu bentuk sistem yang menggabungkan teknologi informasi dengan data berbasis lokasi atau geografis. Konsep Sistem Informasi Geografis atau SIG memungkinkan pengguna untuk mengumpulkan, menyimpan, mengelola, menganalisis, dan memvisualisasikan data yang terkait dengan lokasi geografis. (Wibowo dkk., 2015) Menurut penelitian tersebut, Sistem Informasi Geografis memberikan manfaat dalam konteks pengambilan keputusan, terutama pada aspek-aspek yang memiliki dimensi spasial atau keruangan, kemudian menurut (Bafdal dkk., 2011) dalam bukunya yang berjudul “Buku Ajar Sistem Informasi Geografis , Edisi 1”, Sistem Informasi Geografis (SIG) adalah sebuah sistem informasi yang menggabungkan tiga elemen utama: sistem, informasi, dan unsur geografis. SIG, seperti yang telah dijelaskan sebelumnya, merupakan bentuk dari sistem informasi yang meliputi unsur "Geografis" tambahan. Dengan fokus pada pengertian sistem informasi, SIG bisa dianggap sebagai suatu kesatuan formal yang terdiri dari berbagai komponen fisik dan logis yang berkaitan dengan objek-objek yang ada di permukaan bumi. SIG juga bisa dilihat sebagai jenis perangkat lunak yang memungkinkan pengguna untuk memasukkan, menyimpan, mengelola, menampilkan, dan menganalisis informasi geografis beserta atribut-atributnya.



(Sumber: [uptbgis.bandaacehkota.go.id/](http://uptbgis.bandaacehkota.go.id/))

**Gambar 2.1** Contoh Sistem Informasi Geografis Kota Banda Aceh

Penggunaan SIG dalam pengambilan keputusan sangat terkait dengan analisis spasial yang mendalam. Dengan memanfaatkan data geografis, SIG memungkinkan pengguna untuk mengidentifikasi pola, tren, dan hubungan spasial antar-objek atau fenomena. Dengan demikian, informasi yang diperoleh dapat memberikan wawasan yang lebih baik untuk mendukung keputusan terkait perencanaan wilayah, distribusi populasi, dan perubahan lingkungan. Lebih lanjut, SIG memfasilitasi visualisasi data secara kontekstual pada peta atau grafik spasial, memudahkan pengguna untuk memahami informasi dengan cara yang lebih intuitif (Giaoutzi & Papadopoulou, 2021).

Keunggulan SIG juga terletak pada kemampuannya untuk mengintegrasikan data dari berbagai disiplin ilmu. Integrasi data geografis dengan informasi sosial, ekonomi, atau lingkungan memungkinkan pengguna untuk memperoleh pemahaman yang lebih komprehensif. Oleh karena itu, SIG mendukung pengambilan keputusan yang informasional dan kontekstual, terutama dalam situasi yang melibatkan kompleksitas variabel dan factor (Syaiful Huda, 2019).

Dengan demikian, dari penelitian-penelitian yang sebelumnya dapat diambil kesimpulan yang menggarisbawahi bahwa SIG bukan hanya sebuah alat teknologi informasi, melainkan suatu sistem yang mampu meningkatkan efisiensi dan keakuratan proses pengambilan keputusan dengan memberikan konteks spasial dan informasi yang dapat divisualisasikan dengan jelas.

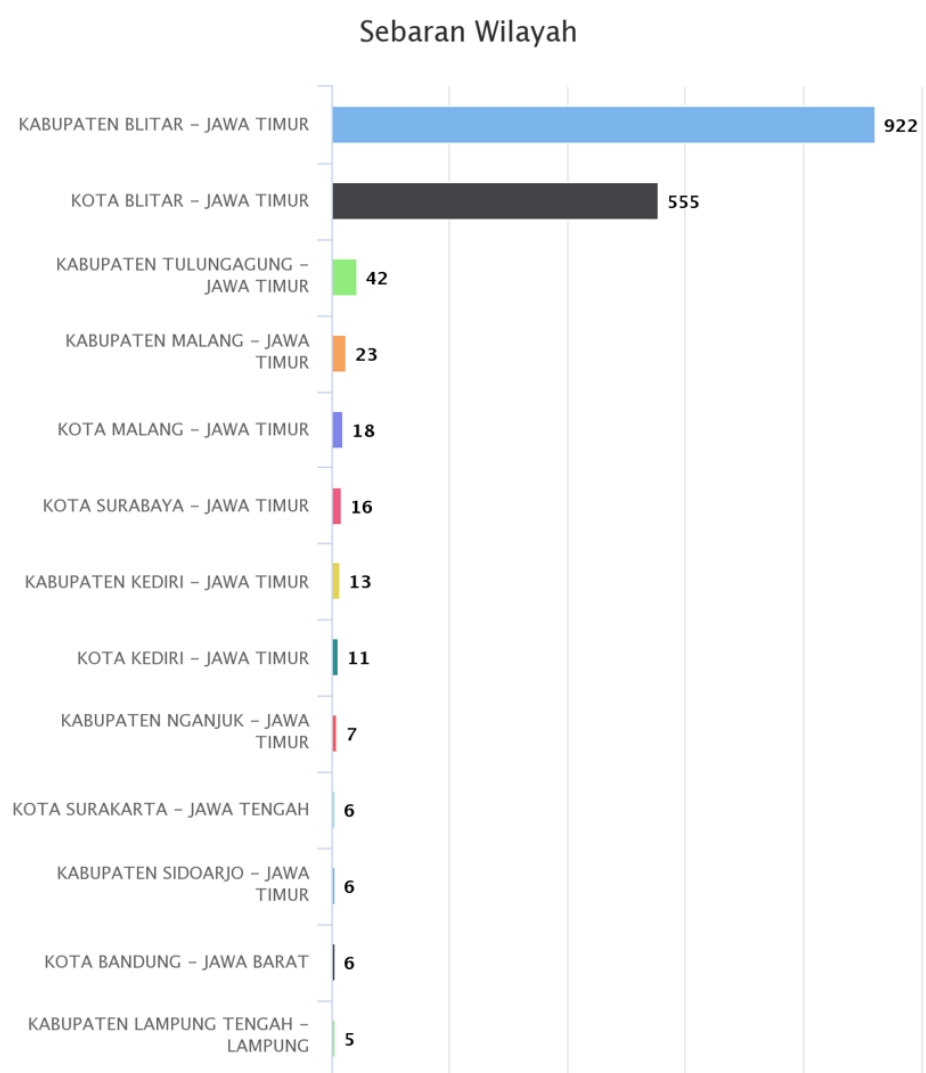
### **2.1.2 Pemetaan Mahasiswa**

Menurut penelitian yang telah dilakukan oleh Masykur (2014), Diah Ayu dkk. (2020) dan Kasyidi dkk. (2021), pemetaan mahasiswa merupakan suatu metode inovatif yang bertujuan untuk mengumpulkan, mengelola, dan menganalisis data spasial mahasiswa melalui penerapan Sistem Informasi Geografis (SIG). Pendekatan ini memberikan sudut pandang baru dalam pengelolaan informasi di lingkungan pendidikan, di mana data mahasiswa tidak hanya diolah secara konvensional, tetapi juga divisualisasikan dalam bentuk peta geografis sesuai dengan data spasial yang terkait dengan setiap mahasiswa.

Proses pemetaan melibatkan pengumpulan data spasial, yang mencakup lokasi tinggal, rute perjalanan harian, dan interaksi sosial mahasiswa. Dengan memanfaatkan teknologi SIG, data tersebut dapat disajikan secara visual dan kontekstual, memungkinkan institusi pendidikan untuk memahami lebih baik pola persebaran mahasiswa di berbagai wilayah kampus. Keberhasilan pemetaan ini dapat memberikan manfaat strategis bagi institusi atau lembaga pendidikan, pemetaan dapat menjadi alat bantu yang efektif dalam pengambilan keputusan terkait strategi promosi kampus. Dengan memahami secara mendalam pola

persebaran mahasiswa, lembaga pendidikan dapat mengidentifikasi area potensial untuk meningkatkan visibilitas dan daya tarik kampus.

Di Universitas Islam Balitar sendiri data pendaftar maupun data mahasiswa disajikan dalam bentuk grafik batang yang menunjukkan sebaran mahasiswa yang dikelompokkan berdasarkan kabupaten dan kota asal mereka.



**Gambar 2.2** Grafik Jumlah Pendaftar

Dari gambar di atas data pendaftar dikelompokkan berdasarkan kabupaten atau kota asal mereka dan mayoritas pendaftar berasal dari Kabupaten Blitar dengan pendaftar sebanyak 922 orang kemudian disusul oleh Kota Blitar dengan 555 orang

pendaftar, untuk kota lain di luar blitar, Jumlah terbanyak berasal dari kota atau kabupaten tetangga, seperti Kabupaten Tulungagung dengan 42 orang pendaftar dan Kabupaten Malang dengan 23 pendaftar, dan untuk Kota Malang ada 18 Pendaftar.

### 2.1.3 *JavaScript*

*JavaScript*, sebagai bahasa pemrograman web yang tergolong dalam *Client Side Programming Language*, artinya *JavaScript* melakukan pemrosesan kode di sisi klien atau perangkat pengguna. Dalam konteks ini, aplikasi klien merujuk pada peramban web seperti *Google Chrome*, *Mozilla Firefox*, *Opera Mini*, dan sejenisnya (Pahlevi dkk., 2018).

Keunggulan *JavaScript* karena menjalankan *Client Side Programming* terletak pada kemampuannya untuk memberikan pengalaman pengguna yang lebih dinamis dan interaktif. Dengan kode yang dieksekusi langsung di perangkat pengguna, halaman web dapat merespons dengan cepat terhadap aksi pengguna, seperti klik tombol atau pengisian formulir. Selain itu, *JavaScript* juga memungkinkan manipulasi DOM (*Document Object Model*) secara langsung, memungkinkan perubahan dinamis pada struktur dan tampilan halaman web tanpa perlu me-refresh seluruh halaman.

Dalam praktiknya, kehadiran *JavaScript* dalam *Client Side Programming* juga membantu mengurangi beban server, karena sebagian besar pemrosesan dapat ditangani di perangkat pengguna. Ini berarti bahwa aplikasi web dapat berjalan lebih efisien dan merespons dengan lebih cepat tanpa harus terus-menerus melakukan permintaan ke server.

#### 2.1.4 *ReactJS*

*ReactJS*, sebuah perpustakaan *JavaScript* yang menjadi salah satu pilar utama dalam pengembangan antarmuka pengguna (UI) pada aplikasi web modern. Dikembangkan oleh Facebook, *ReactJS* memperkenalkan paradigma pengembangan yang revolusioner dengan fokus pada konsep komponen. Dalam ekosistem React, aplikasi web dibangun dari potongan-potongan UI yang dapat diisolasi yang disebut komponen, yang memiliki logika dan tampilan sendiri masing-masing komponen. JSX, ekstensi sintaksis mirip HTML, memungkinkan penulisan struktur UI dengan cara yang nyaman di dalam kode *JavaScript* (Nasution & Iswari, 2021).

Satu aspek kunci yang membedakan React adalah penggunaannya terhadap Virtual DOM. Dengan konsep ini, setiap perubahan pada data menghasilkan pembuatan Virtual DOM, dan React secara cerdas memperbarui DOM hanya untuk perubahan yang diperlukan. Ini menghasilkan kinerja yang lebih efisien dan responsif.

*ReactJS* juga menekankan penggunaan *state* dan *lifecycle methods*. *State* mewakili data internal komponen, dan setiap kali *state* berubah, *React* memperbarui *UI* yang sesuai. *Lifecycle methods* memungkinkan penggunaan sebuah komponen yang sama untuk digunakan lagi di tampilan atau antarmuka yang lain, dengan keluaran yang berbeda juga, meskipun pada dasarnya komponen yang digunakan sama, namun dapat digunakan untuk beberapa skenario tampilan antarmuka, ini memberikan kontrol yang tepat dalam pengelolaan aplikasi terkait efisiensi penulisan baris kode.

Meskipun *ReactJS* memiliki kelebihan yang signifikan, seperti efisiensi dengan Virtual DOM dan kemudahan pengelolaan UI dengan komponen, tidak dapat dihindari bahwa ada tantangan lebih dalam pembelajaran untuk memahami konsep seperti *State*, *Props*, dan *Virtual DOM*. Meski demikian, daya tarik dan kemudahan pengembangan yang ditawarkan oleh *ReactJS* menjadikannya pilihan utama untuk banyak proyek pengembangan web modern.

### **2.1.5 API (*Application Programming Interface*)**

Programmer memanfaatkan API untuk mengintegrasikan atau memanfaatkan fitur-fitur suatu perangkat lunak dalam pembangunan aplikasi yang sedang mereka kembangkan. Melalui API, programmer dapat mengakses fungsionalitas dari perangkat lunak lain tanpa harus memahami secara detail bagaimana implementasinya dilakukan. Sebagai contoh, API memungkinkan seorang pengembang aplikasi mobile menggunakan layanan peta dari penyedia tertentu tanpa perlu memiliki pengetahuan mendalam tentang algoritma pemetaan yang digunakan oleh penyedia tersebut.

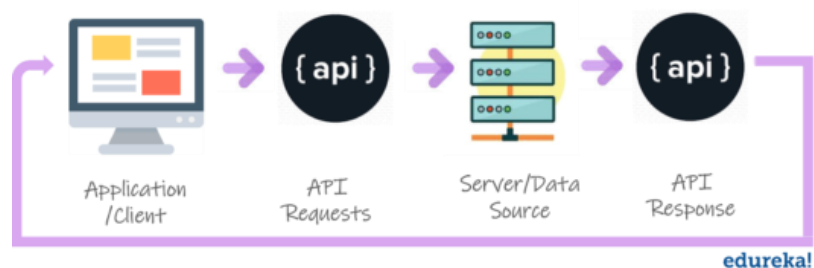
Dokumentasi API yang baik menyajikan informasi secara rinci tentang cara menggunakan setiap fungsi atau layanan yang tersedia. Ini mencakup parameter yang diperlukan, nilai keluaran yang diharapkan, serta contoh-contoh penggunaan. Dengan adanya dokumentasi yang jelas, programmer dapat menghemat waktu dan upaya dalam mengintegrasikan fitur-fitur yang mereka perlukan tanpa harus mendalaminya secara menyeluruh.

Pada intinya, API berfungsi sebagai penghubung yang efisien antara perangkat lunak, menyediakan lapisan abstraksi yang memungkinkan kolaborasi

antarmuka dan menyederhanakan proses pengembangan aplikasi dengan memungkinkan penggunaan fitur-fitur dari perangkat lunak yang ada melalui antarmuka yang terdefinisi dengan baik.

### 2.1.6 MapBox API

API, atau *Application Programming Interface*, merupakan sekumpulan dokumen yang menggambarkan interface, fungsi, kelas, dan elemen-elemen lain dari suatu perangkat lunak. Sebuah API berperan sebagai jembatan yang menghubungkan perangkat lunak satu dengan yang lain, memberikan akses yang terstruktur dan terdokumentasi untuk digunakan oleh para pengembang perangkat lunak. Dengan kata lain, API memungkinkan perangkat lunak yang berbeda untuk saling berkomunikasi dan berinteraksi secara efisien (Ramadhani, 2018).



(Sumber: [edureka.co/blog/what-is-api-testing](https://edureka.co/blog/what-is-api-testing))

**Gambar 2.3** Ilustrasi Pemanggilan API

*Mapbox API* merupakan layanan web yang menyediakan beragam informasi terkait geocoding dan arah perjalanan (*direction*) antara dua node atau titik. Penggunaan *Mapbox API* dipilih karena kemudahan integrasi yang ditawarkan, serta kelengkapan informasi yang dihasilkannya. API ini memfasilitasi pengguna dengan data yang kaya, mencakup jarak tempuh, koordinat, nama jalan, hingga titik-titik perantara (*waypoints*) (Adil dkk., 2023).

### 2.1.7 *Google Firebase*

*Google Firebase* adalah sebuah layanan database *NoSQL* yang bersifat *real-time*, fitur utama yang digunakan sebagai penyimpanan data pada penelitian ini adalah *Firestore* yang menggunakan struktur data dalam format *JSON (JavaScript Notation)* (Nugraha, 2022). *Database* ini memudahkan akses melalui kode web pada aplikasi *hybrid* dengan cepat dan efisien. Selain keunggulan *real-time database*, *Firebase* juga menyediakan berbagai fitur tambahan yang memperkaya pengalaman pengembangan aplikasi. Salah satu fitur yang disediakan adalah sistem autentikasi pengguna yang memungkinkan proses *login* menggunakan kombinasi *email* dan *password* dan juga yang memungkinkan pengguna melakukan *login* menggunakan email, akun *facebook* atau akun *google*. Hal ini memberikan tingkat keamanan dan pengelolaan pengguna yang baik pada aplikasi yang dikembangkan (Ilhami, 2018).

### 2.1.8 *Closed Beta Testing*

*Closed beta testing* merupakan salah satu metode pengujian yang dilakukan pada tahap akhir pengembangan perangkat lunak atau aplikasi sebelum peluncuran resmi. Dalam metode ini, aplikasi atau produk yang sedang dikembangkan hanya tersedia untuk sekelompok pengguna terbatas atau terpilih yang telah diundang atau memiliki akses tertutup. Peserta dalam *closed beta testing* ini biasanya terdiri dari sekelompok pengguna yang dianggap sebagai calon pengguna akhir atau representatif dari target pasar atau pengguna potensial (Susanto dkk., 2020).

Proses *closed beta testing* dimulai dengan pemilihan peserta beta yang sesuai dengan kriteria yang telah ditentukan oleh tim pengembang. Peserta beta ini

kemudian diberikan akses terbatas untuk mengunduh dan menggunakan aplikasi yang sedang diuji. Mereka diminta untuk menjalankan berbagai fitur, mencoba berbagai skenario penggunaan, dan memberikan umpan balik langsung tentang pengalaman pengguna mereka.

$$Y = \frac{x}{Skor\ Ideal} \times 100\% \dots \dots \dots (2.1)$$

**Keterangan:**

$Y$  = Nilai persentase dari pengujian

$X$  = Jumlah perkalian nilai responden dengan skala

$Skor\ Ideal$  = Jumlah responden x jumlah skala penilaian

Rumus 2.1 di atas akan digunakan dalam menghitung nilai persentase pengujian berdasarkan penilaian dari responden untuk mengukur apakah aplikasi sudah layak diluncurkan, penilaian yang diterima dari peserta beta ini sangat berharga bagi pengembang dalam memperbaiki bug, meningkatkan kinerja, dan menyesuaikan fitur aplikasi sebelum peluncuran resmi. Pengembang melakukan pemantauan aktif terhadap kinerja aplikasi, mencatat laporan bug, dan menganalisis umpan balik yang diterima untuk menentukan langkah-langkah perbaikan yang diperlukan.

Dengan *closed beta testing*, pengembang dapat memperoleh informasi yang penting terkait kinerja, keandalan, dan pengalaman pengguna aplikasi sebelum diluncurkan secara luas. Ini memungkinkan mereka untuk memastikan bahwa aplikasi dapat memenuhi ekspektasi pengguna dan memberikan pengalaman pengguna yang baik setelah peluncuran resmi.

### 2.1.9 *Blackbox Testing*

*Black-box testing* adalah metode pengujian perangkat lunak yang fokus pada fungsionalitas eksternal suatu sistem tanpa memperhatikan struktur internal atau implementasi kode. Dalam black-box testing, pengujian dilakukan dengan melihat sistem sebagai suatu "kotak hitam" di mana pengujian dilakukan berdasarkan input yang diberikan dan output yang dihasilkan tanpa memperhatikan bagaimana sistem mencapai hasil tersebut. Pada tahap *black-box testing*, penguji tidak memiliki pengetahuan terperinci tentang struktur atau logika internal dari program yang diuji. Tujuan utamanya adalah untuk memastikan bahwa sistem beroperasi sesuai dengan spesifikasi yang telah ditentukan dan mampu memenuhi kebutuhan fungsional yang diinginkan oleh pengguna (Arwaz dkk., 2019).

Proses *black-box testing* mencakup identifikasi input yang valid dan invalid, pengujian fungsionalitas sistem, pengujian batasan-batasan, serta verifikasi apakah output sesuai dengan yang diharapkan. Metode ini memungkinkan penguji untuk mendapatkan gambaran holistik tentang seberapa baik sistem dapat menjawab input dan situasi yang mungkin terjadi dari perspektif pengguna akhir.

$$Akurasi = \frac{\Sigma \text{Pengujian Berhasil}}{\Sigma \text{Total Berhasil}} \times 100\% \dots \dots \dots (2.2)$$

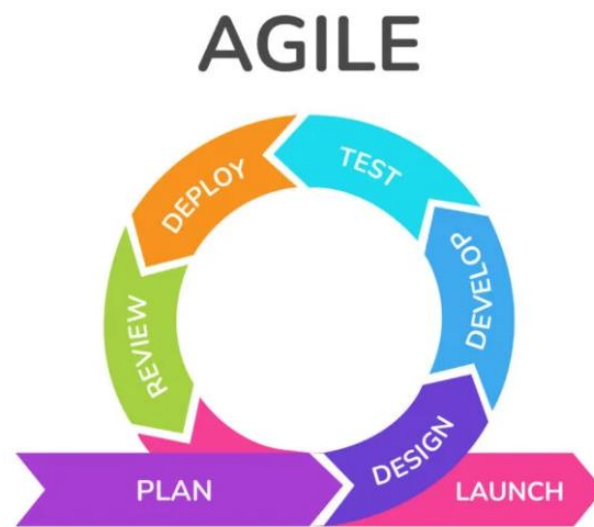
Dengan menggunakan *black-box testing*, pengembang dan penguji dapat mengevaluasi apakah sistem berfungsi dengan benar dan memenuhi kriteria kualitas yang diharapkan, tanpa memerlukan pengetahuan mendalam tentang implementasi internalnya. Hal ini memungkinkan untuk mengidentifikasi bug atau masalah fungsional pada tingkat tinggi, yang kemudian dapat diperbaiki sebelum mencapai pengguna akhir. Dengan rencana pengujian yang sudah dibuat, hasil

pengujian akan dihitung menggunakan persamaan 2.2 di atas, untuk mengetahui persentase keberhasilan dari tiap fungsionalitas yang diuji.

#### **2.1.10 Model Agile**

Model *Agile* adalah sebuah kerangka konseptual dalam pengembangan perangkat lunak yang berpusat pada pengulangan proses. Dalam model ini, aturan dan solusi yang telah disepakati akan diterapkan dalam kerja sama yang terstruktur dan terorganisir antar kelompok. *Agile* adalah pendekatan pengembangan perangkat lunak yang bersifat jangka pendek, di mana tugas-tugas dibagi menjadi iterasi yang lebih kecil, yang sering disebut sebagai "*sprint*". Tidak seperti perencanaan jangka panjang yang kompleks, dalam Agile, rencana mengenai kebutuhan proyek, jumlah sprint, durasi, dan cakupan setiap sprint tidak harus ditentukan di awal (Paksi dkk., 2023).

Jumlah *sprint* yang sebenarnya dapat disesuaikan seiring berjalannya proyek berdasarkan perkembangan, masukan pengguna, dan perubahan kebutuhan. Dalam praktiknya, tim Agile sering kali melakukan perencanaan sprint secara iteratif, dengan menetapkan jumlah sprint untuk beberapa iterasi pertama dan kemudian meninjau kembali dan menyesuaikan rencana berdasarkan hasil analisa yang didapat dari setiap sprint (Suhari dkk., 2022). Ini memungkinkan tim untuk tetap responsif terhadap perubahan kebutuhan dan mengoptimalkan pengembangan produk secara iteratif.



**Gambar 2.4** Model Agile

Tahapan dalam model pengembangan perangkat lunak Agile, seperti yang diilustrasikan pada Gambar 2.4, dimulai dengan tahap perencanaan di mana pengembang dan klien berkolaborasi untuk mendefinisikan kebutuhan proyek. Setelah mengidentifikasi kebutuhan tersebut, langkah selanjutnya adalah tahap perancangan atau desain, di mana alur kerja sistem direncanakan menggunakan berbagai diagram untuk menunjukkan fitur dan penerapan *UI/UX*. Setelah rancangan dibuat, proses pengembangan atau pembangunan dimulai, di mana programmer melakukan pengkodean perangkat lunak. Tahap pengembangan ini dilanjutkan dengan tahap *testing* untuk penyempurnaan produk dengan mendeteksi dan memperbaiki bug serta menjaga kualitas perangkat lunak dan memastikan fungsionalitas yang dibutuhkan dapat berjalan dengan baik, pada penelitian ini tahap testing dilakukan pengujian website kepada pakar atau ahli di bidang teknologi informasi.

Perangkat lunak selanjutnya memasuki tahap *deployment* ke penyedia layanan *hosting* sebagai persiapan untuk melanjutkan ke tahap review yang akan melibatkan pengujian dengan calon pengguna melalui metode *closed beta testing* untuk mengumpulkan umpan balik terkait pengalaman penggunaan website dari calon pengguna untuk menentukan langkah selanjutnya apakah akan berlanjut ke tahap peluncuran atau meneruskan proses ke *sprint* selanjutnya

Dalam konteks model Agile, tahapan ini menunjukkan pendekatan iteratif dalam pengembangan perangkat lunak, di mana setiap tahap berkembang berdasarkan umpan balik dan pembelajaran yang diperoleh dari tahap sebelumnya. Ini memungkinkan fleksibilitas dalam menanggapi perubahan kebutuhan dan memastikan kualitas produk yang lebih baik.

## 2.2 Kajian Penelitian

Tabel 2.1 Penelitian Terdahulu

No	Judul Penelitian	Penulis	Tahun Terbit	Jurnal	Kontribusi Untuk Penelitian
1	<i>Geographic Information System of Public Complaint Testing Based On Mobile Web (Public Complaint)</i>	M. Mahendra, I. Piarsa, D. Putra Githa	2018	Lontar Komputer: Jurnal Ilmiah Teknologi Informasi	Sebagai referensi untuk metode pengujian sistem menggunakan <i>Blackbox Testing</i> .
2	<i>Geographic Information System (GIS) as an Information Media in the Field of Environmental Health</i>	M. Chaniago, H. Taki	2022	Journal of Applied Geospatial Information	Referensi penggunaan SIG dalam menyediakan informasi dalam bentuk visualisasi peta yang dapat membantu dalam proses pengambilan keputusan.
3	Sistem Informasi Geografis Pemetaan Warga Kurang	M. Ali Hasymi, A. Faisol, F. Ariwibisono	2021	JATI (Jurnal Mahasiswa Teknik Informatika)	Sebagai referensi untuk pemetaan dengan subjek manusia yang memiliki beberapa atribut menggunakan

No	Judul Penelitian	Penulis	Tahun Terbit	Jurnal	Kontribusi Untuk Penelitian
	Mampu Di Kelurahan Karang Besuki Menggunakan Metode <i>K-Means Clustering</i>				sistem informasi geografis.
4	Pengenalan <i>Google Firebase</i> Untuk Hybrid Mobile Apps Berbasis <i>Cordova</i>	M. Ilhami	2018	Jurnal Ilmiah IT CIDA	Referensi tentang penggunaan <i>Google Firebase</i> sebagai database.
5	Implementasi Spasial Algoritma <i>Harvesine</i> pada <i>Mapbox API</i> untuk Pemetaan Pariwisata	A. Adil, B. Triwjoyo, I. Dharma	2023	Jurnal Bumigora Information Technology (BITe)	Referensi penggunaan <i>Mapbox API</i> sebagai platform pemetaan.
6	Penerapan <i>React JS</i> Pada Pengembangan <i>FrontEnd</i> Aplikasi Startup <i>Ubaform</i>	Nasution, L. Iswari	2021	Automata Diseminasi Tugas Akhir Mahasiswa	Referensi dalam penggunaan library <i>ReactJS</i> dalam pengembangan aplikasi berbasis website.
7	Implementasi Sistem Informasi Geografis Menggunakan Google Maps Api Dalam Pemetaan Asal Mahasiswa	F. Masykur	2014	Jurnal SIMETRIS	Referensi pemetaan mahasiswa yang dapat memberikan beberapa manfaat untuk kampus.
8	Sistem Informasi Kepegawaian Menggunakan Metode Agile Development di CV. Angkasa Raya	Suhari, A. Faqih, FM. Basyar	2022	Jurnal Teknologi dan Informasi	Referensi penggunaan agile dalam pengembangan sebuah sistem.
9	Pengujian Beta Pada Aplikasi <i>Virtual Tour</i> Destinasi Wisata Di Kabupaten Batang (Study	E. Susanto, M. Maulana, S. Binabar	2020	RISTEK: Jurnal Riset, Inovasi dan Teknologi Kabupaten Batang	Referensi cara pengujian menggunakan <i>closed beta testing</i> .

No	Judul Penelitian	Penulis	Tahun Terbit	Jurnal	Kontribusi Untuk Penelitian
10	Kasus: Bandar Ecopark) Perancangan Sistem Informasi Manajemen Keuangan Menggunakan Metode Scrum (Studi Kasus: CV Kurnia Jaya)	A. Kusuma	2021	Seminar Nasional & Call Paper Fakultas Sains dan Teknologi	Referensi penggunaan <i>closed beta testing</i> dalam pengujian perangkat lunak.

Penelitian pertama yang dibahas adalah “*Geographic Information System of Public Complaint Testing Based On Mobile Web*” yang dilakukan oleh M. Mahendra, I. Piarsa, dan D. Putra Githa pada tahun 2018. Hasil pengujian sistem menggunakan metode *blackbox testing* menunjukkan kinerja optimal, memvalidasi efisiensi layanan pengaduan 24 jumlah berbasis lokasi dengan fitur yang mempermudah penanganan oleh pihak berwenang (Mahendra dkk., 2018).

Artikel kedua membahas peran *Geographic Information System (GIS)* dalam bidang kesehatan lingkungan. M. Chaniago dan H. Taki (2022) menyoroti bahwa GIS dapat memberikan informasi terkini tentang kondisi kesehatan masyarakat, khususnya dalam pencegahan penyakit melalui penyediaan data epidemiologi yang cepat diakses (Chaniago & Taki, 2022).

Selanjutnya, penelitian ketiga oleh M. Ali Hasymi, A. Faisol, dan F. Ariwibisono pada tahun 2021 mengenai “Sistem Informasi Geografis Pemetaan Warga Kurang Mampu Di Kelurahan Karang Besuki Menggunakan *Metode K-Means Clustering*.” Metode ini berhasil mengelompokkan warga kurang mampu dengan akurasi tinggi, memetakan distribusi kebutuhan bantuan secara efektif (Ali Hasymi dkk., 2021).

Penelitian selanjutnya membahas “Implementasi Sistem Informasi Geografis Menggunakan *Google Maps Api* Dalam Pemetaan Asal Mahasiswa” (Masykur, 2014). Penelitian ini memberikan gambaran geografis asal-muasal mahasiswa Universitas Muhammadiyah Ponorogo, menjadi dasar strategi pemasaran dan meningkatkan visibilitas kampus.

Penelitian selanjutnya membahas “Pengenalan *Google Firebase* Untuk *Hybrid Mobile Apps* Berbasis *Cordova*” (Ilhami, 2018), memperkenalkan teknologi aplikasi *hybrid Cordova* dan fitur *Google Firebase* dengan kelebihan pengembangan mudah namun juga mencatat kekurangan dalam performansi aplikasi.

Selanjutnya, “Implementasi Spasial *Algoritma Harvesine* pada *Mapbox API* untuk Pemetaan Pariwisata” oleh Adil dkk. (2023) berhasil menciptakan aplikasi menarik dan *user-friendly* dalam mencari lokasi wisata di Lombok.

“Penerapan *React JS* Pada Pengembangan *FrontEnd* Aplikasi *Startup Ubaform*” yang dilakukan oleh Nasution & Iswari (2021) membahas penggunaan *React JS* dalam pengembangan Front-end startup *Ubaform*, memberikan kemudahan dan efisiensi bagi pengembang dengan reusable komponen yang mempercepat proses *debugging*.

Penelitian selanjutnya, “Pengujian *Black Box* pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik *Equivalence Partitions*” yang mengevaluasi kualitas sistem aplikasi seleksi pemenang tender, menunjukkan bahwa metode *Black Box* berbasis *Equivalence Partitions* membantu dalam

pembuatan kasus pengujian, pengujian kualitas, dan identifikasi kesalahan dengan baik (Arwaz dkk., 2019).

Dalam sebuah penelitian yang dipublikasikan pada tahun 2022 dalam Jurnal Teknologi dan Informasi yang ditulis oleh Suhari, A. Faqih, dan FM. Basyar penelitian tersebut menggunakan metode Agile dalam pengembangan sistem informasi kepegawaian di CV. Angkasa Raya. Hasil penelitian mereka menunjukkan bahwa penerapan model Agile memang berhasil digunakan dalam mengembangkan sebuah sistem informasi kepegawaian yang dapat membantu meningkatkan efisiensi dalam berbagai proses pengelolaan data pegawai termasuk penggajian dan pengajuan cuti (Suhari dkk., 2022).

Penelitian selanjutnya, yang berjudul "Pengujian Beta Pada Aplikasi *Virtual Tour* Destinasi Wisata Di Kabupaten Batang (Study Kasus: Bandar Ecopark)", membahas tentang penggunaan closed beta testing dalam pengujian aplikasi virtual tour destinasi wisata di Kabupaten Batang, dengan studi kasus pada Bandar Ecopark. Hasil penelitian menunjukkan bahwa sebanyak 78,67% pengguna menyatakan setuju dengan aplikasi *virtual tour* yang dikembangkan. Ini mengindikasikan bahwa aplikasi tersebut memenuhi kebutuhan fungsional dan non-fungsional pengguna. Metode kuesioner dengan skala Likert digunakan dalam pengujian *closed beta*, yang memberikan gambaran yang jelas tentang persepsi pengguna terhadap aplikasi (Susanto dkk., 2020)

Penelitian kedua yang membahas *closed beta testing*, berjudul "Perancangan Sistem Informasi Manajemen Keuangan Menggunakan Metode Scrum (Studi Kasus: CV Kurnia Jaya)", membahas pengujian perangkat lunak

untuk sistem informasi manajemen keuangan. Hasil penelitian menunjukkan bahwa dari nilai rata-rata 83,9% untuk pengujian fungsional yang dilakukan oleh peneliti, sistem informasi manajemen keuangan sudah layak digunakan. Peneliti memilih *closed beta testing* karena pengguna aplikasi ini terbatas pada karyawan CV Kurnia Jaya, dan pengujian tidak memakan banyak waktu (Kusuma, 2021).

Dari kedua penelitian tersebut, dapat disimpulkan bahwa *closed beta testing* merupakan metode yang efektif untuk menguji kelayakan dan kegunaan aplikasi atau perangkat lunak sebelum diluncurkan secara luas. Dengan melibatkan pengguna potensial atau pihak terkait dalam tahap pengujian, pengembang dapat memperoleh umpan balik yang berharga untuk memperbaiki dan meningkatkan kualitas aplikasi sebelum peluncuran resmi.