

## **Sistem Informasi Geografis *Tour Guide* Pencarian Rute Terpendek Wisata Kabupaten Ketapang Menggunakan Algoritma Dijkstra**

**Anggi Purwanto<sup>1\*</sup>, Rachmad Wahid Saleh Insani<sup>2</sup>, dan Barry Ceasar Octariadi<sup>3</sup>**  
 Teknik Informatika, Universitas Muhammadiyah Pontianak, Pontianakan, Indonesia  
 \*e-mail *Corresponding Author*: 181220025@unmuhpkn.ac.id

### **Abstrak**

*The Tour Guide Geographic Information System was developed to overcome the difficulty of tourists in finding the shortest route to tourist attractions in Ketapang Regency. This application uses the Dijkstra algorithm in finding the fastest path between tourist sites. Although there are differences in route estimation and travel time with Google Maps, this application still provides accurate route estimates. The results of testing through black box testing and user acceptance testing showed success of 81.44%. This system applies exclusively in Ketapang Regency and meets the established standards.*

**Keywords:** *Geographic Information System; Tour Guide; Dijkstra algorithm; Shortest Route.*

### **Abstrak**

Sistem Informasi Geografis *Tour Guide* dikembangkan untuk mengatasi kesulitan wisatawan dalam menemukan rute terpendek ke objek wisata di Kabupaten Ketapang. Aplikasi ini menggunakan algoritma Dijkstra dalam pencarian jalur tercepat antar lokasi wisata. Meskipun terdapat perbedaan estimasi rute dan waktu tempuh dengan *Google Maps*, aplikasi ini tetap memberikan estimasi rute yang akurat. Hasil pengujian melalui *black box testing* dan *user acceptance testing* menunjukkan keberhasilan sebesar 81,44%. Sistem ini berlaku eksklusif di Kabupaten Ketapang dan memenuhi standar yang ditetapkan.

**Kata Kunci:** *Sistem Informasi Geografis; Tour Guide; Algoritma Dijkstra; Rute Terpendek.*

### **1. Pendahuluan**

Pentingnya memilih jalur terpendek dalam perjalanan merupakan faktor penting bagi setiap pelancong, karena hal ini dapat menghemat waktu, tenaga, dan sumber daya. Namun, menemukan rute terpendek dalam daerah dengan banyak alternatif jalur, seperti Kabupaten Ketapang, memerlukan perhatian yang serius. Terkadang, alternatif jalur diperlukan ketika terdapat hambatan pada jalur utama[1]. Penentuan lintasan terpendek secara efektif dapat dianggap sebagai pencarian nilai minimum dari seluruh kemungkinan lintasan yang tersedia. Namun, dalam praktiknya, mencari rute terpendek dalam perjalanan wisata seringkali rumit karena berbagai pilihan lintasan dan ketidakpastian jarak pada setiap lintasan alternatif[2].

Undang-Undang No. 10 Tahun 2009 tentang Kepariwisata mendefinisikan wisata sebagai perjalanan individu atau kelompok ke tujuan tertentu untuk rekreasi, pengembangan diri, atau mengeksplorasi daya tarik wisata[3]. Kabupaten Ketapang, sebagai contoh, memiliki beragam destinasi wisata yang tersebar di kecamatan yang berjauhan dari pusat kota. Oleh karena itu, penemuan rute terpendek menjadi sangat signifikan dalam perjalanan wisata agar memudahkan akses para pelancong ke tujuan mereka secara efisien dan berhemat biaya[4]. Namun, keterbatasan informasi tentang rute terpendek menuju objek wisata seringkali mengakibatkan kesulitan dalam memilih rute terbaik.

Penggunaan Sistem Informasi Geografis (SIG) telah terbukti sebagai pilihan yang sangat efektif dalam penentuan jalur terbaik menuju destinasi wisata di Kabupaten Ketapang[5]. SIG memungkinkan peneliti untuk melakukan analisis informasi spasial yang mendalam, menghasilkan visualisasi yang jelas, dan menyediakan data dalam bentuk peta digital yang sangat berguna. Elemen-elemen kunci dalam SIG, seperti perangkat lunak khusus, perangkat

keras yang handal, dan data pengguna aplikasi[6], semuanya berkontribusi untuk membantu peneliti mengatasi permasalahan ini dengan lebih efisien.

Dengan menerapkan algoritma Dijkstra pada SIG, para pelancong dapat dengan mudah menemukan jalur terpendek dan efisien ke destinasi wisata di Kabupaten Ketapang. Aplikasi ini tidak hanya menghemat waktu, biaya, dan usaha para pelancong, tetapi juga memberikan solusi praktis untuk navigasi destinasi wisata.

Pengembangan aplikasi "Sistem Informasi Geografis *Tour Guide* Pencarian Rute Terpendek Wisata Kabupaten Ketapang" menjadi solusi yang ideal bagi wisatawan dalam mendapatkan informasi dan jalur terpendek ke objek wisata. Selain itu, aplikasi ini juga mendukung Dinas Pariwisata dan Kebudayaan Ketapang dalam upaya mengelola dan meningkatkan jumlah kunjungan wisatawan di daerah tersebut.

## 2. Tinjauan Pustaka

Penelitian ini menggunakan Sistem Informasi Geografis *web* dan algoritma Dijkstra untuk membantu warga pendatang di Kota Malang menemukan SPBU dengan rute terpendek. Rekomendasi SPBU diberikan berdasarkan fitur-fitur SPBU. Penerapan *Bastpath* pada algoritma Dijkstra meningkatkan akurasi dalam peta *roadmap*[7].

Penelitian ini menggunakan sistem informasi *web* dengan *PgRouting* untuk mempromosikan pariwisata batik di Pekalongan. *PgRouting* menghitung rute terpendek dengan algoritma Dijkstra, hasilnya *WebGIS* dengan info sejarah dan destinasi batik. Validasi lapangan menunjukkan selisih rata-rata jarak 50m dan waktu 32 detik, dipengaruhi oleh pembulatan dan kondisi jalan. Kuisioner menunjukkan tingkat kepuasan responden sekitar 77,7% untuk efisiensi dan 74,5% untuk efektivitas aplikasi *WebGIS* ini[8].

Penelitian ini menggunakan Algoritma Dijkstra dalam aplikasi Android berbasis Google Maps untuk membantu pengguna menemukan rute terbaik ke tempat parkir. Algoritma ini memilih rute terpendek berdasarkan jarak yang dapat dilalui. Penggunaan Java dan Android Studio, serta model *prototyping* dan UML, membantu dalam pengembangan aplikasi ini. Hasil penelitian menunjukkan keberhasilan Algoritma Dijkstra dalam memberikan navigasi rute terbaik dari lokasi pengguna ke tempat parkir tujuan[9].

Penelitian ini mengembangkan aplikasi interaktif untuk memandu wisatawan di lokasi wisata dengan teknologi *augmented reality*. Aplikasi ini menggunakan metode kombinasi, permutasi, dan algoritma Dijkstra untuk memberikan saran urutan kunjungan dengan jalur terpendek. Smartphone Android digunakan dengan fitur seperti kamera, GPS, akselerometer, dan kompas. Hasil pengujian menunjukkan tingkat akurasi yang tinggi, dengan akurasi sudut sekitar 97,43% dan akurasi jarak sekitar 99,79%. Sistem ini efektif dalam memberikan panduan kepada wisatawan[10].

Penelitian ini menggunakan algoritma Dijkstra untuk merancang jalur evakuasi terpendek saat bencana gempa dan tsunami di Kawasan Boulevard, Kota Manado, Indonesia, yang rawan gempa akibat lempeng tektonik. Tujuannya adalah meningkatkan keselamatan warga dengan menemukan rute evakuasi tercepat. Hasilnya menunjukkan algoritma Dijkstra efektif membantu warga dalam situasi darurat[11].

Penelitian ini menerapkan algoritma Dijkstra untuk mencari rute ke museum di Kota Jakarta. Aplikasi Android yang dibuat menggunakan algoritma ini untuk mencari rute terpendek. Pengujian menunjukkan efektivitas algoritma Dijkstra, seperti hanya memerlukan 35% bobot total untuk mencapai Museum Nasional[12].

Penelitian ini mengembangkan aplikasi Android untuk pencarian pendonor darah berbasis lokasi di Bandar Lampung, dengan teknologi geolokasi dan algoritma Dijkstra. Menggunakan metode Extreme Programming dan UML, aplikasi ini memetakan lokasi pendonor dan memberikan rute terdekat. Pengujian menunjukkan *usability* tinggi dengan nilai *understandability* 88%, *learnability* 86.4%, *operability* 85.33%, serta tampilan yang menarik[13].

Penelitian ini fokus pada algoritma Dijkstra untuk menemukan rute terpendek ke objek wisata air di Kabupaten Klaten, destinasi wisata terkenal di Indonesia. Algoritma ini mendukung aksesibilitas bagi wisatawan lokal dan internasional dengan mencari jalur tercepat menuju objek wisata tersebut. Implementasi menggunakan program Tora dan hasilnya menunjukkan sepuluh jalur terpendek dari Terminal Delanggu[14].

Penelitian ini mengembangkan aplikasi Android menggunakan algoritma Dijkstra dan metode *Haversine* untuk pendakian Gunung Merapi via Selo. Aplikasi ini membantu pendaki mencari jalur terpendek dan mengatasi cuaca buruk. Gunakan *Google Maps* dan GPS untuk

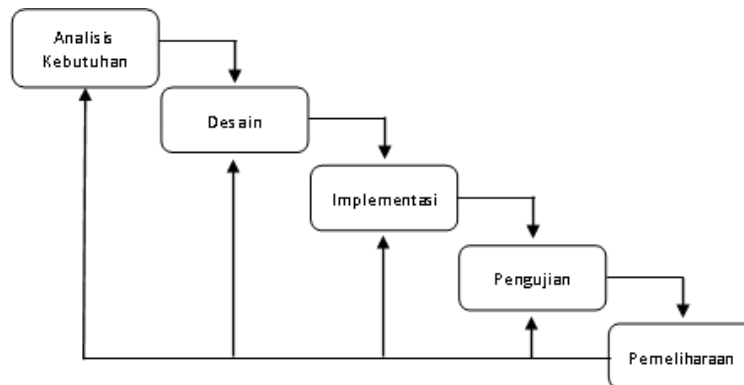
pemetaan. Hasilnya, aplikasi ini mudah digunakan di smartphone dan mendapat skor 77% dalam uji kelayakan[15].

Perbedaan penelitian ini dari penelitian terdahulu dengan fokusnya yang spesifik pada Kabupaten Ketapang sebagai lokasi penelitian. Sebagai perbandingan, penelitian-penelitian sebelumnya telah berfokus pada lokasi dan konteks yang berbeda, seperti Kota Malang, Pekalongan, Manado, Jakarta, dan sebagainya. Selain itu, penelitian ini memiliki tujuan yang unik yaitu memberikan estimasi rute yang akurat kepada wisatawan yang mengunjungi Kabupaten Ketapang, yang mungkin memiliki karakteristik geografis dan wisata yang berbeda dengan daerah-daerah yang telah menjadi subjek penelitian sebelumnya. Oleh karena itu, konsep yang ditawarkan dalam penelitian ini lebih relevan dan sesuai dengan kebutuhan wisatawan di Kabupaten Ketapang, yang mungkin memiliki tantangan dan preferensi yang unik dalam navigasi wisata mereka.

### 3. Metodologi

#### 3.1 Metode Pengembangan

Model Pengembangan Waterfall adalah pendekatan yang membagi proses pengembangan perangkat lunak menjadi beberapa tahapan yang dilaksanakan secara berurutan. Model ini memastikan bahwa setiap tahapan pengembangan dilakukan dengan baik dan sesuai rencana. Dalam konteks pengembangan "Sistem Informasi Geografis *Tour Guide* Pencarian Rute Terpendek Wisata Kabupaten Ketapang dengan Algoritma Dijkstra" model Waterfall diterapkan dengan tahapan-tahapan sebagai berikut:



Gambar 1. Model *Waterfall*

#### 3.2 Analisis Kebutuhan

Dalam penelitian ini, dilakukan analisis terperinci dan menyeluruh terhadap kebutuhan *user*. Adapun analisis kebutuhan fungsional *user* dapat dilihat pada Tabel 2. berikut:

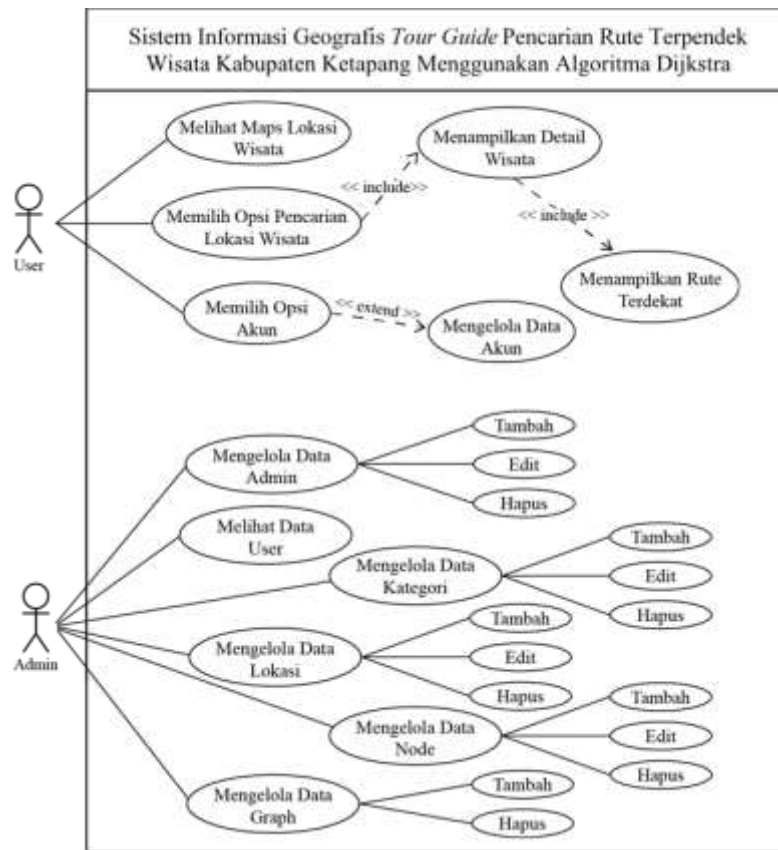
Tabel 1. Analisis Kebutuhan Fungsional *User*

<i>User</i>	Keterangan
Melihat peta persebaran wisata	<i>User</i> dapat melihat persebaran lokasi wisata di Kabupaten Ketapang.
Mencari lokasi wisata Kabupaten Ketapang	<i>User</i> dapat mencari lokasi wisata di Kabupaten Ketapang berdasarkan kategori wisata atau nama lokasi.
Melihat informasi wisata	<i>User</i> dapat melihat informasi detail mengenai lokasi wisata yang dipilih, seperti gambar slide, deskripsi dan tombol rute terdekat.
Melihat rute wisata	<i>User</i> dapat melihat rute perjalanan terpendek dari lokasi awal ke lokasi wisata yang dipilih.

#### 3.3 Desain Sistem

Tahap desain mencakup *activity diagram*, *class diagram*, *use case diagram*, arsitektur sistem, *database*, dan *wireframe*. Desain ini memastikan komponen sistem sesuai dengan kebutuhan dan spesifikasi sebelum implementasi.

### 3.3.1 Use Case Diagram

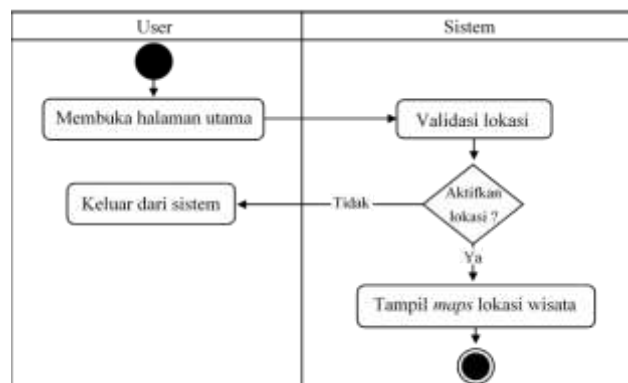


Gambar 2. Use Case Diagram

Pada gambar 2. *use case diagram*, sistem ini menggabungkan *backend* melalui *web server* dan *frontend* di *platform* Android. Terdapat dua jenis pengguna: *user* admin (mengelola data admin, user, kategori, wisata, *node*, dan *graph*) dan *user* reguler. Pengguna reguler bisa gunakan algoritma Dijkstra untuk mencari rute wisata, lihat detail wisata, tampilan rute terdekat, peta lokasi, dan edit info akun.

### 3.3.2 Activity Diagram

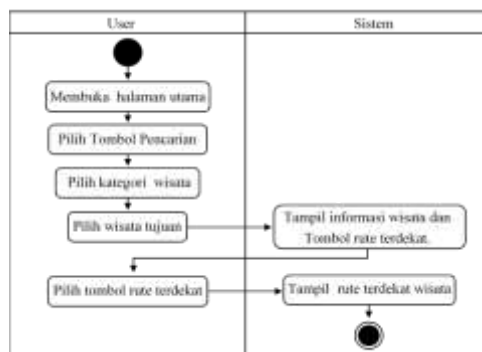
#### 3.3.2.1 Activity Diagram Melihat Maps Lokasi Wisata



Gambar 3. Activity Diagram Melihat Maps Lokasi Wisata

Pada gambar 3. *Activity diagram* melihat maps lokasi wisata, pengguna memasuki halaman utama dan kemudian sistem melakukan verifikasi lokasi. Jika lokasi diaktifkan, maka akan ditampilkan maps lokasi wisata. Namun, jika lokasi tidak diaktifkan, pengguna akan keluar dari aplikasi.

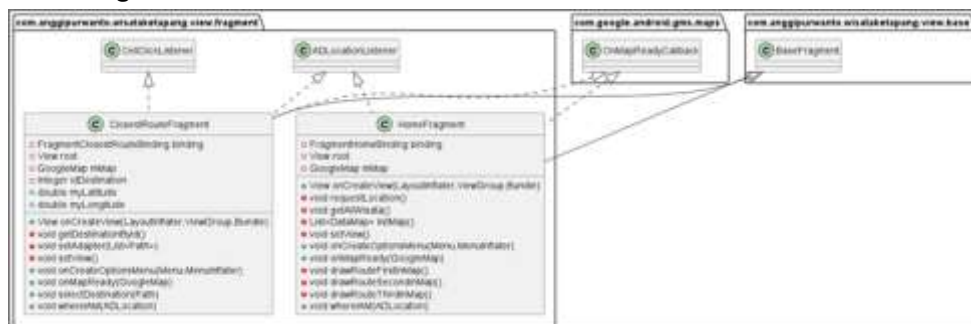
### 3.3.2.2 Activity Diagram Pencarian Lokasi Wisata



Gambar 4. Activity Diagram Pencarian Lokasi Wisata

Pada gambar 4. Activity diagram mencari lokasi wisata, pengguna memasuki halaman utama dan memilih tombol pencarian. Selanjutnya, pengguna memilih kategori wisata dan memilih wisata tujuan. Setelah itu, akan ditampilkan informasi wisata beserta tombol rute terdekat.

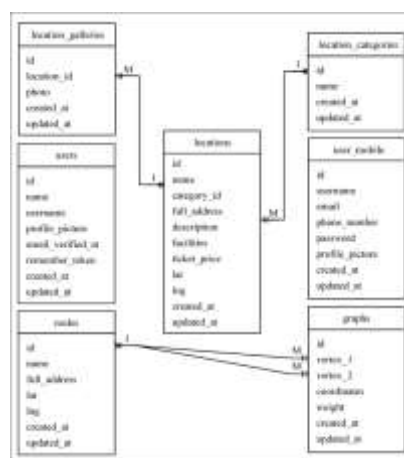
### 3.3.3 Class Diagram



Gambar 5. Class Diagram

Pada gambar 5. Class diagram, terdapat kotak untuk kelas atau objek, segitiga untuk pewarisan, dan bulat hijau untuk metode atau fungsi. Closest Route Fragment mengatur tampilan fragment, inialisasi peta google maps, serta mengambil dan menggambar rute terdekat. Homefragment adalah tampilan utama dengan peta google maps dan penanda lokasi wisata. Keduanya terhubung melalui base fragment dengan ad locationlistener dan onmapreadycallback.

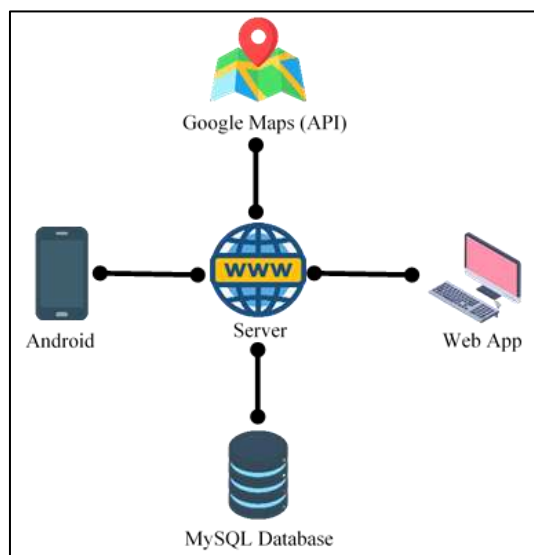
### 3.3.4 Database



Gambar 6. Skema Database

Tabel-tabel dalam skema database memiliki relasi beragam. Tabel *locations* terhubung dengan *location\_galleries* dalam hubungan *one-to-many*, serta dengan *location\_categories* dalam *many-to-one*. Tabel *graphs* dan *nodes* memiliki relasi *many-to-one* di antara keduanya. Tabel *users* dan *user\_mobiles* memiliki peran terpisah tanpa relasi langsung. Semua ini mencerminkan kompleksitas dan kebutuhan unik dalam struktur basis data sistem.

### 3.3.5 Arsitektur Sistem



Gambar 7. Arsitektur Sistem

Sistem ini menggunakan aplikasi seluler Android dengan data geospasial serta integrasi Google Maps API untuk peta dan lokasi. Manajemen data admin dilakukan melalui platform Web App, dengan informasi disimpan dalam basis data MySQL. Koneksi internet berfungsi sebagai penghubung antara klien dan server.

### 3.3.6 Penerapan Algoritma Dijkstra

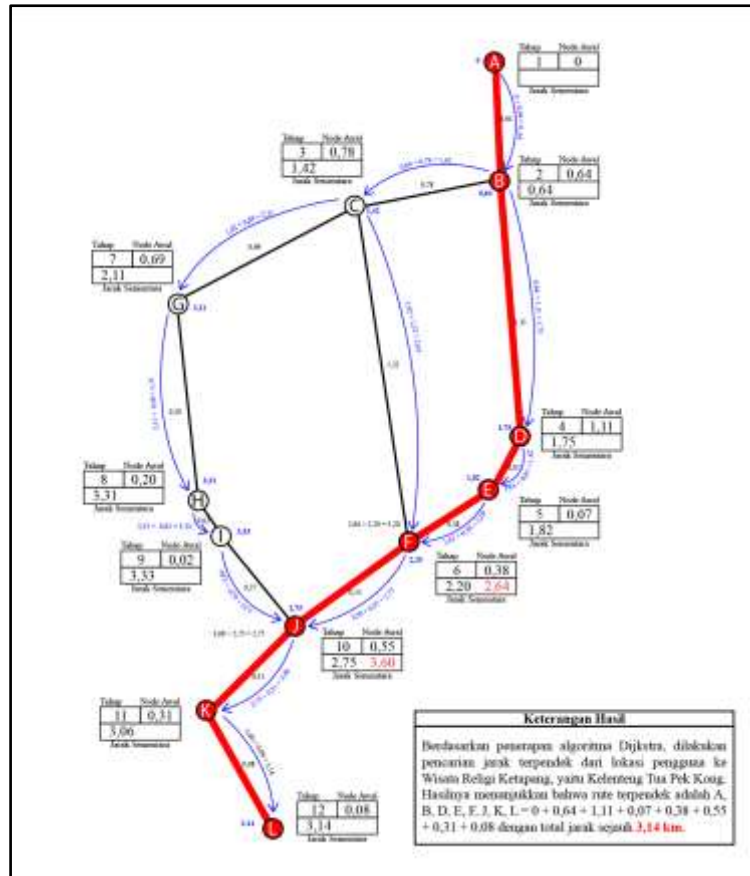
Pada sub-bab ini, dilakukan pemrograman sistem informasi geografis untuk mencari rute wisata terpendek. Algoritma Dijkstra digunakan dengan prinsip pemutusan yang terstruktur untuk mencari jalur terbaik pada setiap tahap. Pada tahap ke- $n$ , algoritma ini berhasil menemukan jalur terpendek dari *node* awal ke  $n$  *node* lainnya. Berikut adalah langkah-langkah algoritma Dijkstra:

- 1) Pada langkah pertama, tentukan *node* awal dari mana pencarian akan dimulai. Berikan bobot jarak pada *node* awal ke *node* terdekat satu per satu. Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain secara tahap demi tahap.
- 2) Setelah menentukan *node* awal, beri nilai bobot (jarak) untuk setiap titik ke titik lainnya dalam *graph*. Kemudian, atur nilai 0 pada *node* awal dan nilai tak terhingga pada *node* lain yang belum terisi.
- 3) Setel semua *node* yang belum dilalui sebagai *node* keberangkatan, dimulai dari *node* awal.
- 4) Dari *node* keberangkatan, pertimbangkan *node* tetangga yang belum dilalui dan hitung jaraknya dari titik keberangkatan. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya), hapus data lama dan simpan ulang data jarak dengan jarak yang baru.
- 5) Setelah kita mempertimbangkan setiap jarak terhadap *node* tetangga, tandai *node* yang telah dilalui sebagai *node* dilewati. *Node* yang telah dilewati tidak akan pernah diperiksa kembali, dan jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.

Set *node* belum dilewati dengan jarak terkecil (dari *node* keberangkatan) sebagai *node* keberangkatan" selanjutnya dan ulangi langkah 5 yaitu tandai *node* yang telah dilalui sebagai *node* dilewati[16].

### 3.4 Implementasi

#### 3.4.1 Implementasi Algoritma Dijkstra



Gambar 8. Contoh Implementasi algoritma Dijkstra

Berdasarkan penerapan algoritma Dijkstra, dilakukan pencarian jarak terpendek dari lokasi pengguna yaitu Jl. Brigjend. Katamso No.270, Sukaharja, Kec. Delta Pawan, Kabupaten Ketapang ke lokasi tujuan Wisata Religi Ketapang, yaitu Kelenteng Tua Pek Kong. Hasilnya menunjukkan bahwa rute terpendek adalah A, B, D, E, F, J, K, L = 0 + 0,64 + 1,11 + 0,07 + 0,38 + 0,55 + 0,31 + 0,08 dengan total jarak sejauh 3,14 km.

#### 3.4.2 Parameter dalam Penentuan Rute

Pada sub-bagian ini, akan disajikan penjelasan singkat mengenai parameter-parameter yang digunakan dalam proses penentuan rute dalam sistem. Berikut ini adalah uraian dari setiap tahap dalam pencarian rute terpendek berdasarkan gambar 8. Pada tahap awal, titik awal diberi label 0 sebagai jarak sementara. Selanjutnya, langkah-langkah berikut dilakukan untuk mencari rute terpendek:

Tabel 2. Detail Tahapan Algoritma Dijkstra

Tahap	Node Awal	Jarak Sementara	Rute Terdekat	Keterangan
1	0	-	A	Tahap awal, memulai perjalanan dari <i>node</i> A sebagai <i>node</i> awal.
2	0,64	0,64 km	A - B	Pada tahap ini, mencari rute terdekat dari A ke B. Menggunakan jalur langsung dari A ke B, mendapatkan jarak sementara sebesar 0,64 km.
3	0,78	1,42 km	A - B - C	Pada tahap ini, mencari rute terdekat dari A ke B (0,64 km) dengan jarak dari B ke C (0,78 km), mendapatkan jarak sementara sebesar 1,42 km.

Tahap	Node Awal	Jarak Sementara	Rute Terdekat	Keterangan
4	1,11	1,75 km	A - B - D	Pada tahap ini, mencari rute terdekat dari A ke B (0,64 km) dengan jarak dari B ke D (1,11 km), mendapatkan jarak sementara sebesar 1,75 km.
5	0,07	1,82 km	A - B - D - E	Pada tahap ini, mencari rute terdekat dari A ke B (0,64 km) dengan jarak dari B ke D (1,11 km) dan jarak dari D ke E (0,07 km), mendapatkan jarak sementara sebesar 1,82 km.
6	0,38	2,20 km	A - B - D - E - F	Pada tahap ini, mencari rute terdekat dari A ke F. Membandingkan dua rute yang berbeda: <ul style="list-style-type: none"> <li>Rute terdekat dari A ke F melalui B, D, dan E dengan jarak sementara 2,20 km.</li> <li>Rute alternatif dari A ke F melalui C dengan jarak sementara 2,64 km.</li> </ul> Karena jarak sementara untuk rute pertama lebih pendek ( $2,20 \text{ km} < 2,64 \text{ km}$ ), memilih rute A - B - D - E - F sebagai rute terdekat untuk mencapai node F dengan jarak sementara 2,20 km.
7	0,69	2,11 km	A - B - C - G	Pada tahap ini, mencari rute terdekat dari A ke B (0,64 km) dengan jarak dari B ke C (0,78 km) dan jarak dari C ke G (0,69 km), mendapatkan jarak sementara sebesar 2,11 km.
8	0,20	3,31 km	A - B - C - G - H	Pada tahap ini, mencari rute terdekat dari A ke B (0,64 km) dengan jarak dari B ke C (0,78 km), jarak dari C ke G (0,69 km), dan jarak dari G ke H (0,20 km), mendapatkan jarak sementara sebesar 3,31 km.
9	0,02	3,33 km	A - B - C - G - H - I	Pada tahap ini, mencari rute terdekat dari A ke B (0,64 km) dengan jarak dari B ke C (0,78 km), jarak dari C ke G (0,69 km), jarak dari G ke H (0,20 km), dan jarak dari H ke I (0,02 km), mendapatkan jarak sementara sebesar 3,33 km.
10	0,55	2,75 km	A - B - D - E - F - J	Pada tahap ini, mencari rute terdekat dari A ke J. Membandingkan dua rute yang berbeda: <ul style="list-style-type: none"> <li>Rute terdekat dari A ke J melalui B, D, E, dan F dengan jarak sementara 2,75 km.</li> <li>Rute alternatif dari A ke J melalui B, C, G, H dan I dengan jarak sementara 3,33 km.</li> </ul> Karena jarak sementara untuk rute pertama lebih pendek ( $2,75 \text{ km} < 3,33 \text{ km}$ ), memilih rute A - B - D - E - F - J sebagai rute terdekat untuk mencapai node J dengan jarak sementara 2,75 km.
11	0,31	3,06 km	A - B - D - E - F - J - K	Pada tahap ini, mencari rute terdekat dari A ke B (0,64 km) dengan jarak dari B ke D (1,11 km), jarak dari D ke E (0,07 km), jarak dari E ke F (0,38 km), jarak dari F ke J (0,55 km), dan jarak dari J ke K (0,31 km), mendapatkan jarak sementara sebesar 3,06 km.
12	0,08	3,14 km	A - B - D - E - F - J - K - L	Pada tahap ini, telah mencapai node L sebagai tujuan akhir. Menambahkan jarak dari A ke B (0,64 km) dengan jarak dari B ke D (1,11 km), jarak dari D ke E (0,07 km), jarak dari E ke F (0,38 km), jarak dari F ke J (0,55 km), jarak dari J ke K (0,31 km), dan jarak dari K ke L (0,08 km), mendapatkan jarak total sebesar 3,14 km.



Dengan demikian, setelah melalui semua tahap pencarian, ditemukan rute terpendek dari titik awal (A) ke titik tujuan (L) dengan total jarak 3,14 km.

**3.4.3 Implementasi Antar Muka**

**1) Halaman Login**

Halaman *login* meminta pengguna memasukkan nama pengguna dan kata sandi. Verifikasi dengan *database* menentukan akses ke halaman utama. Setelah sukses *login*, pengguna diarahkan ke halaman utama.

**2) Halaman Kategori**

Halaman kategori membantu pencarian destinasi wisata berdasarkan kategori. Pengguna memilih kategori, hasilnya ditampilkan. Halaman ini punya tombol *refresh* untuk hapus pilihan kategori, kembali ke halaman pencarian.



Gambar 9. Halaman Kategori

**3) Halaman Detail Wisata**

Halaman ini memberikan informasi tentang destinasi wisata melalui slide foto dan deskripsi singkat. Deskripsi mencakup lokasi, fasilitas, dan sejarah. Foto-foto memungkinkan pengguna melihat suasana tempat. Terdapat tombol rute terdekat untuk mengakses halaman rute terdekat yang menampilkan peta dan rute terpendek ke destinasi.



Gambar 10. Halaman Detail Wisata

**4) Halaman Rute Terdekat**

Halaman rute terdekat menampilkan rute terpendek dari lokasi pengguna ke destinasi wisata via algoritma Dijkstra. Peta menunjukkan rute dan lokasi yang dilalui. Informasi jarak dan estimasi waktu juga ditampilkan. Halaman ini membantu pengguna menentukan rute cepat ke destinasi wisata.



Gambar 11. Halaman Rute Terdekat

### 3.5 Pengujian

#### 3.5.1 Pengujian *Black Box*

Pengujian *Black box* dilakukan untuk memastikan bahwa sistem informasi geografis tour guide pencarian rute terpendek wisata Kabupaten Ketapang menggunakan algoritma Dijkstra berjalan dengan baik dan sesuai dengan spesifikasi yang telah ditentukan sebelumnya.

#### 3.5.2 Pengujian UAT (*User Acceptance Testing*)

Tahap ini adalah tahap pengujian terakhir sebelum sistem diimplementasikan. Pengguna akan menguji sistem dan melaporkan masalah atau bug yang ditemukan. Tujuannya adalah memastikan sistem memenuhi kebutuhan dan harapan pengguna sebelum diimplementasikan secara resmi. *User Acceptance Testing* (UAT) melibatkan pengguna sistem memberikan umpan balik tentang fitur dan kinerja sistem melalui kuesioner di *Google Form*.

### 3.6 Pemeliharaan

Setelah peluncuran, tahap pemeliharaan dimulai. Ini melibatkan dukungan, penambahan fitur baru, perbaikan bug, dan pembaruan sistem secara berkala untuk menjaga kinerja dan kebutuhan pengguna terpenuhi.

## 4. Hasil dan Pembahasan

### 4.1 Hasil Pengujian *Black Box*

Berikut ini adalah tabel hasil pengujian *black box* menggunakan teknik *Equivalence Partitioning*:

Tabel 3. Hasil Pengujian *Black Box Backend*

ID	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
A001	Login dengan "anggi" dan "12345678", tekan "Masuk"	Diterima dan masuk ke dashboard	Berhasil masuk ke dashboard	Valid
B001	Aktifkan fitur lokasi	Tampil peta dengan lokasi wisata terinput	Berhasil tampilkan peta dengan lokasi wisata	Valid
B002	Tidak aktifkan fitur lokasi	Tampil notifikasi dan pilihan aktifkan atau keluar	Berhasil tampilkan notifikasi dan pilihan	Valid
C001	Cari rute terdekat ke lokasi wisata	Tampil rute terdekat ke lokasi wisata	Berhasil tampilkan rute terdekat	Valid

### 4.2 Hasil Pengujian *User Acceptance Testing* (UAT)

Berikut ini adalah hasil UAT melalui kuisisioner *Google Form* yang mencerminkan pengalaman dan umpan balik pengguna terhadap sistem:

Tabel 4. Rekap Hasil Pengisian Kuesioner Pengguna

Pertanyaan	Jawaban Responden				
	Sangat Tidak Setuju (STS)	Tidak Setuju (TS)	Netral (N)	Setuju (S)	Sangat Setuju (SS)
P1	0	0	7	12	13
P2	0	1	5	16	10
P3	0	1	10	13	8
P4	0	2	10	8	12
P5	0	2	6	16	8
P6	0	0	5	18	9
P7	0	0	3	18	11
P8	0	0	6	16	10
P9	0	0	7	16	9
P10	0	1	5	15	11
JUMLAH	0	7	64	148	101

Bobot atau skor yang diberikan untuk pertanyaan yang diajukan kepada responden adalah sebagai berikut:

Tabel 5. Bobot pertanyaan

Bobot	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Tabel 6. Tingkat *Usability* Sistem

Persentase	Klasifikasi
Index 0% - 19,99 %	Sangat Lemah
Index 20% - 39,99 %	Lemah
Index 40% - 59,99 %	Cukup
Index 60% - 79,99 %	Kuat
Index 80% - 100 %	Sangat Kuat

Dari data yang diperoleh dan berdasarkan skor yang telah ditetapkan dapat dihitung seperti berikut :

Jumlah skor dari jawaban SS	=	101	x	5	=	505
Jumlah skor dari jawaban S	=	148	x	4	=	592
Jumlah skor dari jawaban N	=	64	x	3	=	192
Jumlah skor dari jawaban TS	=	7	x	2	=	14
Jumlah skor dari jawaban STS	=	0	x	1	=	0
Jumlah Skor Total					=	1303

Berdasarkan kategori jawaban tersebut, perhitungan persentase berdasarkan kategori adalah sebagai berikut:

Persentase kategori jawaban	= ( Jumlah skor / Jumlah skor total ) x 100%
Persentase menjawab SS	= ( 505 / 1303 ) x 100% = 38,76%
Persentase menjawab S	= ( 592 / 1303 ) x 100% = 45,43%
Persentase menjawab N	= ( 192 / 1303 ) x 100% = 14,74%
Persentase menjawab TS	= ( 14 / 1303 ) x 100% = 1,07%
Persentase menjawab STS	= ( 0 / 1303 ) x 100% = 0,00%

Berikut adalah hasil perhitungan dari responden sebanyak 32 orang dengan menggunakan skala bobot 1-5 untuk 10 pertanyaan, dengan nilai tertinggi (sangat setuju) memiliki bobot 5 dan nilai terendah (sangat tidak setuju) memiliki bobot 1:

Nilai tertinggi (sangat setuju)	= 32 x 10 x 5 = 1600
Nilai terendah (sangat tidak setuju)	= 32 x 10 x 1 = 320

Berdasarkan perhitungan yang menyatakan nilai tertinggi adalah 1600 dapat dicari persentase sebagai berikut:

Persentase total	= Jumlah skor total / Nilai tertinggi x100%
	= 1303 / 1600 x100%
	= 81,44%

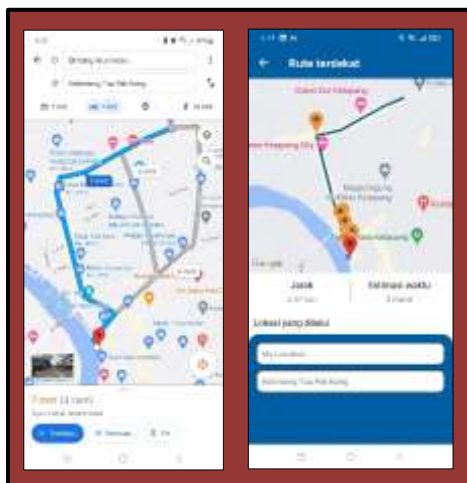
Berdasarkan tabel 6. tingkat *usability* sistem maka persentase 81,44%, dapat disimpulkan bahwa tingkat kegunaan sistem informasi geografis *tour guide* pencarian rute terpendek wisata Kabupaten Ketapang sangat tinggi dan diterima dengan baik oleh pengguna.

### 4.3 Hasil Uji Coba Perbandingan

Hasil pengujian bertujuan untuk membuktikan bahwa aplikasi yang telah dikembangkan sesuai dengan tujuan pengembangan dan mampu memenuhi kebutuhan pengguna.

#### 4.3.1 Perbandingan Kelenteng Tua Pek Kong

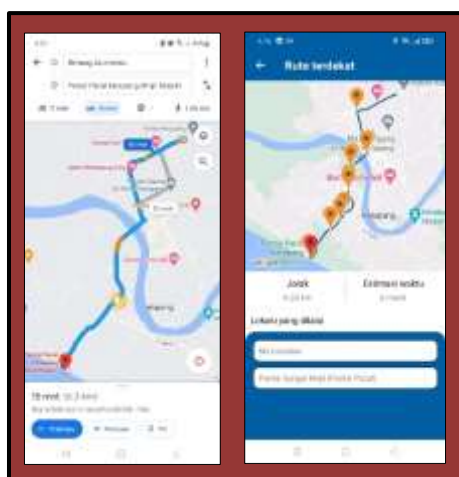
Perbandingan rute dari Bintang Alumindou ke Kelenteng Tua Pek Kong (Objek Wisata Religi Kabupaten Ketapang) menggunakan Google Maps dan algoritma Dijkstra. Rute dari Bintang Alumindou ke Kelenteng Tua Pek Kong memiliki perbedaan jarak sebesar 0,13 km.



Gambar 12. Perbandingan Kelenteng Tua Pek Kong

#### 4.3.2 Perbandingan Pantai Pecal Ketapang

Perbandingan rute dari Bintang Alumindou ke Pantai Pecal Ketapang (Objek Wisata Alam Kabupaten Ketapang) menggunakan Google Maps dan algoritma Dijkstra



Gambar 13. Perbandingan Pantai Pecal Ketapang

Rute dari Bintang Alumindou ke Pantai Pecal Ketapang memiliki perbedaan jarak sebesar 0,07 km.

Perbedaan jarak antara Google Maps dan aplikasi wisata Ketapang yang menggunakan algoritma Dijkstra mungkin disebabkan oleh beberapa faktor, tetapi aplikasi Ketapang masih dianggap akurat.

- 1) Rute yang Berbeda: Google Maps dan sistem ini menampilkan rute berbeda. Google Maps menekankan kecepatan, sementara Ketapang fokus pada jarak terpendek.
- 2) Data yang Digunakan: Google Maps memiliki data terperinci, termasuk lalu lintas dan cuaca, sedangkan sistem ini memiliki data yang lebih sederhana.
- 3) Implementasi Algoritma Dijkstra: Algoritma Dijkstra digunakan dalam sistem ini untuk mencari jalan terpendek, bukan yang tercepat. Ini membuatnya akurat dalam hal jarak, tetapi tidak selalu dalam waktu tempuh.

#### 5. Simpulan

Penelitian berhasil menciptakan sistem informasi geografis tour guide rute terpendek wisata di Kabupaten Ketapang. Hasil penelitian juga menunjukkan akurasi sistem aplikasi wisata Ketapang berbasis algoritma Dijkstra, dengan perbedaan jarak minimal 0,13 km dibandingkan Google Maps pada rute Bintang Alumindo ke Kelenteng Tua Pek Kong. Uji coba *black box* dan

*user acceptance testing* sukses, dengan persentase 81,44%. Saran penelitian selanjutnya melibatkan berbagai algoritma *routing* seperti Dijkstra,  $A^*$ , *Bellman-Ford*, dan algoritma *genetika* dibandingkan untuk menemukan metode paling efisien, serta formulasi matematis seperti jarak *Euclidean* dan *haversine* digunakan untuk mengestimasi jarak.

#### Daftar Referensi

- [1] S. Ardyan and A. Suyitno, "Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten Gunungkidul Dengan Program Visual Basic," *Unnes Journal of Mathematics*, vol. 6, no. 2, pp. 109–116, 2017.
- [2] M. K. Harahap and N. Khairina, "Pencarian Jalur Terpendek dengan Algoritma Dijkstra," *Jurnal & Penelitian Teknik Informatika*, vol. 2, no. 2, pp. 18–23, 2017.
- [3] D. M. Sari, "Partisipasi Masyarakat Dalam Mengembangkan Sarana Prasarana Kawasan Desa Wisata Borobudur," *Modul*, vol. 15, no. 2, pp. 133–140, 2015.
- [4] M. Mohamad, I. Ahmad, and Y. Fernando, "Pemetaan Potensi Pariwisata Di Kabupaten Waykanan Menggunakan Algoritma Dijkstra," *Jurnal Komputer Terapan*, vol. 3, no. 2, pp. 169–178, 2017.
- [5] H. A. Haridhi, *Buku Ajar Sistem Informasi Geografis Kelautan*. Banda Aceh: Syiah Kuala University Press, 2020.
- [6] G. C. B. K. Sugeng, "Aplikasi Pemandu Kurir Pengiriman Paket untuk Menentukan Tujuan Terdekat dengan Metoda Dijkstra Berbasis Android," *Techné Jurnal Ilmiah Elektroteknika*, vol. 22, no. 1, pp. 73–84, 2023.
- [7] M. Y. Rukmana and F. Ramdani, "Implementasi Algoritme Dijkstra pada Webgis untuk Pencarian Lokasi SPBU di Kota Malang," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 6, pp. 2141–2149, 2018.
- [8] R. Aminullah, A. Suprayogi, and A. Sukmono, "Aplikasi Pgrouting Untuk Penentuan Rute Alternatif Menuju Wisata Batik Di Kota Pekalongan Berbasis Webgis," *Jurnal Geodesi Undip*, vol. 7, no. 1, pp. 2337–845, 2018.
- [9] C. Prianto and M. Kusnadi, "Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada Mobile E-Parking Berbasis Sistem Informasi Geografis," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 3, no. 3, pp. 329–335, Oct. 2018.
- [10] A. N. Afyuddin, I. Arwani, and T. Afirianto, "Rancang Bangun Aplikasi Mobile Media Informasi Wisata Suroboyo Carnival Berbasis Augmented Reality," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 8, pp. 2307–2316, 2018.
- [11] A. G. Rumondor, S. R. Sentinuwo, and A. M. Sambul, "Perancangan Jalur Terpendek Evakuasi Bencana di Kawasan Boulevard Manado Menggunakan Algoritma Dijkstra," *Jurnal Teknik Informatika*, vol. 14, no. 2, pp. 261–268, 2019.
- [12] A. Cantona, F. Fauziah, and W. Winarsih, "Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta," *Jurnal Teknologi dan Manajemen Informatika*, vol. 6, no. 1, pp. 27–34, 2020.
- [13] S. Ahdan and Setiawansyah, "Pengembangan Sistem Informasi Geografis Untuk Pendor Darah dengan Algoritma Dijkstra berbasis Android," *Jurnal Sains dan Informatika*, vol. 6, no. 2, pp. 67–77, 2020.
- [14] N. A. Sudibyoy, P. E. Setyawan, and Y. P. S. R. Hidayat, "Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten Klaten," *Riemann Research of Mathematics and Mathematics Education*, vol. 2, no. 1, pp. 1–9, 2020.
- [15] P. Harsadi and D. Nugroho, "Implementasi Algoritma Dijkstra Dan Metode Haversine Pada Penentuan Jalur Terpendek Pendakian Gunung Merapi Jalur Selo Berbasis Android," *Jurnal Teknologi Informasi dan Komunikasi (TIKOMSiN)*, vol. 8, no. 1, pp. 61–67, Apr. 2020.
- [16] S. S. Putro, D. R. Anamisa, and F. A. Mufarroha, *Algoritma Pemrograman*. Malang: Media Nusa Creative (MNC Publishing), 2019.