

Perbandingan Algoritma *Dijkstra* dan *A-Star* dalam Pencarian Rute Terpendek Sekolah Menengah Atas (SMA) di Kota Palangka Raya Berbasis *Website*

Ferdha Alif Pratama¹⁾, Abertun Sagit Sahay²⁾, Nahumi Nugrahaningsih³⁾

^{1) 2) 3)} Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya
Kampus UPR Tunjung Nyaho Jl. Yos Sudarso, Palangka Raya

¹⁾ ferdhapratama021@gmail.com

²⁾ abertun@it.upr.ac.id

³⁾ nahumi@it.upr.ac.id

Abstrak

Pencarian rute terpendek adalah proses untuk menemukan jalur paling efisien antara dua titik dalam suatu graf. Masalah ini merupakan salah satu bentuk optimasi penting dalam teori graf, dengan tujuan meminimalkan total bobot lintasan, seperti jarak atau waktu tempuh. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pencarian rute terpendek menggunakan algoritma *Dijkstra* dan *A-star*, serta membandingkan kinerja kedua algoritma dalam konteks pencarian rute menuju Sekolah Menengah Atas (SMA) di Kota Palangka Raya. Berdasarkan data dari situs resmi dapo.kemdikbud.go.id, terdapat sekitar 28 Sekolah Menengah Atas (SMA) yang tersebar di seluruh wilayah kota ini.

Data dalam penelitian ini dikumpulkan melalui beberapa metode, termasuk observasi langsung, studi literatur, dan wawancara dengan masyarakat setempat. Pengembangan perangkat lunak mengikuti model air terjun (*waterfall*), yang mencakup tahapan analisis kebutuhan, desain sistem, implementasi, dan pengujian. Data koordinat sekolah diperoleh dari situs *Google Maps*.

Hasil dari percobaan 1 hingga percobaan 8 menunjukkan bahwa algoritma *A-star* secara umum menghasilkan rute yang lebih efisien, dengan rata-rata jarak 0,78 km lebih pendek dan waktu tempuh lebih cepat sekitar 4,71 menit dibandingkan algoritma *Dijkstra*. Namun, pada Percobaan 8, algoritma *Dijkstra* menunjukkan keunggulan dengan menghasilkan rute yang lebih pendek sebesar 0,17 km, waktu tempuh yang lebih cepat 0,21 menit, serta memeriksa 2 node lebih sedikit dibandingkan *A-star*. Penelitian ini merekomendasikan pemilihan algoritma yang sesuai dengan kondisi dan kebutuhan spesifik pengguna, serta mempertimbangkan penerapan *Vincenty's Formula* untuk meningkatkan akurasi dalam pencarian nilai heuristik.

Kata kunci: Rute Terpendek, Sekolah Menengah Atas, Algoritma *Dijkstra*, Algoritma *A-star*, *Haversine Formula*

Abstract

The shortest path search is the process of finding the most efficient route between two points in a graph. This problem is an important form of optimization in graph theory, aiming to minimize the total weight of the path, such as distance or travel time. This study aims to design and implement a shortest path search system using Dijkstra and A-Star algorithms, as well as to compare the performance of both algorithms in the context of finding routes to high schools (SMA) in Palangka Raya City. Based on data from the official site dapo.kemdikbud.go.id, there are approximately 28 high schools (SMA) distributed throughout the city.

Data in this research were collected through several methods, including direct observation, literature study, and interviews with local residents. The software development followed the waterfall model, which includes stages of requirement analysis, system design, implementation, and testing. The coordinate data of the schools were obtained from Google Maps.

Results from experiments 1 to 8 show that the A-Star algorithm generally produces more efficient routes, with an average distance of 0.78 km shorter and a travel time approximately 4.71 minutes faster compared to the Dijkstra algorithm. However, in Experiment 8, the Dijkstra algorithm demonstrated an advantage by producing a route that was 0.17 km shorter, with a travel time that was 0.21 minutes faster, and examining 2 fewer nodes compared to A-Star. This study recommends selecting the algorithm that suits the specific conditions and needs of users, as well as considering the application of Vincenty's Formula to improve accuracy in heuristic value

Keywords: Shortest Path, High School, Dijkstra Algorithm, A-Star Algorithm, Haversine Formula.

1. PENDAHULUAN

Algoritma merupakan sekumpulan instruksi atau aturan yang terdefinisi dengan baik untuk menyelesaikan masalah atau melakukan perhitungan secara sistematis. Dalam dunia komputasi, algoritma sering kali digunakan untuk memecahkan berbagai masalah optimasi, termasuk dalam pencarian rute terpendek (*Shortest Path Problem*). Pencarian rute terpendek telah diterapkan di berbagai bidang untuk mengoptimalkan kinerja sistem, baik dalam meminimalkan biaya maupun mempercepat suatu proses. Masalah pencarian rute terpendek merupakan salah satu persoalan dalam teori graf, yang bertujuan untuk meminimalkan bobot lintasan dalam suatu graf.

Menurut Mirza Ali Arsyad et al. (2019) Algoritma *Dijkstra* bekerja dengan mencari jarak terpendek dari simpul awal ke semua simpul lainnya dalam graf dengan menggunakan prinsip *greedy*, di mana pada setiap langkah dipilih sisi dengan bobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Sedangkan *A-star* menggabungkan keunggulan dari Algoritma *Dijkstra* dan *Greedy* dengan menggunakan jarak dari simpul awal ditambah estimasi ke tujuan untuk menentukan rute. *A-star* berakhir saat simpul tujuan ditemukan, memastikan rute optimal dengan menganalisis dan mengevaluasi beberapa jalur. Semakin kecil nilai $f(n)$ suatu *node*, semakin besar prioritasnya sebagai jalur terpendek.

Kombinasi dua algoritma dalam penyelesaian suatu masalah dapat dilakukan seperti pada penelitian Rosita et al. (2019) yang menggunakan dua gabungan algoritma *Dijkstra* dan *MCDM* dalam menyelesaikan permasalahan distribusi rute yang optimal untuk suatu kasus dan menghasilkan jaringan rute terpendek secara optimal. Selain itu, perbandingan kinerja antara algoritma *Dijkstra* dan *A-star* dalam pencarian rute terpendek telah dikaji oleh Yogaswara, D., & Suhartono, S. (2017) dalam konteks pencarian jalur evakuasi tsunami terpendek menuju shelter di Kabupaten Bantul berbasis *Android*. Penelitian ini berhasil mengembangkan Perbandingan Algoritma *A-star* dan *Dijkstra* pada Pencarian Jalur Evakuasi Tsunami Terpendek menuju Shelter di Kabupaten Bantul berbasis Aplikasi *Android*. Aplikasi dapat menampilkan rute dari posisi pengguna menuju shelter terdekat dengan algoritma *A-star* dan *Dijkstra*. Algoritma *A-star* menghasilkan rata-rata lama eksekusi 0.14 detik lebih cepat dan rata-rata jumlah *node* yang dicek 224 lebih sedikit dibandingkan Algoritma *Dijkstra*.

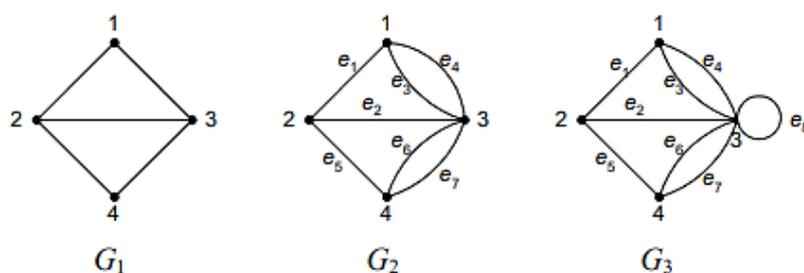
Berdasarkan uraian di atas, diajukan sebuah sistem berbasis *website* dengan judul **“Perbandingan Algoritma *Dijkstra* dan *A-star* dalam Pencarian Rute Terpendek Sekolah Menengah Atas (SMA) di Kota Palangka Raya Berbasis *Website*.”** Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pencarian rute terpendek yang menggunakan algoritma *Dijkstra* dan *A-star*, serta membandingkan kinerja kedua algoritma dalam konteks pencarian rute menuju Sekolah Menengah Atas (SMA) di Kota Palangka Raya. Kota Palangka Raya dipilih sebagai studi kasus karena kondisi geografis dan infrastruktur pendidikannya yang berkembang, dengan 28 SMA tersebar di wilayah ini. Penerapan sistem ini diharapkan memberikan solusi efisien dan akurat bagi pelajar dan orang tua dalam menentukan rute ke sekolah. Selain itu, penelitian ini bertujuan menunjukkan perbedaan kinerja kedua algoritma, sekaligus mendukung pengembangan teknologi informasi dan komunikasi di Kota Palangka Raya yang semakin penting seiring pertumbuhan kota tersebut.

2. TINJAUAN PUSTAKA

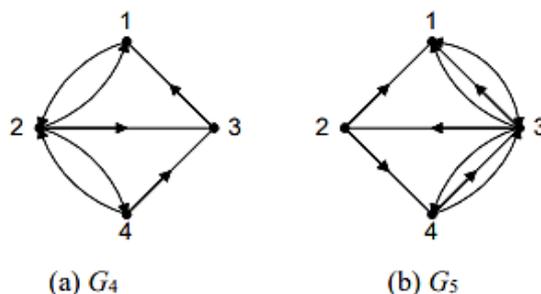
2.1 Teori Dasar Graf

Graf pada dasarnya terdiri dari dua komponen, yaitu titik/*node* (*vertex* atau V) dan busur (*edge* atau E). Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis, yaitu graf tak berarah dan graf berarah. Graf tak berarah adalah graf yang sisinya tidak mempunyai orientasi arah. Pada graf tak berarah, dua buah sisi (u, v) dan (v, u) adalah sisi yang sama, atau dengan kata lain $(u, v) = (v, u)$. Contoh graf tak berarah disajikan pada Gambar 2.3. Pada gambar tersebut terlihat bahwa G_1 , G_2 dan G_3 merupakan graf tak berarah karena setiap sisinya tidak mempunyai orientasi arah (Yogaswara, D., & Suhartono, S, 2017).

Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah. Pada graf berarah, dua buah sisi (u, v) dan (v, u) adalah sisi yang berbeda, dengan kata lain $(u, v) \neq (v, u)$. Pada penelitian ini, graf yang digunakan untuk merepresentasikan jaringan jalan yang terdapat di Kota Palangka Raya adalah graf berarah. Contoh graf berarah disajikan pada Gambar 2.4. Pada gambar tersebut terlihat bahwa G_4 dan G_5 merupakan graf berarah karena setiap sisinya mempunyai orientasi arah (R. Munir, 2012).



Gambar 2.1. Contoh Graf Tak-Berarah

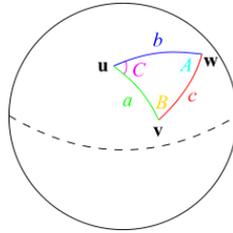


Gambar 2.2. Contoh Graf Berarah

2.2 Haversine Formula

Rumus *Haversine* merupakan persamaan penting pada sistem navigasi; nanti akan digunakan untuk menghitung jarak terpendek diantara dua tempat, seperti pada bola berdasarkan garis bujur dan garis lintang. Rumus ini pertama kali ditemukan oleh Jamez Andrew pada 1805 dan pertama kali digunakan oleh Josef de Mendoza y Ros pada 1801. Prof. James Inman menciptakan kata *Haversine* pada tahun 1835. Dalam studinya tentang “*Principal Problems of Maritime Astronomy*” (Proc. Royal Soc, 22 Des. 1796), Josef de Mendoza y Ros menggunakan *Haversine* untuk pertama kalinya. *Haversine* untuk menentukan jarak antar bintang. Rumus *Haversine* merupakan persamaan navigasi yang menghitung jarak melingkar diantara dua tempat di permukaan bola (bumi) menurut garis bujur dan garis lintang. Rumus *Haversine* adalah teknik untuk menghitung jarak antara dua tempat dengan asumsi bumi adalah permukaan datar dengan berbagai tingkat kelengkungan. Penerapan rumus ini mengandaikan pengabaian efek ellipsoidal, yang relatif akurat untuk memprediksi tingginya bukit dan dalamnya lembah di permukaan bumi (Rahayu, S., Fanni, R., & Bima, K, 2022).

Adapun Penelitian yang dilakukan oleh Rahayu, S., Fanni, R., & Bima, K. (2022) menunjukkan bahwa metode perhitungan jarak Euclidean memiliki nilai rata-rata akurasi sebesar 99,78%, sedangkan metode *Haversine* memiliki nilai rata-rata ketepatan sebesar 99,88% ketika dibandingkan dengan pengukuran jarak menggunakan *Google maps*. Kesimpulan dari penelitian tersebut adalah bahwa metode *Haversine* memiliki nilai rata-rata ketepatan yang lebih tinggi dibandingkan dengan metode *Euclidean Distance*.

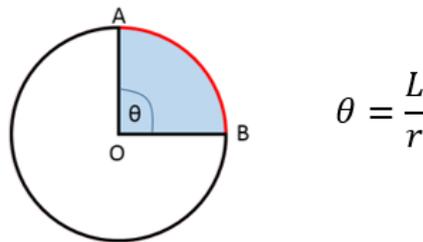


Gambar 2.3. Hukum *Haversine* dalam Segitiga Bola

Diberikan sebuah bola satuan, sebuah “segitiga” pada permukaan bola didefinisikan oleh lingkaran-lingkaran besar yang menghubungkan tiga titik u, v, dan w pada bola. Jika panjang ketiga sisi tersebut adalah a (dari u ke v), b (dari u ke w), dan c (dari v ke w), serta sudut yang berhadapan dengan sisi c adalah C, maka hukum *Haversine* menyatakan :

$$\text{hav}(c) = \text{hav}(a - b) + \sin(b) \text{hav}(c)$$

Dimana a,b,c merupakan jarak yang bersatuan radian/sudut karena berada dalam bidang bola, yang bisa kita korelasikan dengan persamaan busur di bawah ini :



Gambar 2.4. Rumus Busur Lingkaran

$$\text{haversine}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2}$$

Penggunaan rumus ini mengasumsikan pengabaian efek *ellipsoidal*, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi :

$$d = 2r \cdot \arcsin \sqrt{\sin^2\left(\frac{Lat_1 - Lat_2}{2}\right) + \cos(lat_1) \cos(lat_2) \sin^2\left(\frac{Long_1 - Long_2}{2}\right)}$$

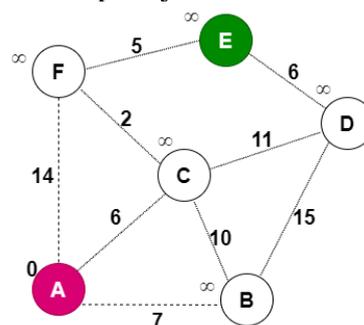
Keterangan :

- d : Jarak
- Lat_1 : Longitude Awal
- Lat_2 : Latitude Akhir
- $Long_1$: Longitude Awal
- $Long_2$: Latitude Akhir
- r : Radius Bumi = 6371 km
- 1° : 0.0174532925 radian

2.3 Algoritma Dijkstra

Algoritma *Dijkstra* adalah sebuah algoritma yang dikembangkan oleh seorang ilmuwan komputer dari Belanda, Edsger *Dijkstra*. Algoritma *Dijkstra* adalah sebuah algoritma yang menyelesaikan pencarian jalur terpendek pada graf dengan nilai non negatif untuk bobot setiap simpul, menghasilkan pohon jalur terpendek (Yogaswara, D., & Suhartono, S, 2017). Algoritma ini akan mencari jalur dengan *cost* minimum antara *node* tersebut dengan *node* lainnya. Algoritma *Dijkstra* memberikan *cost* untuk mencapai *node* $g(n)$ pada setiap *node* dan mengevaluasi *node-node* tersebut berdasarkan nilai $g(n)$ dari *node*. Semakin kecil nilai $g(n)$ dari *node* maka semakin besar prioritas *node* untuk dipilih menjadi jalur terpendek menuju *node* goal. Algoritma ini dapat digunakan untuk mencari total biaya(*cost*) dari lintasan terpendek yang dibentuk dari sebuah *node* ke sebuah *node* tujuan.

Algoritma *Dijkstra* bekerja dengan membuat jalur ke satu simpul optimal pada setiap langkah. Jadi pada langkah ke- n , setidaknya ada n *node* yang sudah diketahui jalur terpendek (Sunardi, Yudhana and Kadim, 2019). Algoritma *Dijkstra* mempunyai dasar mencari nilai *cost* yang terdekat dengan tujuan yang berguna dalam suatu graf berarah serta berbobot, sehingga bisa membantu memberi opsi jalur. Contohnya titik diumpamakan sebagai gedung serta garis diumpamakan jalan, maka Algoritma *Dijkstra* melakukan perhitungan bagi seluruh peluang bobot terkecil pada tiap titik. Pada teknik algoritma *Dijkstra*, titik maupun tanda lokasi dipakai sebab algoritma *Dijkstra* memakai graf langsung guna membentuk sebuah jarak terdekat Adi, N. H., Giatman, M., Simatupang, W., Afrina, A., & Watrianthos, R. (2021). Di bawah ini disajikan contoh graf melalui bobotnya guna menetapkan jalur memakai Algoritma *Dijkstra*.



Gambar 2.5. Contoh Graf Algoritma *Dijkstra*

2.4 Algoritma A-Star

Algoritma *A-STAR* (*A-star*) adalah salah satu algoritma pencarian jalur yang paling populer dan efisien dalam ilmu komputer, terutama digunakan dalam pemrograman game dan sistem navigasi. Algoritma ini menggabungkan fitur dari algoritma *Dijkstra* dan algoritma pencarian *Best-First Search* untuk menemukan jalur terpendek dari titik awal ke titik tujuan. Menurut Riki Idayat & Ita Hidayani (2022) Algoritma *A-star* (*A-STAR*) merupakan Algoritma *pathfinding* pengembangan dari Algoritma *BFS* (*Breadth First Search*). Seperti hanya pada *BFS*, untuk menemukan solusi, *A-STAR* juga ‘dituntun’ oleh fungsi *heuristik*. Perbedaan cara kerja *A-STAR* dengan *BFS* adalah selain memperhitungkan *cost* dari *currentstate* ke tujuan dengan fungsi *heuristik* (seperti *BFS*), Algoritma ini juga mempertimbangkan *cost* yang telah ditempuh selama ini dari *initial state* (posisi yang dituju) menuju ke *current state* (posisi awal). Jadi, apabila jalan yang ditempuh sudah terlalu panjang dan ada jalan lain yang lebih kecil *cost*-nya namun memberikan solusi yang sama apabila dilihat dari goal, maka jalan yang baru yang lebih pendek itulah yang akan dipilih.

Sedangkan menurut Mirza Ali Arsyad et al (2019) menyatakan bahwa : Algoritma *A-STAR* merupakan algoritma yang digunakan untuk mencari lintasan terpendek dengan menggunakan biaya yang paling rendah. Algoritma *A-STAR* adalah algoritma gabungan antara algoritma pencarian *Uniform Cost* dan *Greedy-Best First*. Implementasi dari algoritma *A-star* yaitu dapat memberikan solusi yang terbaik dengan waktu yang optimal. Karakteristik yang menjelaskan algoritma *A-STAR* adalah pengembangan dari “daftar tertutup” untuk merekam area yang

dievaluasi. Daftar tertutup ini adalah sebuah daftar untuk merekam area berdekatan yang sudah dievaluasi kemudian melakukan perhitungan jarak yang dikujungi dari “titik awal” dengan jarak diperkirakan ke “titik tujuan”. Algoritma *A-STAR* mengevaluasi terhadap *node-n* dengan menggabungkan $g(n)$, adalah biaya yang dikeluarkan untuk mencapai *node*, dan $h(n)$ adalah biaya yang diperlukan untuk mencapai *node*, ditunjukkan dalam persamaan matematika yaitu sebagai berikut :

$$f(n) = g(n) + h(n)$$

Keterangan :

$f(n)$ = solusi biaya estimasi termurah *node n* untuk mencapai tujuan

$g(n)$ = biaya yang sudah dikeluarkan dari titik awal sampai keadaan *n*

$h(n)$ = estimasi biaya untuk sampai pada titik tujuan dimulai dari keadaan *n*

Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah simpul awal (*starting point*) menuju simpul tujuan dengan memperhatikan harga (*F*) terkecil. Diawali dengan menempatkan *A* pada *starting point*, kemudian memasukkan seluruh simpul yang bertetangga dan tidak memiliki atribut rintangan dengan *A* ke dalam *open list*. Kemudian mencari nilai *H* terkecil dari simpul-simpul dalam *open list* tersebut. Kemudian memindahkan *A* ke simpul yang memiliki nilai *H* terkecil. Simpul sebelum *A* disimpan sebagai *parent* dari *A* dan dimasukkan ke dalam *closed list*. Jika terdapat simpul lain yang bertetangga dengan *A* (yang sudah berpindah) namun belum termasuk ke dalam anggota *open list*, maka masukkan simpul-simpul tersebut ke dalam *open list*. Setelah itu, bandingkan nilai *G* yang ada dengan nilai *G* sebelumnya (pada langkah awal, tidak perlu dilakukan perbandingan nilai *G*). Jika nilai *G* sebelumnya lebih kecil maka *A* kembali ke posisi awal. Simpul yang pernah dicoba dimasukkan ke dalam *closed list*. Hal tersebut dilakukan berulang-ulang hingga terdapat solusi atau tidak ada lagi simpul lain yang berada pada *open list*.

2.5 Sistem Pencarian Rute Terpendek Berbasis Website

Jalur terpendek (*shortest path*) adalah jalur optimum yang dapat diselesaikan dengan menggunakan *graph*. Jalur ini biasanya ditentukan oleh rute yang memiliki total biaya perjalanan yang paling kecil atau murah. Jika diaplikasikan dengan *graph* maka setiap garis pada simpul titik memiliki bobot berupa nilai dan apabila dijumlahkan bobot dari garis yang dilalui maka memiliki nilai yang minimal. Berikut adalah algoritma-algoritma untuk menyelesaikan masalah jalur (Ramadhan, Zuhri, Muhammad Zarlis, Syahril Efendi, Andysah Putera, dan Utama Siahaan, 2018.). Adapun beberapa jenis persoalan yang dihadapi dalam pencarian lintasan terpendek adalah sebagai berikut :

1. Lintasan terpendek antara dua buah simpul tertentu (*a pair shortest path*).
2. Lintasan terpendek antara semua pasangan simpul (*all pairs shortest path*).
3. Lintasan terpendek dari simpul tertentu ke semua simpul yang lain (*single source shortest path*).
4. Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (*intermediate shortest path*).

Pencarian rute terpendek adalah proses menemukan jalur paling efisien antara dua titik dalam suatu graf, yang biasanya direpresentasikan sebagai peta jalan.

3. METODE PENELITIAN

3.1 Objek Penelitian

Penelitian ini dilaksanakan di Kota Palangka Raya. Alasan pemilihan lokasi ini adalah karena Kota Palangka Raya belum memiliki sistem yang memfasilitasi pencarian rute terpendek dengan penerapan algoritma *Dijkstra* dan *A-star*. Penelitian ini juga bertujuan untuk membandingkan kedua algoritma tersebut dalam menemukan rute optimal di lingkungan sekolah menengah atas (SMA) di kota ini.

3.2 Metode Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan informasi yang dibutuhkan melalui observasi dan pencatatan data (Seprianto, 2022). Metode yang digunakan dalam penelitian ini adalah :

1. Metode Pengamatan (*Observasi*)

Observasi diterapkan dengan melakukan peninjauan langsung ke lokasi penelitian guna memperoleh data atau gambar serta keterangan terhadap sistem.

2. Studi *Literature*

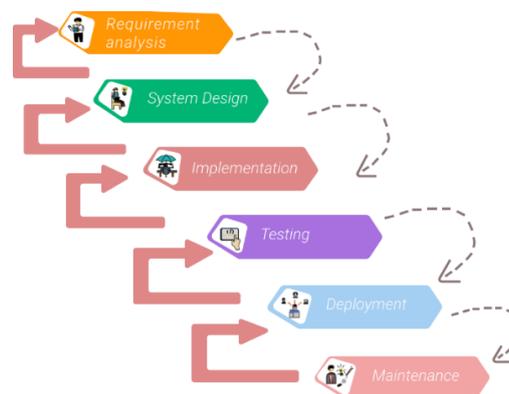
Studi literatur dilakukan dengan mengumpulkan informasi dari buku, artikel, dan sumber online terkait metode pengembangan web serta penerapan algoritma *Dijkstra* dan *A-star* dalam pencarian rute terpendek. Dari referensi tersebut, diambil kesimpulan mengenai perancangan sistem, teknik pelaksanaan, dan metode yang akan digunakan dalam tugas akhir ini.

3. Metode Wawancara (*Interview*)

Tahapan ini merupakan pengumpulan data dengan melakukan wawancara serta melakukan tanya jawab secara langsung dengan pihak yang bersangkutan yaitu Masyarakat Kota Palangka Raya baik pelajar ataupun orang tua siswa.

3.3 Metode Pengembangan Perangkat Lunak

Model air terjun (*waterfall*) mendorong tim pengembang untuk merinci kebutuhan sistem dan menentukan apa yang harus dilakukan oleh perangkat lunak (Janner S., 2010). Tahapan-tahapan penelitian dalam studi ini disusun berdasarkan kerangka kerja seperti yang ditunjukkan pada Gambar 3.1 di bawah ini :



Gambar 3.1. Waterfall Model (Ian Sommerville, 2011)

Metode penelitian merupakan bagian dari naskah hasil penelitian yang menjelaskan tentang langkah-langkah yang dilakukan dalam penelitian, alasan pemilihan sampel, proses validasi, dan pengukuran yang dilakukan. Adapun tahapan dari metode *waterfall* adalah sebagai berikut :

1. Analisis kebutuhan dan definisi

Pengembang mengidentifikasi masalah pada sistem lama melalui wawancara dan observasi, kemudian menentukan kebutuhan serta definisi sistem baru, termasuk proses bisnis dan fitur yang akan dibuat.

2. Sistem dan desain perangkat lunak

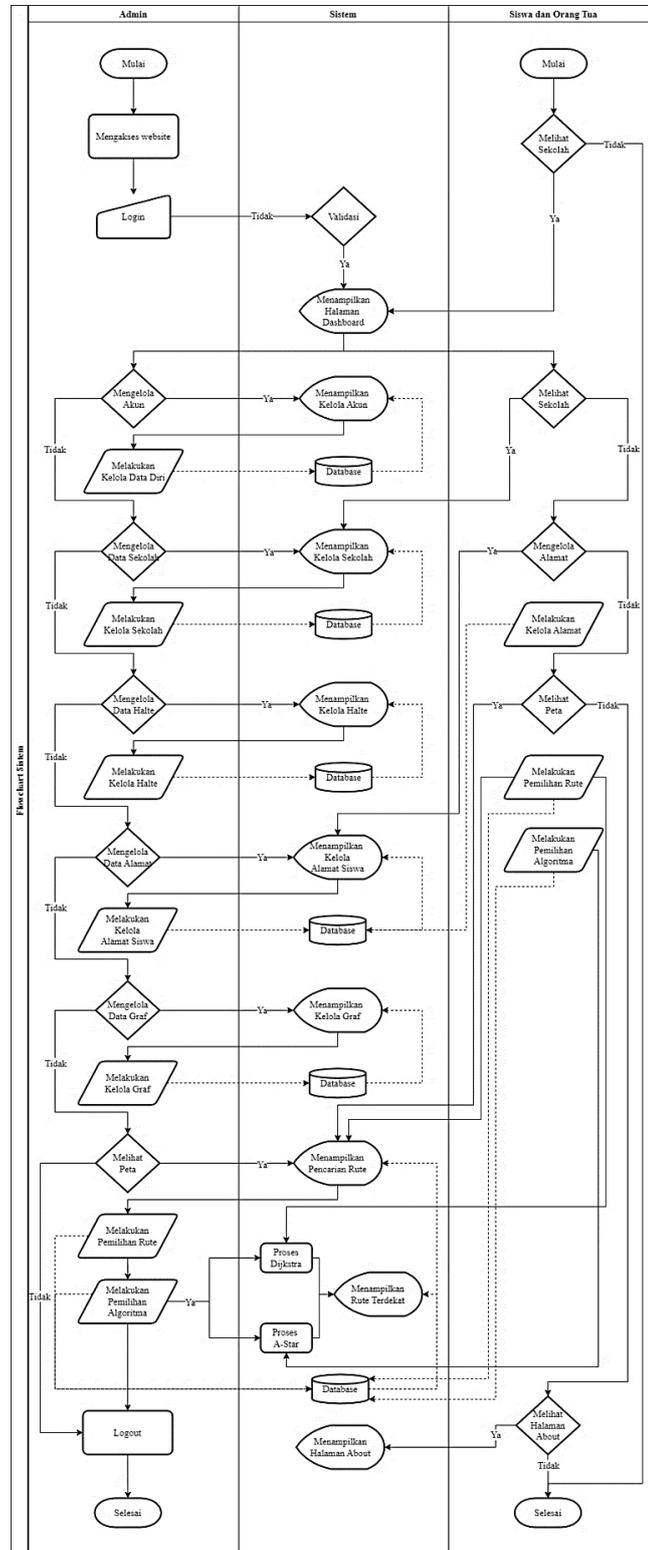
Informasi yang diperoleh digunakan untuk membuat flowchart, pemodelan sistem dengan UML, dan prototipe UI dari sistem yang akan dibuat.

3. Implementasi dan pengujian sistem

Sistem diimplementasikan dalam bentuk program sesuai hasil desain, dan diuji menggunakan metode blackbox testing.

3.4 System and Software Design
 1. Flowchart

Pada sistem yang dibangun, terdapat dua jenis pengguna, yaitu Admin dan siswa atau orang tua. Berikut adalah flowchart dari sistem yang akan dibangun :

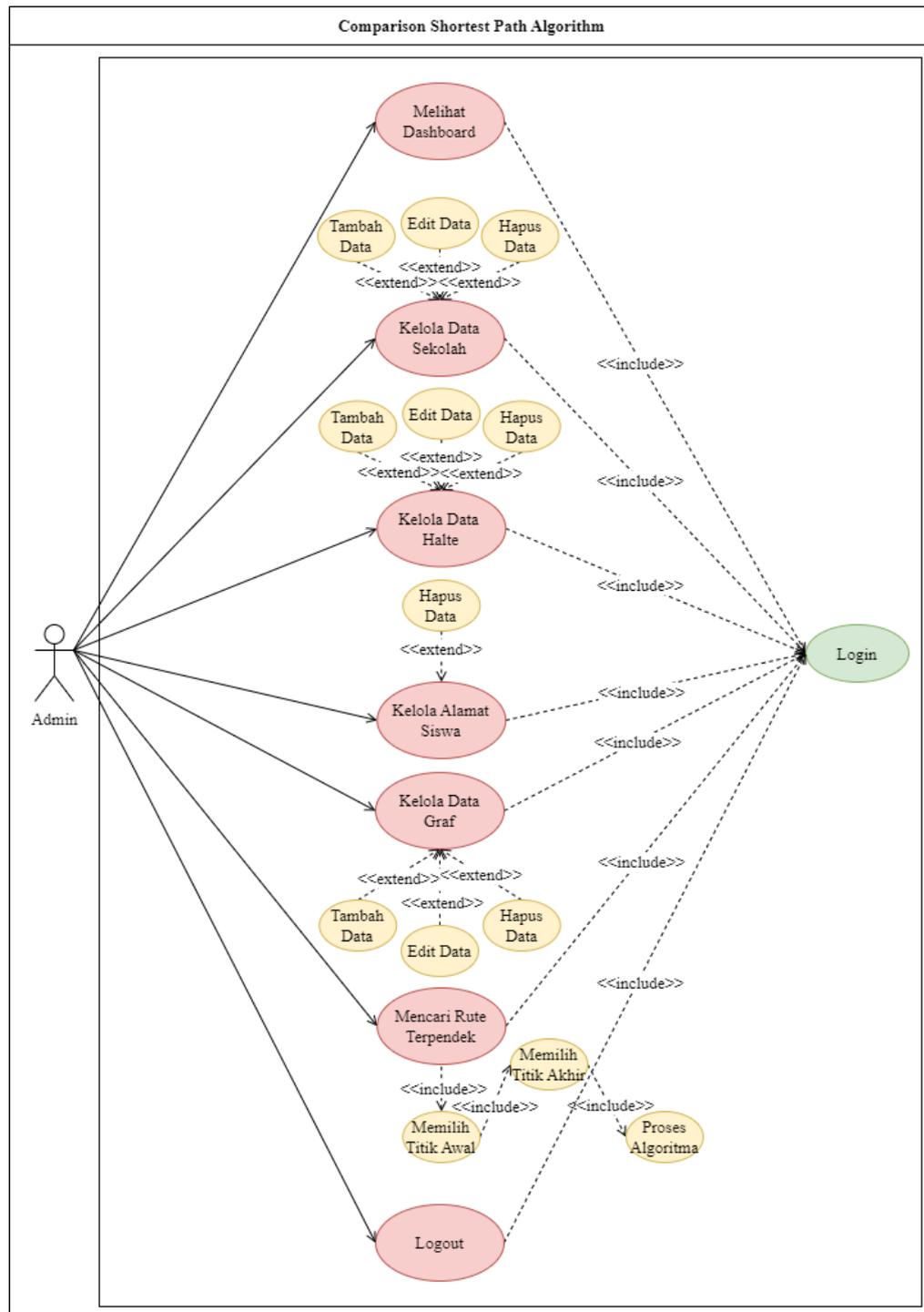


Gambar 3.1. Flowchart Sistem

2. Use Case Diagram

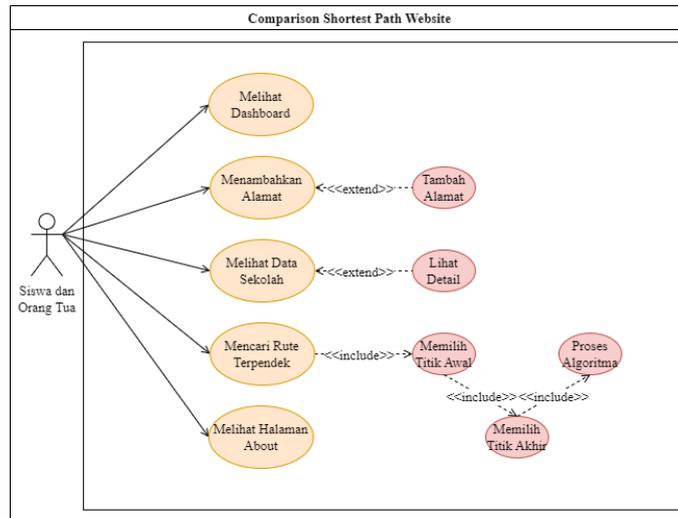
Diagram *use case* dapat digunakan pada fase analisis untuk membantu dalam mengembangkan persyaratan sistem dan memahami operasi yang diinginkan dari sistem tersebut. Pada *use case* diagram terdapat definisi aktor, definisi *use case* serta *use case* diagram. *Use case* diagram ini menggambarkan sistem dari sudut pandang aktor (*user*) berdasarkan fungsionalitas sistem.

1. Use Case Admin



Gambar 3.2. Use Case Admin

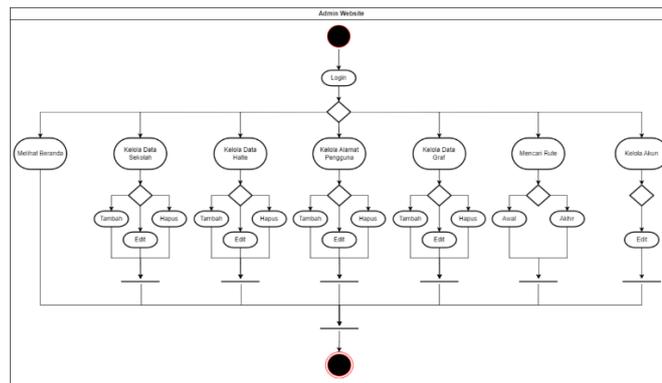
2. Use Case Siswa dan Orang Tua



Gambar 3.3. Use Case Siswa dan Orang Tua

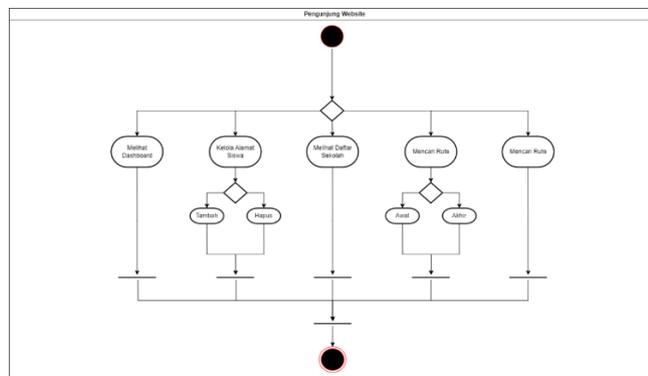
3. Activity Diagram

1. Activity Diagram Admin



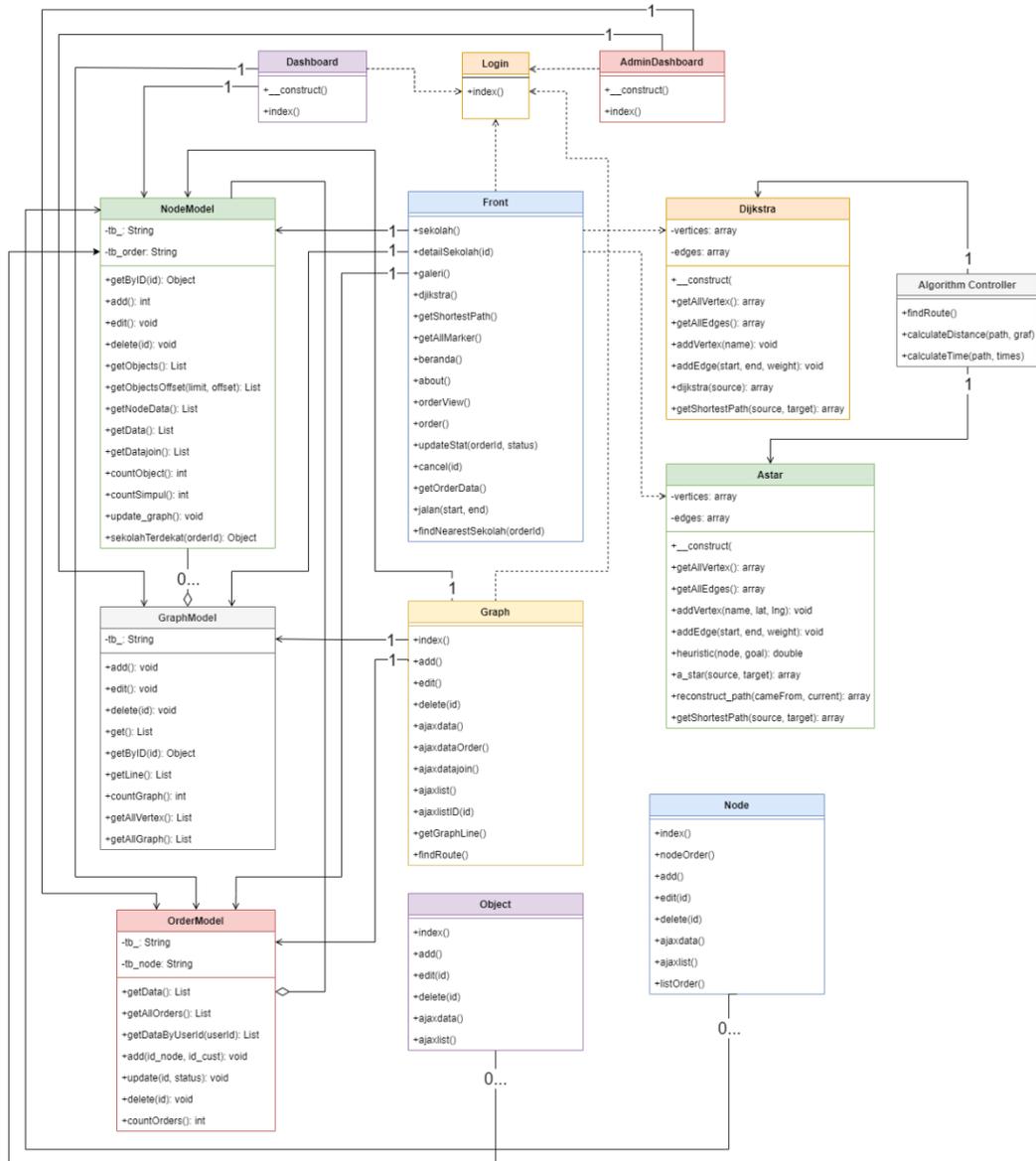
Gambar 3.4. Activity Diagram Admin

2. Activity Diagram Siswa dan Orang Tua



Gambar 3.5. Activity Diagram Siswa dan Orang Tua

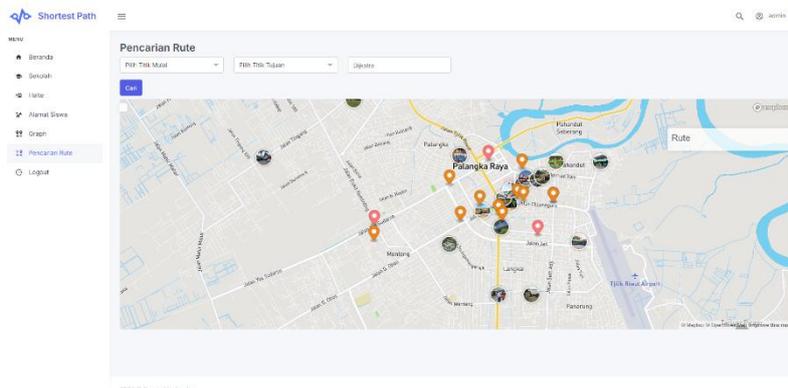
4. Class Diagram



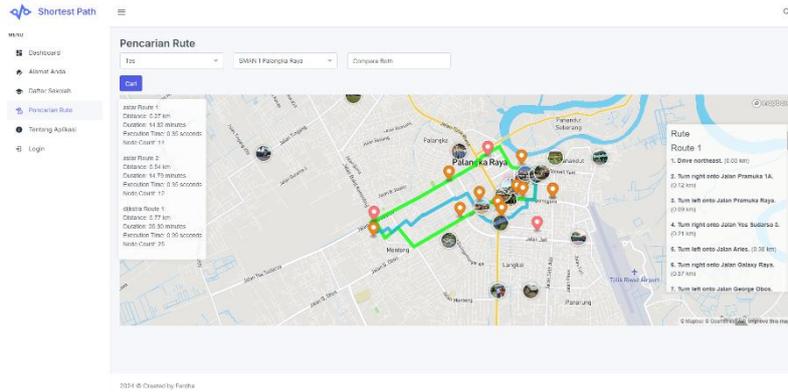
Gambar 3.6. Class Diagram

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem



Gambar 4.1. Implementasi Sistem Admin

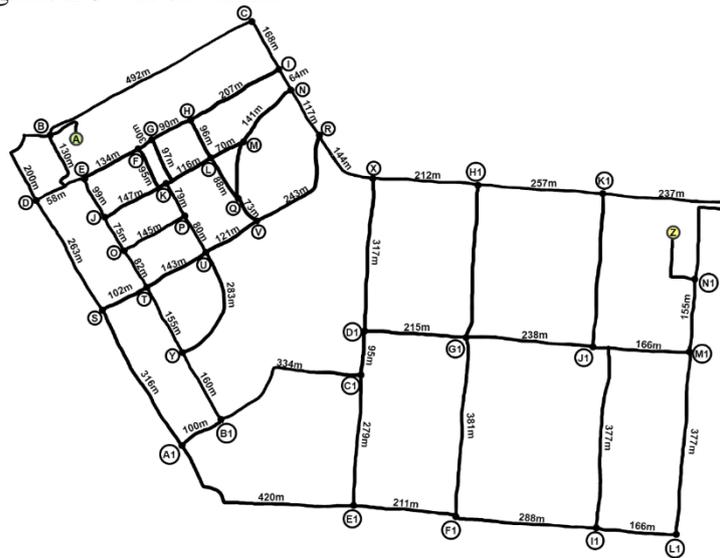


Gambar 4.2. Implementasi Halaman Pencarian Rute

4.2 Implementasi Algoritma

1. Pengujian Algoritma Dijkstra

Dalam penelitian ini, terdapat 28 Sekolah Menengah Atas di Kota Palangka Raya yang terdata dalam sistem, serta 18 halte yang tersebar di seluruh Kota Palangka Raya. Penelitian ini menggunakan satu contoh kasus graf untuk mencari rute terpendek dari Dinas Pendidikan Provinsi Kalimantan Tengah ke SMAN 2 Palangka Raya. Berikut adalah contoh graf sederhana yang dibuat dengan menetapkan sejumlah jalur yang dapat dilalui dari Dinas Pendidikan Provinsi Kalimantan Tengah ke SMAN 2 Palangka Raya, dengan bobot tertentu pada setiap rute yang ditunjukkan pada gambar 3.4 di bawah ini :



Gambar 4.3. Contoh Graf dan Node

Setelah memperoleh graf *node*, langkah berikutnya adalah melakukan inisialisasi untuk setiap *node* yang tersedia, seperti yang ditunjukkan pada Tabel 4.1 berikut :

Tabel 4.1. Keterangan Node

No	Node	Keterangan Node
1	A	Dinas Pendidikan Kalimantan Tengah
2	B	Jl. DI. Panjaitan
3	C	Simpang 4 Jl. S. Parman
4	D	Simpang 4 Jl. Jend. Sudirman
5	E	Simpang 3 Jl. Jend. Urip Sumoharjo
6	F	Simpang 3 Jl. Sutanegara I
7	G	Simpang 3 Jl. Sutanegara II

8	H	Simpang 3 Jl. Sutanegara III
9	I	Simpang 4 Jl. S. Parman dengan Jl. Jend Sudirman
10	J	Simpang 3 Jl. Gajah Mada
11	K	Simpang 4 Jl. Sutanegara I,II dengan Jl. Gajah Mada
12	L	Simpang 4 Jl. Ir. Juanda
13	M	Simpang 3 Jl. Gajah Mada dengan Jl. Ir. Juanda
14	N	Simpang 4 Jl. Ahmad Yani
15	O	Simpang 3 Jl. Hausman Baboe
16	P	Simpang 3 Jl. Hausman Baboe dengan Jl. Sutanegara
17	Q	Jl. Ir. Juanda
18	R	Simpang 4 Jl. Ahmad Yani dengan Jl. Let. Jend. Suprpto
19	S	Simpang 3 Jl. Imam Bonjol dengan Jl. Let. Jend. Suprpto
20	T	Simpang 4 Jl. Let. Jend. Suprpto dan Jl. Jend. Urip Sumoharjo
21	U	Simpang 4 Jl. Dr. Soetomo
22	V	Simpang 4 Jl. Let. Jend. Suprpto dengan Jl. Ir. Juanda
23	X	Simpang 4 Jl. Ahmad Yani dengan Jl. Tambun Bungai
24	Y	Simpang 3 Jl. Dr. Soetomo dengan Jl. Jend. Urip Sumoharjo
25	A1	Simpang 3 Jl. Imam Bonjol dengan Jl. Patih Rumbih
26	B1	Simpang 3 Jl. Patih Rumbih dengan Jl. Jend. Urip Sumoharjo
27	C1	Simpang 3 Jl. Tambun Bungai dengan Jl. Patih Rumbih
28	D1	Simpang 3 Jl. Tambun Bungai dengan Jl. R.A. Kartini
29	E1	Simpang 3 Jl. Diponegoro dengan Jl. Tambun Bungai
30	F1	Simpang 3 Jl. Diponegoro dengan Jl. W. Sudirohusodo
31	G1	Simpang 3 Jl. W. Sudirohusodo dengan Jl. R. A. Kartini
32	H1	Simpang 3 Jl. W. Sudirohusodo dengan Jl. Ahmad Yani
33	I1	Simpang 3 Jl. Diponegoro dengan Jl. AIS. Nasution
34	J1	Simpang 3 Jl. R. A. Kartini dengan Jl. AIS. Nasution
35	K1	Simpang 3 Jl. AIS. Nasution dengan Jl. Ahmad Yani
36	L1	Simpang 3 Jl. Diponegoro dengan Jl. K. S. Tubun
37	M1	Simpang 4 Jl. R. A. Kartini
38	N1	Jl. K. S. Tubun
39	Z	SMAN 2 Palangka Raya

Berdasarkan graf rute diatas, diperoleh sejumlah jalur dari Dinas Pendidikan Provinsi Kalimantan Tengah menuju SMAN 2 Palangka Raya sebagai berikut :

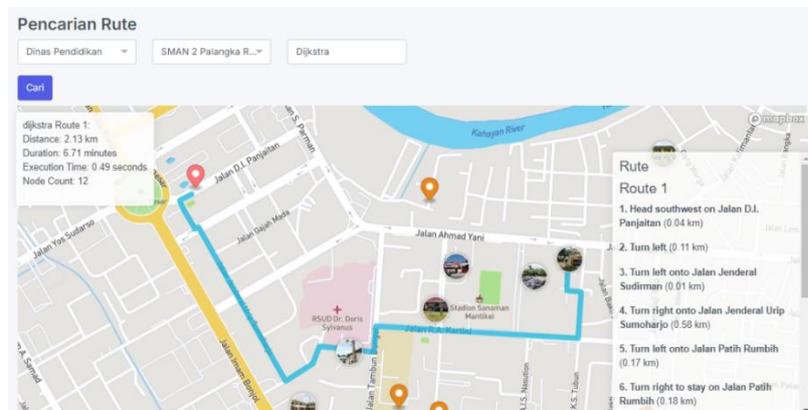
1. Rute pertama yakni dari Dinas Pendidikan Provinsi Kalimantan Tengah, ke Jl. DI. Panjaitan, ke simpang 3 Jl. Jend. Urip sumoharjo, ke simpang 3 Jl. Gajah Mada, ke simpang 3 Jl. Hausman Baboe, ke simpang 4 Jl. Let. Jend. Suprpto dengan Jl. Jend. Urip Sumoharjo, ke simpang 3 Jl. Dr. Soetomo dengan Jl. Jend. Urip Sumoharjo, ke simpang 3 Jl. Patih Rumbih dengan Jl. Jend. Urip Sumoharjo, ke simpang 3 Jl. Tambun Bungai dengan Jl. Patih Rumbih, ke simpang 3 Jl. Tambun Bungai dengan Jl. R. A. Kartini, ke simpang 3 Jl. W. Sudirohusodo dengan Jl. R. A. Kartini, ke simpang 3 Jl. R. A. Kartini dengan Jl. AIS. Nasution, ke simpang 4 Jl. R. A. Kartini, ke Jl. K. S. Tubun dan berakhir di SMAN 2 Palangka Raya. Pada rute pertama ini dinotasikan $A \rightarrow B \rightarrow E \rightarrow J \rightarrow O \rightarrow T \rightarrow Y \rightarrow B1 \rightarrow C1 \rightarrow D1 \rightarrow G1 \rightarrow J1 \rightarrow M1 \rightarrow N1 \rightarrow Z$ dengan bobot rute $73 \text{ m} + 167 \text{ m} + 99 \text{ m} + 75 \text{ m} + 82 \text{ m} + 155 \text{ m} + 160 \text{ m} + 334 \text{ m} + 95 \text{ m} + 215 \text{ m} + 238 \text{ m} + 166 \text{ m} + 155 \text{ m} + 125 \text{ m} = 2.139 \text{ km}$
2. Rute kedua yakni dari Dinas Pendidikan Provinsi Kalimantan Tengah, ke Jl. DI. Panjaitan, ke simpang 4 Jl. S. Parman, ke simpang 4 Jl. S. Parman dengan Jl. Jend. Sudirman, ke simpang 4 Jl. Ahmad Yani, ke simpang 4 Jl. Ahmad Yani dengan Jl. Let. Jend. Suprpto, ke simpang 4 Jl. Ahmad Yani dengan Jl. Tambun Bunga, ke simpang 3 Jl. W. Sudirohusodo dengan Jl. Ahmad Yani, ke simpang 3 Jl. W. Sudirohudodo dengan Jl. R. A. Kartini, ke simpang 3 Jl. R. A. Kartini dengan Jl. AIS. Nasution, ke simpang 4 Jl. R. A. Kartini, ke Jl.

K. S. Tubun dan berakhir di SMAN 2 Palangka Raya. Pada rute kedua ini dinotasikan dengan $A \rightarrow B \rightarrow C \rightarrow I \rightarrow N \rightarrow R \rightarrow X \rightarrow H1 \rightarrow G1 \rightarrow J1 \rightarrow M1 \rightarrow N1 \rightarrow Z$ dengan bobot rute $73 \text{ m} + 492 \text{ m} + 168 \text{ m} + 64 \text{ m} + 117 \text{ m} + 144 \text{ m} + 212 \text{ m} + 317 \text{ m} + 238 \text{ m} + 166 \text{ m} + 155 \text{ m} + 125 \text{ m} = 2.271 \text{ km}$

3. Rute ketiga yakni dari Dinas Pendidikan Provinsi Kalimantan Tengah, ke Jl. DI. Panjaitan, ke simpang 4 Jl. Jend. Sudirman, ke simpang 3 Jl. Imam Bonjol dengan Jl. Let. Jend. Suprpto, ke simpang 3 Jl. Imam Bonjol dengan Jl. Patih Rumbih, ke simpang 3 Jl. Diponegoro dengan Jl. Tambun Bungai, ke simpang 3 Jl. Diponegoro dengan Jl. W. Sudirohusodo, ke simpang 3 Jl. Diponegoro dengan Jl. AIS. Nasution, ke simpang 3 Jl. Diponegoro dengan Jl. K. S. Tubun, ke simpang 4 Jl. R. A. Kartini, ke Jl. K. S. Tubun dan berakhir di SMAN 2 Palangka Raya. Pada rute ketiga ini dinotasikan dengan $A \rightarrow B \rightarrow D \rightarrow S \rightarrow A1 \rightarrow E1 \rightarrow F1 \rightarrow I1 \rightarrow L1 \rightarrow M1 \rightarrow N1 \rightarrow Z$ dengan bobot rute $73 \text{ m} + 209 \text{ m} + 263 \text{ m} + 318 \text{ m} + 420 \text{ m} + 211 \text{ m} + 288 \text{ m} + 166 \text{ m} + 377 \text{ m} + 155 \text{ m} + 125 \text{ m} = 2.605 \text{ km}$

Rute keempat yakni dari Dinas Pendidikan Provinsi Kalimantan Tengah, ke ke Jl. DI. Panjaitan, ke simpang 4 Jl. Jend. Sudirman, ke simpang 3 Jl. Imam Bonjol dengan Jl. Let. Jend. Suprpto, ke simpang 3 Jl. Imam Bonjol dengan Jl. Patih Rumbih, ke simpang 3 Jl. Patih Rumbih dengan Jl. Jend. Urip Sumoharjo, ke simpang 3 Jl. Tambun Bungai dengan Jl. Patih Rumbih, ke simpang 3 Jl. Tambun Bungai dengan Jl. R. A. Kartini, ke simpang 3 Jl. W. Sudirohusodo dengan Jl. R. A. Kartini, ke simpang 3 Jl. R. A. Kartini dengan Jl. AIS. Nasution, ke simpang 4 Jl. R. A. Kartini, ke Jl. K. S. Tubun dan berakhir di SMAN 2 Palangka Raya. Pada rute ketiga ini dinotasikan dengan $A \rightarrow B \rightarrow D \rightarrow S \rightarrow A1 \rightarrow B1 \rightarrow C1 \rightarrow D1 \rightarrow G1 \rightarrow J1 \rightarrow M1 \rightarrow N1 \rightarrow Z$ dengan bobot rute $73 \text{ m} + 209 \text{ m} + 263 \text{ m} + 318 \text{ m} + 100 \text{ m} + 334 \text{ m} + 95 \text{ m} + 215 \text{ m} + 238 \text{ m} + 166 \text{ m} + 155 \text{ m} + 125 \text{ m} = 2.291 \text{ km}$.

Dapat disimpulkan bahwa dari penjabaran di atas, dari keempat rute yang dapat dilalui dari Dinas Pendidikan Provinsi Kalimantan Tengah ke SMAN 2 Palangka Raya, diperoleh rute $A \rightarrow B \rightarrow E \rightarrow J \rightarrow O \rightarrow T \rightarrow Y \rightarrow B1 \rightarrow C1 \rightarrow D1 \rightarrow G1 \rightarrow J1 \rightarrow M1 \rightarrow N1 \rightarrow Z$ sebagai rute terpendek dengan total bobot terkecil, yaitu 2.139 km. Penjabaran rute tersebut adalah sebagai berikut : Dinas Pendidikan Provinsi Kalimantan Tengah, ke Jl. DI. Panjaitan, ke simpang 3 Jl. Jend. Urip sumoharjo, ke simpang 3 Jl. Gajah Mada, ke simpang 3 Jl. Hausman Baboe, ke simpang 4 Jl. Let. Jend. Suprpto dengan Jl. Jend. Urip Sumoharjo, ke simpang 3 Jl. Dr. Soetomo dengan Jl. Jend. Urip Sumoharjo, ke simpang 3 Jl. Patih Rumbih dengan Jl. Jend. Urip Sumoharjo, ke simpang 3 Jl. Tambun Bungai dengan Jl. Patih Rumbih, ke simpang 3 Jl. Tambun Bungai dengan Jl. R. A. Kartini, ke simpang 3 Jl. W. Sudirohusodo dengan Jl. R. A. Kartini, ke simpang 3 Jl. R. A. Kartini dengan Jl. AIS. Nasution, ke simpang 4 Jl. R. A. Kartini, ke Jl. K. S. Tubun dan berakhir di SMAN 2 Palangka Raya, sehingga menghasilkan rute yang ditampilkan pada gambar 4.32 dan 4.33 berikut ini :



Gambar. 4.4. Tampilan Web Dari Dinas Pendidikan Provinsi Kalimantan Tengah Menuju SMAN 2 Palangka Raya

2. Pengujian Algoritma A-Star

Untuk pencarian rute sekolah terdekat menggunakan algoritma *A-star*, diperlukan lokasi titik awal yang akan digunakan dalam implementasi *formula Haversine*. Dalam penelitian ini, lokasi titik awal adalah Dinas Pendidikan Provinsi Kalimantan Tengah. Data yang dikumpulkan mencakup nama sekolah, alamat lengkap, *latitude* dan *longitude* dari lokasi akhir adalah sebagai berikut :

Tabel 4.2. Data Lengkap Lokasi Akhir

No	Nama Sekolah	Alamat	Latitude	Longitude
1	SMAN 2 Palangka Raya	JL. KS Tubun No. 02	- 2.20940	113.92907
2	SMAN 3 Palangka Raya	JL. G. Obos NO. 12	- 2.21618	113.91689
3	SMAN 4 Palangka Raya	JL. Sisinga Magaraja III	- 2.22344	113.90993
4	SMA Muhammadiyah 1 Palangka Raya	JL. RTA Milono KM. A.5	- 2.21984	113.92098

Tabel 4. 1. Data Lengkap Lokasi Awal

No	Lokasi Awal	Alamat	Latitude	Longitude
1	Dinas Pendidikan Provinsi Kalimantan Tengah	Jl. Mayjen DI Pandjaitan	-2.20761	113.91808

1. Dari Dinas Pendidikan Provinsi Kalimantan Tengah ke SMAN 2 Palangka Raya
 - a. Lokasi Awal : Dinas Pendidikan Provinsi Kalimantan Tengah
Latitude : -2.20761
Longitude : 113.91808
 - b. Lokasi Akhir : SMAN 2 Palangka Raya
Latitude : -2.20940
Longitude : 113.92907
 Heuristik : 1.27 km
 : 1.270 m
2. Dari Dinas Pendidikan Provinsi Kalimantan Tengah ke SMAN 3 Palangka Raya
 - a. Lokasi Awal : Dinas Pendidikan Provinsi Kalimantan Tengah
Latitude : -2.20761
Longitude : 113.91808
 - b. Lokasi Akhir : SMAN 3 Palangka Raya
Latitude : -2.21618
Longitude : 113.91689
 Heuristik : 1.02 km
 : 1.020 m
3. Dari Dinas Pendidikan Provinsi Kalimantan Tengah ke SMAN 4 Palangka Raya
 - a. Lokasi Awal : Dinas Pendidikan Provinsi Kalimantan Tengah
Latitude : -2.20761
Longitude : 113.91808
 - b. Lokasi Akhir : SMAN 4 Palangka Raya
Latitude : -2.22344
Longitude : 113.90993
 Heuristik : 1.98 km
 : 1.980 m

4. Dari Dinas Pendidikan Provinsi Kalimantan Tengah ke SMA Muhammadiyah 1 Palangka Raya

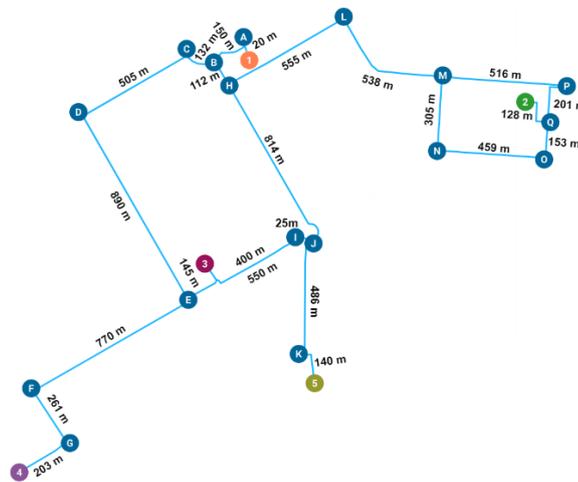
a. Lokasi Awal : Dinas Pendidikan Provinsi Kalimantan Tengah

Latitude : -2.20761
 Longitude : 113.91808

b. Lokasi Akhir : SMA 1 Muhammadiyah Palangka Raya

Latitude : -2.21984
 Longitude : 113.92098
 Heuristik : 1.34 km
 : 1.340 m

Setelah menentukan nilai heuristik, langkah berikutnya adalah menghitung nilai $g(n)$, yaitu jarak total dari lokasi awal ke akhir menggunakan fitur "Measure distance" pada Google maps. Fitur ini mengukur jarak antar simpul sesuai jalur jalan sebenarnya, bukan garis lurus. Berikut gambar simpul-simpul di sekitar 4 SMA terdekat dari Dinas Pendidikan Provinsi Kalimantan Tengah.

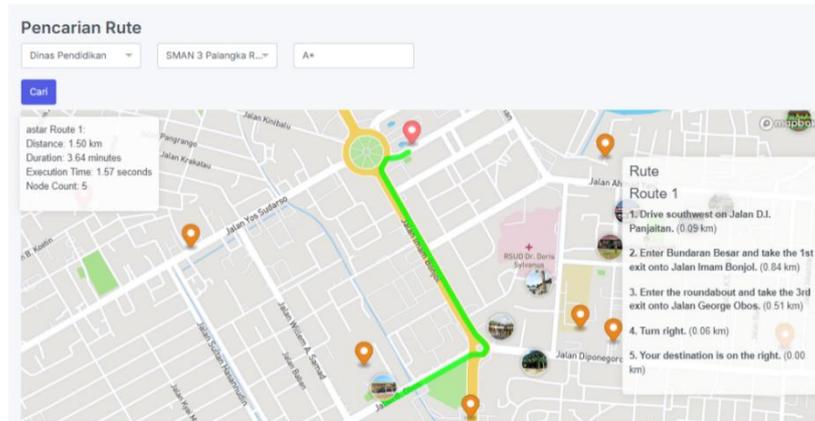


Gambar 4.5. Contoh Simpul

Tabel 4. 2. Nilai $f(n)$ Setiap Simpul Dari Dinas Pendidikan Provinsi Kalimantan Tengah Menuju SMAN 3 Palangka Raya

No	Simpul	No	Simpul
1	<p>Simpul A</p> $f(n) = g(n) + h(n)$ $= 20 + 1,070$ $= 21.070 \text{ m}$	4	<p>Simpul J</p> $f(n) = g(n) + h(n)$ $= 814 + 1,070$ $= 815.070 \text{ m}$
2	<p>Simpul B</p> $f(n) = g(n) + h(n)$ $= 150 + 1,070$ $= 151.070 \text{ m}$	5	<p>Simpul I</p> $f(n) = g(n) + h(n)$ $= 25 + 1,070$ $= 26.070 \text{ m}$
3	<p>Simpul H</p> $f(n) = g(n) + h(n)$ $= 112 + 1,070$ $= 113.070 \text{ m}$	6	<p>Simpul End</p> $f(n) = g(n) + h(n)$ $= 400 + 1,070$ $= 401.070 \text{ m}$
<p>Total : 1.527.420 m : 1.52742 km</p>			

Hasil total perhitungan jarak dari Dinas Pendidikan Provinsi Kalimantan Tengah menuju SMAN 3 Palangka Raya menggunakan algoritma A-star menunjukkan jarak sebesar **1.527.420 meter** atau **1.52742 kilometer**. Sebagai bahan pertimbangan dan perbandingan, hasil perhitungan ini juga dibandingkan dengan informasi yang diperoleh dari Website Pencarian Rute Terpendek serta Google maps.



Gambar 4.6. Hasil Dari Website

4.3 Perbandingan Algoritma

Terdapat empat aspek yang diteliti dalam perbandingan ini, yaitu perbedaan jarak, perbedaan durasi waktu tempuh, perbedaan waktu eksekusi, dan perbedaan jumlah *node* yang diperiksa. Rincian percobaan algoritma dapat dilihat pada Tabel 4.5, sementara hasil perbandingan algoritma ditampilkan pada Tabel 4.6 dan 4.7.

Tabel 4. 5. Detail Percobaan Algoritma

No	Pengguna		SMA Tujuan		Nama SMA	Nama Percobaan
	Latitude	Longitude	Latitude	Longitude		
1	-2.20761	113.91808	-2.21008	113.92805	SMAN 1	Percobaan 1
2	-2.20761	113.91808	-2.20966	113.92891	SMAN 2	Percobaan 2
3	-2.20761	113.91808	-2.21642	113.91693	SMAN 3	Percobaan 3
4	-2.20761	113.91808	-2.22358	113.9098	SMAN 4	Percobaan 4
5	-2.20761	113.91808	-2.21397	113.92111	SMA Kristen	Percobaan 5
6	-2.20761	113.91808	-2.22009	113.9206	Muhammadiyah	Percobaan 6
7	-2.20761	113.91808	-2.18098	113.88634	MAN Model	Percobaan 7
8	-2.20761	113.91808	-2.21116	113.92504	Isen Mulang	Percobaan 8

Tabel 4. 6. Hasil Perbandingan Algoritma (Jarak & Durasi Tempuh)

No	Nama Percobaan	Jarak (km)			Durasi Tempuh (menit)		
		Dijkstra	A-star	Selisih	Dijkstra	A-star	Selisih
1	Percobaan 1	2.02	1.85	0.17	6.74	5.52	1.22
2	Percobaan 2	2.13	1.94	0.19	6.71	5.53	1.18
3	Percobaan 3	2.65	1.50	1.15	9.05	3.61	5.44
4	Percobaan 4	3.83	2.89	0.94	12.47	6.57	5.90
5	Percobaan 5	1.66	0.95	0.71	7.48	2.31	5.17
6	Percobaan 6	2.29	1.62	0.67	7.49	4.06	3.43
7	Percobaan 7	7.51	4.94	2.57	22.95	8.79	14.16
8	Percobaan 8	1.44	1.61	0.17	4.62	4.41	0.21

Tabel 4. 7. Hasil Perbandingan Algoritma (Lama Eksekusi & Jumlah Node)

No	Nama Percobaan	Lama Eksekusi (s)			Jumlah Node		
		Dijkstra	A-star	Selisih	Dijkstra	A-star	Selisih
1	Percobaan 1	0.72	0.34	0.38	13	7	6
2	Percobaan 2	1.22	0.39	0.83	12	7	5
3	Percobaan 3	1.05	0.31	0.74	16	5	11
4	Percobaan 4	0.18	0.16	0.02	20	6	14
5	Percobaan 5	1.44	0.25	1.19	10	4	6
6	Percobaan 6	1.03	0.34	0.69	13	6	7
7	Percobaan 7	1.07	0.32	0.75	32	4	28
8	Percobaan 8	0.14	0.14	0	10	8	2

Berdasarkan data yang telah dianalisis dari percobaan penggunaan algoritma *Dijkstra* dan *A-star*, berikut kesimpulan yang dapat diambil :

1. Rata-rata selisih eksekusi antara kedua algoritma menunjukkan bahwa algoritma *A-star* secara umum lebih cepat dalam melakukan eksekusi, dengan rata-rata selisih 0,57 detik. Algoritma *A-star* cenderung lebih efisien dalam hal waktu eksekusi karena memanfaatkan heuristik yang mengarahkan pencarian ke *node* yang lebih relevan, mengurangi waktu yang dibutuhkan.
2. Jumlah *node* yang dicek juga menunjukkan bahwa *A-star* lebih efisien. Rata-rata selisih jumlah *node* yang diperiksa adalah 9 *node* lebih sedikit pada *A-star* dibandingkan *Dijkstra*. Ini berarti *A-star* memerlukan lebih sedikit eksplorasi terhadap *node*, mengurangi jumlah operasi yang harus dilakukan.
3. Namun, pada beberapa kasus tertentu, seperti pada Percobaan 8, algoritma *Dijkstra* dapat lebih unggul. Dalam percobaan ini, meskipun waktu eksekusi kedua algoritma sama, *Dijkstra* mampu menemukan rute yang lebih efisien dalam hal jarak (lebih pendek 0,17 km), durasi tempuh (lebih cepat 0,21 menit), dan jumlah *node* yang diperiksa (2 *node* lebih sedikit) dibandingkan *A-star*. Hal ini dapat terjadi pada kasus graf tertentu, di mana struktur dan penempatan simpul membuat penggunaan heuristik pada *A-star* kurang efektif, sehingga eksplorasi menyeluruh yang dilakukan *Dijkstra* menjadi lebih optimal.

Rata-rata selisih jarak dan durasi tempuh juga tidak menunjukkan perbedaan signifikan antara kedua algoritma. Perbedaan jarak rata-rata adalah 0,78 km, dan perbedaan waktu tempuh rata-rata adalah 4,71 menit. Ini menunjukkan bahwa dalam hal hasil rute akhir, baik *A-star* maupun *Dijkstra* cenderung menghasilkan solusi yang serupa.

Berdasarkan hasil perbandingan algoritma *Dijkstra* dan *A-star* dari percobaan 1 hingga 8, kita dapat menarik kesimpulan menyeluruh mengenai performa kedua algoritma tersebut. Dari tabel yang diberikan, terlihat bahwa *A-star* secara konsisten lebih unggul dalam hal waktu eksekusi dan jumlah *node* yang diperiksa pada 7 percobaan pertama. Ini terjadi karena *A-star* menggunakan heuristik (dalam hal ini, *formula Haversine*) yang memandu pencariannya menuju tujuan dengan lebih cepat. Heuristik ini bekerja dengan memperkirakan jarak garis lurus dari *node* saat ini ke *node* tujuan, sehingga *A-star* tidak perlu mengeksplorasi semua kemungkinan rute, melainkan hanya berfokus pada jalur yang diperkirakan lebih mendekati tujuan. Hal ini memungkinkan *A-star* untuk memeriksa lebih sedikit *node* dan mempercepat waktu eksekusi secara signifikan, terutama pada graf yang relatif sederhana atau dengan struktur jalur yang lebih linear.

Namun, rute yang dihasilkan oleh *Dijkstra* dan *A-star* dapat berbeda, meskipun mereka menggunakan graf yang sama. Ini terjadi karena pendekatan yang digunakan masing-masing algoritma dalam mencari rute terpendek. *Dijkstra*, yang tidak menggunakan heuristik, memeriksa semua *node* di graf secara menyeluruh tanpa memperkirakan jarak ke tujuan. Ini berarti bahwa *Dijkstra* akan menemukan rute yang benar-benar optimal dalam hal jarak atau biaya, karena ia mempertimbangkan setiap kemungkinan jalur yang ada. Sebaliknya, *A-star* berfokus pada jalur yang tampak lebih mendekati tujuan berdasarkan jarak garis lurus, tetapi tidak selalu mempertimbangkan panjang jalur yang sebenarnya di graf. Pada graf dengan banyak simpul atau rute berbelok, heuristik yang digunakan oleh *A-star* mungkin memberikan estimasi yang kurang akurat, sehingga bisa menghasilkan rute yang berbeda dan kadang-kadang sedikit lebih panjang dari *Dijkstra*.

Kesimpulannya, *A-star* lebih efisien pada graf linear atau kurang kompleks, karena heuristik mempercepat pencarian rute. Sebaliknya, *Dijkstra* lebih akurat pada graf kompleks yang membutuhkan eksplorasi menyeluruh. Pemilihan algoritma tergantung pada jenis graf yang digunakan dimana *A-star* cocok untuk graf linear, sementara *Dijkstra* lebih efektif untuk graf yang kompleks.

5. KESIMPULAN

Berdasarkan rumusan masalah dan hasil penelitian, sistem pencarian rute terpendek berbasis *website* untuk Sekolah Menengah Atas (SMA) di Kota Palangka Raya dirancang menggunakan metode *Waterfall* (Sommerville, 2011). Setelah tahap perancangan, sistem diimplementasikan dengan menggunakan bahasa pemrograman PHP dengan *framework CodeIgniter*, serta *JavaScript* untuk pengelolaan tata letak peta yang disediakan oleh *Mapbox*. Sistem ini juga memanfaatkan *database MySQL* yang dioperasikan pada server *database*.

Berdasarkan hasil perbandingan algoritma *Dijkstra* dan *A-star* dari percobaan 1 hingga percobaan 8, diperoleh beberapa temuan penting. Dari aspek jarak dan durasi tempuh, *A-star* secara umum menghasilkan rute yang lebih pendek, dengan selisih rata-rata jarak sekitar 0,78 km dan waktu tempuh lebih cepat dengan selisih rata-rata 4,71 menit dibandingkan *Dijkstra*. Dalam hal waktu eksekusi, *A-star* terbukti lebih unggul dengan selisih rata-rata 0,57 detik lebih cepat, serta memeriksa 9 *node* lebih sedikit berkat penggunaan heuristik *Haversine*. Namun, pada kasus tertentu, seperti pada Percobaan 8, algoritma *Dijkstra* justru lebih unggul. Pada percobaan ini, meskipun waktu eksekusi kedua algoritma sama, *Dijkstra* berhasil menemukan rute yang lebih efisien dalam hal jarak (lebih pendek 0,17 km), durasi tempuh (lebih cepat 0,21 menit), dan jumlah *node* yang diperiksa (2 *node* lebih sedikit) dibandingkan *A-star*. Hal terjadi pada graf kompleks atau jalur tidak linear, di mana heuristik *A-star* kurang efektif, sedangkan eksplorasi menyeluruh *Dijkstra* lebih optimal. Pemilihan algoritma bergantung pada kebutuhan, *A-star* lebih cocok untuk graf sederhana karena efisiensinya, sementara *Dijkstra* unggul pada graf kompleks meskipun waktu eksekusinya lebih lama. Kesimpulan merupakan pernyataan singkat tentang hasil dan temuan yang didapatkan.

6. SARAN

Berdasarkan hasil penelitian, disarankan agar pemilihan algoritma pencarian rute terpendek disesuaikan dengan kondisi dan kebutuhan spesifik lingkungan yang dihadapi. Algoritma *A-star* sebaiknya digunakan pada graf yang relatif sederhana atau memiliki struktur linear, karena kemampuannya menghasilkan rute dengan efisiensi waktu lebih baik, serta jumlah *node* yang diperiksa lebih sedikit. Namun, pada graf yang lebih kompleks atau dalam lingkungan dengan banyak hambatan, algoritma *Dijkstra* dapat memberikan hasil yang lebih optimal meskipun membutuhkan waktu eksekusi lebih lama.

Penggunaan heuristik juga bisa diperluas dengan mempertimbangkan *Vincenty's Formula*, yang memiliki keunggulan dalam perhitungan jarak antara dua titik di permukaan bumi dengan lebih akurat dibandingkan *Haversine Formula*. *Formula* ini dapat digunakan untuk meningkatkan akurasi pencarian rute terutama pada area yang memerlukan perhitungan jarak lebih presisi. Dengan penerapan algoritma yang tepat dan penambahan *formula* heuristik seperti *Vincenty*, sistem pencarian rute diharapkan dapat lebih efisien dalam memberikan panduan bagi pengguna, khususnya dalam aplikasi yang lebih kompleks di wilayah perkotaan seperti Kota Palangka Raya.

DAFTAR PUSTAKA

- Adi, N. H., Giatman, M., Simatupang, W., Afrina, A., & Watrianthos, R. (2021). Penerapan metode *Dijkstra* pada jalur distribusi LPG untuk penentuan jarak terpendek. *Building of Informatics, Technology and Science (BITS)*, 3(3), 235-243.
- Al Hakim, R. R., Satria, M. H., Arief, Y. Z., Pangestu, A., Jaenul, A., Hertin, R. D., & Nugraha, D. (2021). Aplikasi Algoritma *Dijkstra* dalam Penyelesaian Berbagai Masalah. *Expert*, 11(1), 345994.
- ARDYAN, S., MULYONO & SUYITNO, A. (2017) 'Implementasi Algoritma *Dijkstra* Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten', *UNNES Journal of Mathematics*, 6(2), pp. 108–116.
- Aziz, M. A. (2023, March). Implementasi Algoritma *Dijkstra* dalam Menentukan Rute Terpendek Terhadap Destinasi Wisata Kabupaten Bojonegoro. In *Prosiding Seminar Pendidikan Matematika dan Matematika (Vol. 7)*.

-
- Cantona, A., Fauziah, F., & Winarsih, W. (2020). Implementasi algoritma *Dijkstra* pada pencarian rute terpendek ke museum di Jakarta. *Jurnal Teknologi dan Manajemen Informatika*, 6(1), 27-34.
- Miftahuddin, Y., Umaroh, S., & Karim, F. R. (2020). Perbandingan metode perhitungan jarak euclidean, *Haversine*, dan manhattan dalam penentuan posisi karyawan (Studi Kasus: Institut Teknologi Nasional Bandung). *Jurnal Tekno Insentif*, 14(2), 69-77.
- Perayoga, R., Hendradi, P., & Setiawan, A. (2023). Implementasi Algoritma *Dijkstra* Pada Pencarian Rute Terpendek Objek Wisata. *KLIK: Kajian Ilmiah Informatika dan Komputer*, 4(3), 1471-1482.
- Prasetyo, A. C., Arnandi, M. P., Hudnanto, H. S., & Setiaji, B. (2019). Perbandingan Algoritma Astar dan Dijkstra Dalam Menentukan Rute Terdekat. *Sisfotenika*, 9(1), 36-46.
- Rahayu, S., Fanni, R., & Bima, K. (2022). Perbandingan *Haversine Formula* dan Euclidean Distance dalam Pencarian Jarak Terdekat Rumah Penampungan Hewan (Shelter). *Jurnal Ilmiah FIFO*, 14(1), 25-34.
- Rosita, Y. D., Rosyida, E. E., & Rudiyanto, M. A. (2019). Implementation of *Dijkstra* algorithm and multi-criteria decision-making for optimal route distribution. *Procedia Computer Science*, 161, 378-385.
- Setiawan, W. (2010). Pembahasan pencarian lintasan terpendek menggunakan algoritma *Dijkstra* dan *A-STAR*. *J. Mat. ITB*.
- SUNARDI, S., YUDHANA, A. & KADIM, A. A. (2019) 'Implementasi Algoritma *Dijkstra* dan Algoritma Semut Untuk Analisis Rute Transjogja Berbasis *Android*', *It Journal Research and Development*, 01, pp. 32–38. doi: 10.25299/itjrd.2019.vol4(1).2483.
- Sunardi, S., Yudhana, A., & Kadim, A. A. (2019). Implementasi Algoritma *Dijkstra* Untuk Analisis Rute Transportasi Umum Transjogja Berbasis *Android*. *J. Sist. Inf. Bisnis*, 9(1), 32.
- Umar, R., Yudhana, A., & Prayudi, A. (2021). Analisis Perbandingan Algoritma *Dijkstra*, *A-star*, dan Floyd Warshall dalam Pencarian Rute Terdekat pada Objek Wisata Kabupaten Dompu. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(2), 227-234.
- Yogaswara, D., & Suhartono, S. (2017). Perbandingan Algoritma *A-star* dan *Dijkstra* pada Pencarian Jalur Evakuasi Tsunami Terpendek Menuju Shelter di Kabupaten Bantul Berbasis Aplikasi *Android*. *Jurnal Masyarakat Informatika*, 12(1), 10-18.