Journal of Artificial Intelligence and Software Engineering
Vol. 5, No. 2, Juni 2025, pp. 606~615
E-ISSN: 2777-001X, DOI: 10.30811/jaise.v5i2.6839

Development of the Shortest Path Navigation Feature in a 360° Virtual Campus Tour Using Dijkstra's Algorithm

Muhammad Riza Alifi[1*], Ade Hodijah[1], Urip Teguh Setijohatmo[1], Sri Ratna Wulan[1], Hashri Hayati[1]
[1] Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Kab. Bandung Barat, 40559, Indonesia

Informasi Artikel

Diterima : 30 April 2025
Revisi : 25 Mei 2025
Publikasi : 20 Juni 2025

Kata Kunci:

Virtual Campus Tour
Virtual Tour
Virtual Reality
Virtual Reality Tour
Shortest Path
Dijkstra

ABSTRAK

360° virtual campus tour memungkinkan pengguna untuk menjelajahi seluruh pemandangan dalam format foto panorama 360° secara mandiri melalui fitur navigasi yang bersifat *self-guided.* Namun, tidak semua fitur navigasi yang tersedia mampu memberikan rekomendasi rute eksplorasi yang optimal bagi pengguna. Hal ini menjadi kendala, terutama ketika jumlah pemandangan mencapai ratusan, karena pengguna dapat mengalami kebingungan dalam menentukan titik awal dan akhir penjelajahan. Dalam beberapa kasus, interaksi yang terlalu lama di lingkungan virtual dapat menimbulkan ketidaknyamanan, seperti *motion sickness.* Penerapan algoritma pencarian rute terpendek menjadi salah satu solusi potensial untuk mempermudah eksplorasi berdasarkan rute yang direkomendasikan, sekaligus mengurangi durasi interaksi di lingkungan virtual. Studi ini mengembangkan fitur navigasi pencarian rute terpendek berbasis algoritma Dijkstra pada *virtual campus tour*, yang terdiri atas: (1) komponen navigasi pada sisi *front-end* untuk antarmuka pengguna dalam pencarian rute, dan (2) komponen pemrosesan rute pada sisi *back-end* dengan pendekatan struktur graf. Hasil implementasi menunjukkan bahwa fitur navigasi rute ini memiliki performa yang efisien, dengan rata-rata waktu eksekusi sebesar 4,94 ms dan konsumsi memori yang rendah, ditunjukkan oleh *resident set size* sebesar 710,47 KB dan *used heap* sebesar 668,61 KB.

ABSTRACT

A 360° virtual campus tour allows users to independently explore all available scenes in the form of 360° panoramic photos through a self-guided navigation feature. However, not all navigation tools provided are capable of generating route recommendations for users to follow. This presents a challenge, as users may feel overwhelmed when deciding where to begin and end the tour—particularly when the number of scenes reaches into the hundreds. In certain scenarios, prolonged interaction within a virtual reality environment may lead to discomfort due to motion sickness. Implementing a shortest path algorithm offers a potential solution by guiding users through recommended routes, thereby improving exploration efficiency and reducing interaction time. This study integrates a shortest path-based navigation feature into a virtual campus tour using Dijkstra's algorithm, consisting of: (1) a front-end navigation component for the user interface of route searching, and (2) a back-end routing component that processes pathfinding using a graph-based structure. The implemented navigation feature demonstrates high efficiency, with an average execution time of only 4.94 ms and low memory consumption, as measured by a resident set size of 710.47 KB and used heap memory of 668.61 KB.

Penulis Koresponden

Email: muhammad.riza@polban.ac.id

Cara sitasi IEEE:

M. R. Alifi, A. Hodijah, U. T. Setijohatmo, S. R. Wulan, & Hayati H., "Development of the Shortest Path Navigation Feature in a 360° Virtual Campus Tour Using Dijkstra's Algorithm," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 2, pp. 606-615, Juni 2025. doi: 10.30811/jaise.v5i2.6839

1. PENDAHULUAN

Virtual campus tour merupakan suatu media *virtual reality* yang umumnya berbasis web, disediakan oleh perguruan tinggi untuk memberikan pengalaman berkeliling di area kampus secara digital dengan menggunakan teknologi foto panorama 360° [1], [2], [3]. *Virtual campus tour* sangat berguna bagi calon mahasiswa untuk melihat lingkungan kampus dari berbagai sudut dan ruangan tanpa perlu berkunjung langsung secara fisik, khususnya bagi calon mahasiswa yang terkendala waktu, jarak dan transportasi [1], [4]. Bagi beberapa calon mahasiswa dan orang tua calon mahasiswa, fasilitas untuk mengeksplorasi secara visual area kampus merupakan aspek yang krusial dalam memotivasi mereka untuk memilih kampus yang akan dituju [1], [5]. Selain itu, bagi mahasiswa baru, *campus virtual tour* dapat dimanfaatkan sebagai alat bantu untuk memudahkan proses adaptasi dengan lingkungan kampus pada periode orientasi [2].

Campus tour yang dilakukan secara langsung, seringkali diorganisir dengan melibatkan pemandu agar dapat memberikan pengalaman dan kesan positif kepada calon mahasiswa, terutama pada saat mengenalkan fasilitas kampus yang menjadi bahan pertimbangan bagi calon mahasiswa baru. Namun, ketersediaan pemandu dan waktu menjadi kendala ketika dalam situasi tertentu [2]. Pada *virtual campus tour*, peran pemandu dapat direpresentasikan dalam bentuk tekstual, audio, dan visual berdasarkan interaksi pengguna [2], [6].

Spesifik mengenai salah satu tugas penting pemandu yaitu mengarahkan partisipan untuk menjelajahi lingkungan kampus berdasarkan rute yang telah direncanakan. *Virtual campus tour* memiliki karakteristik *self-guided*, sehingga pengguna dapat mengeksplorasi seluruh pemandangan (*scene*) pada *virtual campus tour* secara mandiri melalui fitur navigasi yang telah disediakan [2]. Namun, tidak setiap fitur navigasi pada *virtual campus tour* mampu menghasilkan rekomendasi rute yang dapat dijelajahi pengguna. Hal ini dapat menjadi masalah karena pengguna akan kebingungan (*overwhelmed*) dalam mengawali dan mengakhiri *tour*, terutama jika pengguna mengakses *virtual campus tour* yang memiliki banyak titik pemandangan seperti di kampus Politeknik Negeri Bandung (POLBAN), terdapat 164 titik pemandangan. Penyediaan fitur navigasi yang dapat memberikan rekomendasi rute merupakan hal yang penting bagi pengguna, seperti yang telah dibahas pada studi [7] dalam membangun aplikasi *virtual tour guide*.

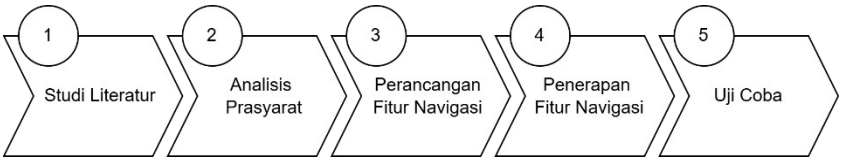
Terdapat studi [8] yang membahas mengenai pembangunan aplikasi dengan pendekatan Dijkstra untuk menentukkan rute terpendek menuju destinasi-destinasi wisata menarik dalam waktu 3-4 detik. Diikuti dengan studi [9] yang memanfaatkan Dijkstra dan Floyd-warshall untuk menentukkan rute terdekat dan terbaik yang dapat ditempuh oleh ambulans menuju rumah sakit dalam penanganan insiden kecelakaan di jalan. Studi lainnya yang menggunakan Dijkstra dan variasinya dapat secara efektif menemukan jalur terpendek dalam *virtual tour*, disertai dengan peningkatan efisiensi, kecepatan, penggunaan memori, dan A* dapat menemukan jalur terpendek dalam kasus seleksi berdasarkan titik dengan nilai heuristik [10], [11], [12].

Dalam skenario tertentu, durasi dalam menggunakan *virtual reality tour* merupakan masalah kritis yang dapat mempengaruhi pengalaman pengguna. Berinteraksi lebih lama dengan lingkungan virtual dapat membuat pengguna kelelahan, bahkan mengalami *motion sickness*. Dengan menerapkan metode pencarian jalur yang optimal, berdampak kepada menurunnya durasi dalam berinteraksi dengan lingkungan virtual [13].

Berdasarkan beberapa studi yang telah dijelaskan sebelumnya [8], [9], [10], [11], [12], [13], studi ini mengusulkan penyelesaian masalah dengan menyediakan fitur navigasi rute pada *virtual campus tour* yang dikembangkan POLBAN (https://vtour.polban.dev/) untuk menentukkan rute terpendek. Studi ini menawarkan kebaruan dalam menerapkan fitur navigasi rute penjelajahan berbasis *shortest path* menggunakan algoritma Dijkstra pada *360° virtual campus tour*. Dibandingkan studi [8] dan [9], penerapan *shortest-path* pada studi yang diusulkan tidak hanya memvisualisasikan pada peta, dalam hal ini menyediakan juga sinkronisasi dengan fitur navigasi pemandangan pada *360° virtual campus tour*.

2. METODE

Studi ini membahas pengembangan fitur navigasi *shortest path* pada *360° virtual campus tour* di kampus POLBAN yang terdiri dari lima tahapan, yaitu: (1) Studi Literatur; (2) Analisis Prasyarat; (3) Perancangan Fitur Navigasi; (4) Pengembangan Fitur Navigasi; dan (5) Uji Coba. Setiap tahapan dilakukan secara sekuensial dan satu arah, sehingga setiap tahapannya harus tuntas sebelum menuju ke tahap berikutnya, seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Metode Penelitian

2.1. Studi Literatur

Mengumpulkan dan mempelajari literatur yang membahas topik pengembangan *360° virtual campus tour*, algoritma yang dapat diterapkan untuk fitur navigasi rute penjelajahan pada *360° virtual campus tour*, khususnya algoritma *shortest path* pada *graph*, pustaka pendukung untuk menerapkan algoritma *shortest path*, dan evaluasi kinerja algoritma *shortest path*.

2.2. Analisis Prasyarat

Menelaah dan mendefinisikan permasalahan, yang selanjutnya mendefinisikan daftar kebutuhan untuk melakukan eksperimen dalam menyelesaikan masalah. Pada tahap ini luaran yang dihasilkan berupa daftar kebutuhan untuk dapat menjalankan skenario tertentu, terdiri dari struktur data, kebutuhan data, interaksi pengguna dengan sistem, komponen sistem, aliran data antar proses dan teknologi pendukungnya.

2.2.1. Struktur Data

Graph dengan representasi *adjacency list* merupakan struktur data yang digunakan pada *360° virtual campus tour* POLBAN, yang utamanya memuat: (a) *identifier* yang unik untuk setiap pemandangan; (b) *vertex* untuk mewakili setiap pemandangan; (c) *edge* untuk menunjukkan keterhubungan antar *vertex*; (d) *weight* sebagai properti bobot pada *edge*, yang dalam hal ini adalah perkiraan jarak antar *vertex* dengan satuan meter.

2.2.2. Kebutuhan Data

Data yang dibutuhkan untuk menjalankan skenario pada studi ini, yaitu: (a) data pemandangan dalam bentuk foto panorama 360°; (b) data deskripsi pemandangan; (c) data *vertex* berdasarkan pemandangan; (d) data *edge* untuk mengetahui keterhubungan antar pemandangan; (e) data *weight* yang bernilai positif untuk setiap *edge*; (f) data koordinat pemandangan (garis lintang dan bujur) untuk menyediakan informasi pendukung lokasi pemandangan dalam bentuk peta. Data yang digunakan pada studi ini dapat dilihat pada Tabel 1.

Tabel 1. Kebutuhan Data

No.	Kategori	Jumlah
1	Pemandangan	164
2	Deskripsi Pemandangan	164
3	*Vertex*	164
	a. *Point of Interest*	49
	b. Persimpangan	13
	c. Jalan dan Lainnya	97
4	*Edge*	372
5	*Weight*	372
6	Koordinat Pemandangan	164

2.2.3. Interaksi Pengguna dengan Sistem

Hubungan pengguna atau aktor yang terlibat dengan sistem, ruang lingkupnya difokuskan hanya untuk pengembangan fitur navigasi pencarian rute terpendek. Analisis interaksi pengguna dengan sistem bertujuan untuk mendefinisikan fitur dari sudut pandang pengguna, serta memecah proses-proses yang

diperlukan agar lebih modular (*use case*) dan memperlihatkan keterkaitannya. Hasil analisis tersebut memperoleh *use case*: (1) menampilkan pemandangan; (2) menampilkan rute; (3) memproses pencarian rute terpendek; (4) menampilkan peta; dan (5) memvisualisasikan rute pada peta.

2.2.4. Proses Bisnis

Proses bisnis yang dianalisis, fokus mengelaborasi proses pencarian rute penjelajahan pada fitur navigasi. Mulai dari pengguna melengkapi input yang diperlukan, pengguna memicu inisiasi proses pencarian, sistem memproses pencarian rute terpendek, sampai dengan sistem menghasilkan rekomendasi rute terpendek untuk dijelajahi secara bertahap oleh pengguna.

2.2.5. Komponen Sistem

Komponen utama sistem terdiri dari: (1) Pemandangan (*front-end*) untuk menayangkan foto panorama 360°; (2) Navigasi merupakan antarmuka untuk memudahkan pengguna dapat menelusuri setiap pemandangan yang hendak dituju; (3) Peta untuk menayangkan secara visual posisi pemandangan berdasarkan garis lintang dan bujur pada bidang datar; (4) Konfigurasi menyediakan opsi pengaturan untuk penayangan pemandangan; (5) Pemandangan (*back-end*) untuk mengelola *asset* pemandangan dan metadata terkait; (6) API (*Application Programming Interface*) untuk menerima *request* dan mengirimkan hasil pemrosesan terkait pencarian rute; dan (7) Rute untuk memproses pencarian rute menggunakan algoritma tertentu. Selain komponen, terdapat juga artifak Graphlib yang merupakan pustaka untuk membangun struktur *graph*.

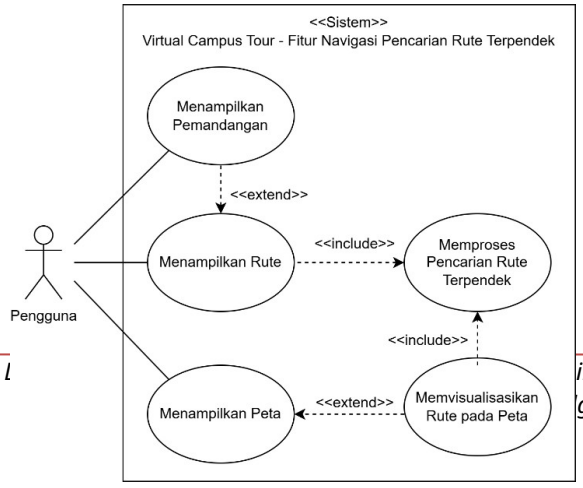
2.2.6. Teknologi Pendukung

Teknologi yang digunakan untuk pengembangan fitur navigasi pencarian rute terpendek, yaitu: (1) Marzipano yang merupakan alat bantu untuk menghasilkan *360° virtual tour* berbasis web [14], sebagai fondasi utama pembentuk *virtual campus tour*; (2) Node.js merupakan *runtime environment* JavaScript [15], dalam hal ini digunakan pada *server-side*, termasuk *API*; (3) Express.js merupakan *framework* untuk melengkapi Node.js [16]; (4) Vue.js merupakan *progressive web framework* yang mengadopsi arsitektur komponen, dalam hal ini digunakan pada *client-side* (*user interface*) [17]; (5) Tailwind CSS sebagai *styling framework* untuk situs web modern [18] yang dapat melengkapi Vue.js; (6) Graphlib (`@dagrejs/graphlib`) adalah JavaScript *library* yang telah menyediakan struktur data untuk fundamental atribut dan operasi *graph* beserta dengan algoritma yang bisa digunakan pada struktur data *graph* [19]; dan (7) OpenLayer yang digunakan untuk menandai pemandangan, terutama untuk visualisasi rute penjelajahan pada peta [20].

2.3. Perancangan Fitur Navigasi

Perancangan yang dilakukan bersumber dari tahapan analisis prasyarat dengan merujuk referensi [21] dan referensi [22]. Rancangan diagram yang dihasilkan meliputi: (1) Interaksi Pengguna dengan Sistem yang direpresentasikan dalam bentuk *Use Case Diagram* pada Gambar 2; (2) Proses Bisnis yang diuraikan dalam bentuk *Activity Diagram* pada Gambar 3; (3) Komponen Sistem yang divisualisasikan dalam bentuk *Deployment Diagram* pada Gambar 4; dan (4) Aliran data pada sistem untuk mendukung konstruksi dan implementasi sistem dalam bentuk *Physical Data Flow Diagram* pada Gambar 5.

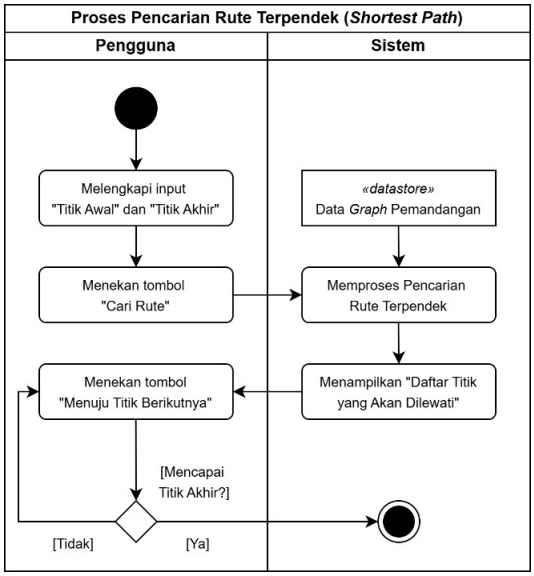
Pada Gambar 2 menunjukkan pengguna berinteraksi dengan *use case* (1) menampilkan pemandangan, yang dalam kondisi tertentu (*extend*) memerlukan (2) menampilkan rute. *Use case* menampilkan rute selalu (*include*) memanggil *use case* (3) memproses pencarian rute terpendek. Pada *use*



case (4) menampilkan peta, yang sifatnya opsional memanggil (*extend*) *use case* (5) memvisualisasikan rute pada peta.

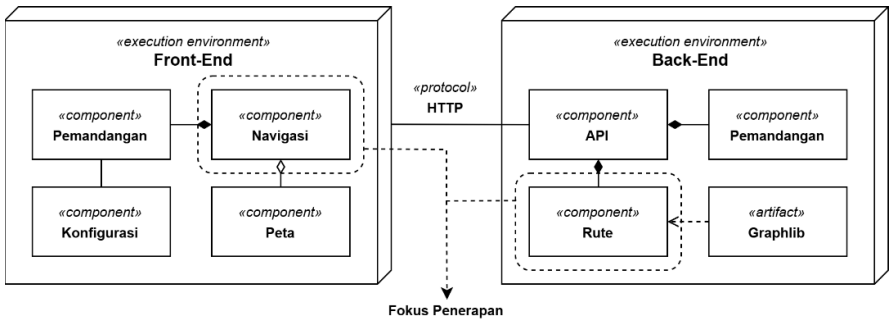
Gambar 2. Interaksi Pengguna dengan Fitur Navigasi Pencarian Rute Terpendek

Pada Gambar 3 menguraikan proses bisnis yang melibatkan pengguna dalam menggunakan fitur navigasi *shortest path*. Dimulai dari pengguna mengisi formulir pencarian rute yang terdiri dari input titik awal dan titik tujuan. Selanjutnya, pengguna perlu menekan tombol "cari rute" agar sistem melakukan proses komputasi untuk menemukan *shortest path* berdasarkan input yang telah diberikan pengguna. Setelah proses komputasi selesai, pengguna memperoleh rekomendasi rute penjelajahan yang ditampilkan berupa daftar titik yang akan dilewati. Pengguna dapat menekan tombol "menuju titik berikutnya" untuk menjelajahi setiap pemandangan secara bertahap, sampai menuju titik akhir.



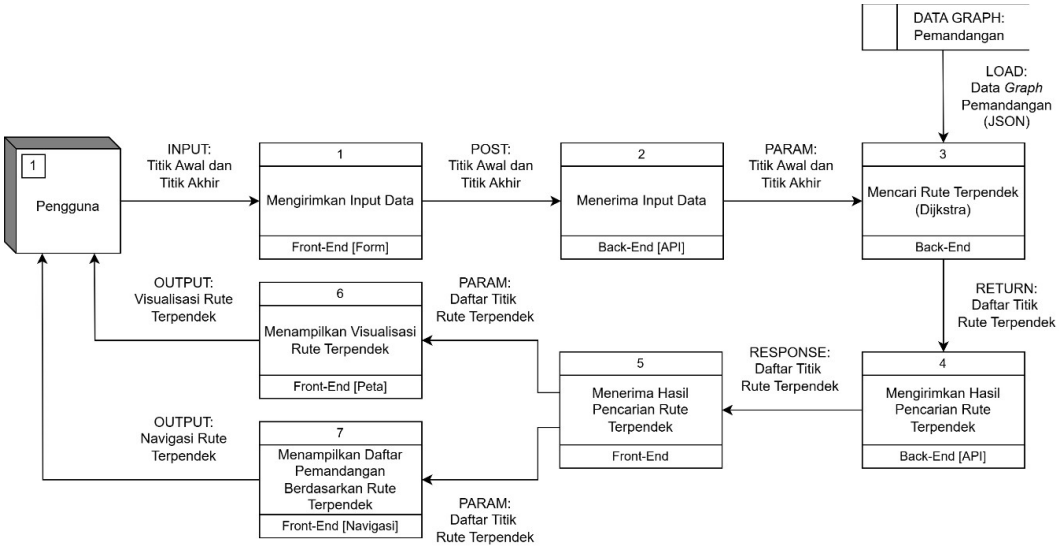
Gambar 3. Proses Pencarian Rute Terpendek pada Fitur Navigasi

Pada Gambar 4 merupakan visualisasi arsitektur *360° Virtual Campus Tour* POLBAN yang telah disertai dengan fitur navigasi pencarian rute terpendek. Arsitektur tersebut terdiri dari dua bagian utama, yaitu: (1) *back-end* yang mengelola seluruh proses komputasi dan *asset* pemandangan, khususnya untuk pemrosesan dalam menghasilkan rute *shortest path*; dan (2) *front-end* yang berfungsi sebagai antarmuka pengguna. Studi ini memfokuskan pengembangan komponen navigasi yang berada pada sisi *front-end*, spesifik untuk antarmuka pengguna pencarian rute, dan komponen rute yang berada pada sisi *back-end* untuk memproses pencarian rute berbasis struktur dan algoritma *graph*.



Gambar 4. Arsitektur *360° Virtual Campus Tour* POLBAN disertai Fitur Navigasi Pencarian Rute Terpendek

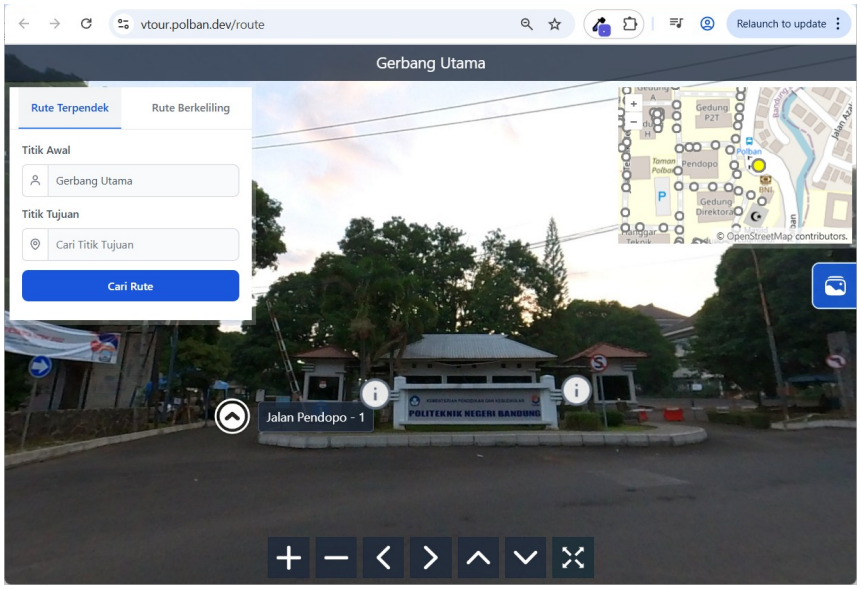
Untuk memudahkan konstruksi dan implementasi sistem, diperlukan model proses yang disertai dengan aliran data antar proses dan *entity* terkait. Model proses tersebut direpresentasikan dalam bentuk *Physical Data Flow Diagram* yang dapat dilihat pada Gambar 5. Pada *Physical Data Flow Diagram* tersebut menunjukkan bahwa pengguna (*external entity*) melakukan input data berupa titik awal dan titik akhir melalui formulir pada bagian *front-end*. Selanjutnya, terdapat proses (1) mengirimkan input data, yang kemudian *back-end* (2) menerima input data untuk dijadikan parameter dalam (3) mencari rute terpendek menggunakan algoritma Dijkstra. Pada proses pencarian rute terpendek, diperlukan data *graph* pemandangan yang dimuat dalam format JSON. Hasil pencarian rute terpendek berupa daftar titik rute terpendek, yang kemudian dilanjutkan dengan proses (4) mengirimkan hasil tersebut sebagai *response. Front-end* akan menerima (5) hasil pencarian rute terpendek, kemudian dikemas menjadi parameter untuk (6) menampilkan visualisasi rute terpendek pada peta, dan (7) menampilkan daftar pemandangan berdasarkan rute terpendek kepada pengguna.



Gambar 5. Aliran Data pada Fitur Navigasi Pencarian Rute Terpendek

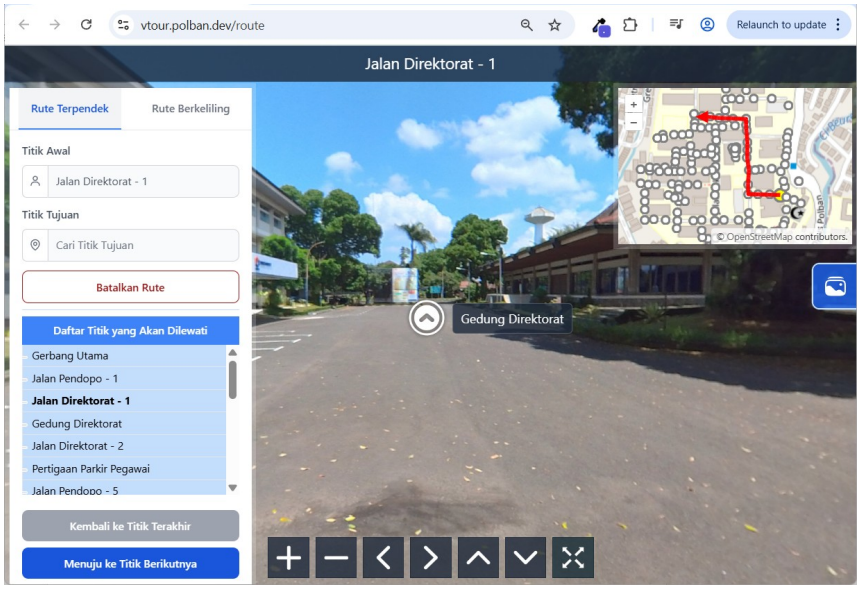
2.4. Pengembangan Fitur Navigasi

Fitur navigasi pencarian rute yang diterapkan sesuai dengan rancangan pada tahap perancangan. Pada Gambar 6 menunjukkan antarmuka pengguna fitur navigasi sebelum melakukan pencarian rute, sedangkan pada Gambar 7 menunjukkan setelah melakukan pencarian rute. Tampilan antarmuka pengguna yang telah menghasilkan rute akan menayangkan daftar titik yang akan dilewati. Daftar titik tersebut dapat dijelajahi secara bertahap dengan menekan tombol “menuju ke titik berikutnya”.



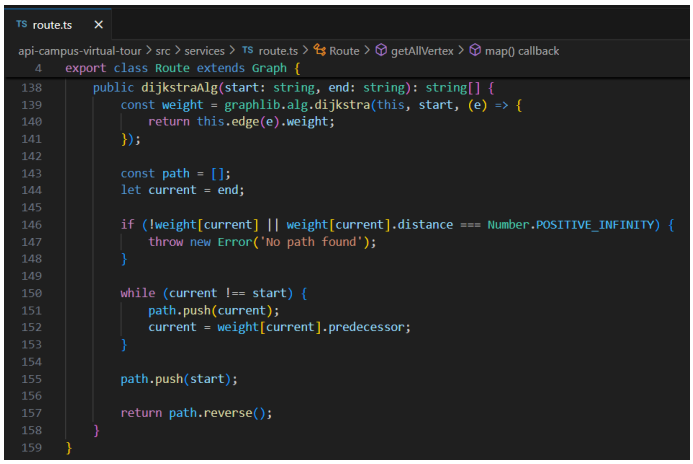
Gambar 6. Antarmuka Pengguna pada Fitur Navigasi Rute Terpendek – Sebelum Melakukan Pencarian

Pada bagian kanan atas Gambar 7, terdapat area peta yang memvisualisasikan hasil pencarian rute dengan garis merah pada setiap titik yang akan dilewati. Garis merah tersebut disertai juga anak panah untuk menunjukkan arah yang akan dilalui. Setiap perpindahan titik, tampilan pemandangan dan visualisasi peta akan menyesuaikan dengan perubahan secara dinamis.



Gambar 7. Antarmuka Pengguna pada Fitur Navigasi Rute Terpendek – Setelah Melakukan Pencarian

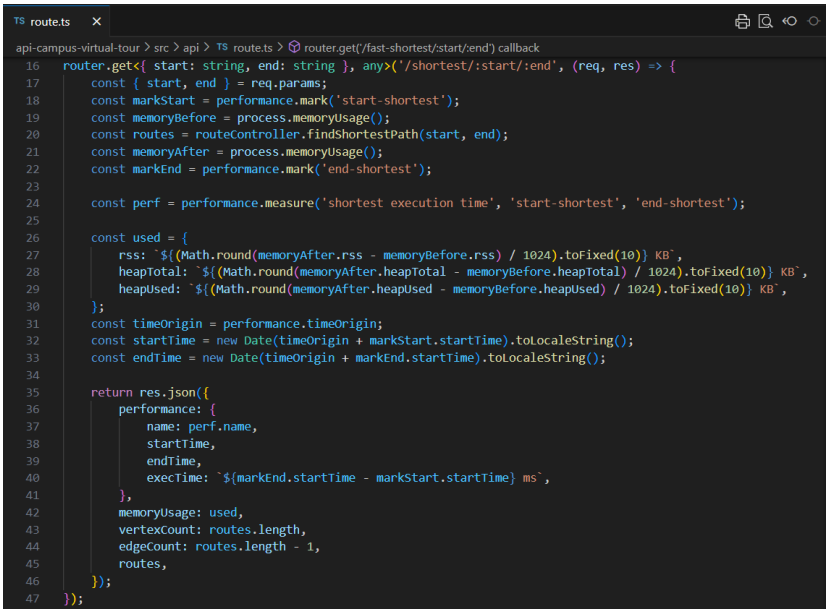
Pada Gambar 8 menunjukkan potongan kode sumber (*code snippet*) berupa *function* `dijkstraAlg` yang menerima parameter input *start* dan *end*, sesuai dengan input pada form navigasi pencarian rute. *Function* `dijkstraAlg` di dalamnya memanggil *function* algoritma Dijkstra yang dapat langsung diterapkan, diperoleh dari pustaka `@dagrejs/graphlib`.



Gambar 8. Potongan Kode Sumber Penerapan Algoritma Dijkstra – Graphlib pada Komponen Rute

2.5. Uji Coba

Uji coba yang dilakukan yaitu mengukur performa dan penggunaan memori dengan menggunakan *performance and measurement API* (`perf_hooks`) pada Node.js [23]. Uji coba mencakup lima skenario yang dijalankan, yang masing-masing skenario memiliki titik awal dan titik akhir yang berbeda. Serta, untuk memperoleh hasil yang konsisten, setiap skenario dilakukan sebanyak tiga iterasi untuk diambil nilai rata-ratanya. Pada Gambar 8 menunjukkan potongan kode sumber untuk mengukur proses pencarian rute terpendek dengan ruang lingkup pengujiannya mulai dari *request* yang dimasukkan ke API sampai dengan hasil yang dikeluarkan dari API. Variabel yang diukur yaitu: (1) waktu eksekusi; (2) *resident set size* (*RSS*) yang merupakan bagian yang ditempati oleh suatu proses dalam memori utama; (3) *total heap* merupakan ukuran seluruh *heap* yang baru dialokasikan ketika akan adanya objek baru; dan (4) *used heap* adalah ukuran *heap* yang aktual digunakan.



Gambar 9. Potongan Kode Sumber Pengukuran Performa Proses Pencarian Rute Terpendek

3. HASIL DAN PEMBAHASAN

Fitur navigasi *shortest path* pada *360° virtual campus tour* POLBAN telah berhasil dikembangkan sesuai dengan rancangan dan dapat diakses pada tautan https://vtour.polban.dev/route. Penerapan algoritma *shortest path* Dijkstra pada fitur tersebut didukung oleh pustaka Graphlib. Fungsionalitas utama fitur tidak menemui kendala dan sudah memberikan hasil yang sesuai seperti pada Gambar 7. Fitur tersebut dilengkapi juga dengan uji coba dengan pengukuran performa dan penggunaan memori seperti yang telah dijelaskan

pada Bagian 2.5. Detail skenario uji coba yang telah diurutkan berdasarkan kompleksitasnya dari yang paling rendah ke tinggi, dapat dilihat pada Tabel 2.

Tabel 2. Skenario Uji coba

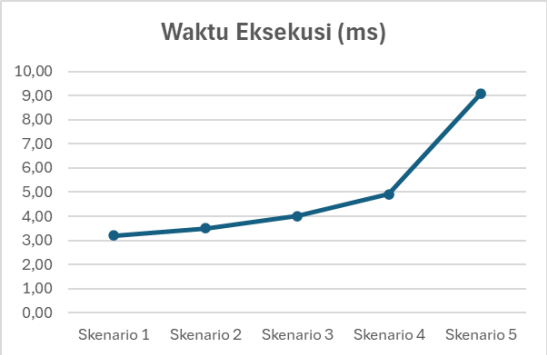
No. Skenario	Titik Awal	Titik Akhir	Jumlah *Vertex*	Jumlah *Edge*
1	Gerbang Utama	Laboratorium Artificial Intelligence	15	14
2	Gerbang Utama	Gedung Teknik Komputer dan Informatika	20	19
3	Gerbang Utara	Masjid Lukmannul Hakim	22	21
4	Gerbang Utara	Laboratorium Aeronautika - 1	23	22
5	Gerbang Utara	Laboratorium Fabrikasi (FabLab)	29	28

Hasil uji coba pada Tabel 3, menunjukkan bahwa semakin kompleks skenario berdasarkan indikator jumlah *vertex* dan *edge*, semakin tinggi waktu eksekusi yang dihasilkan. Mengenai pengukuran *total heap* pada seluruh skenario sama sekali tidak menghasilkan nilai karena tidak ada pembentukkan objek baru. Terakhir, nilai yang diperoleh untuk *used heap* cukup terlihat memiliki pola berdasarkan kompleksitas.

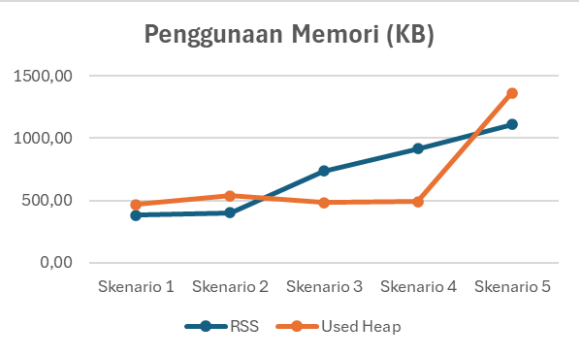
Tabel 3. Hasil Uji coba

No. Skenario	Iterasi	Performa (ms)	Penggunaan Memori (KB)		
		Waktu Eksekusi	RSS	*Total Heap*	*Used Heap*
1	1	3,00	496,00	0,00	472,17
	2	1,99	264,00	0,00	456,27
	3	4,62	392,00	0,00	476,42
	Rata-rata	3,20	384,00	0,00	468,28
2	1	5,60	656,00	0,00	609,35
	2	2,17	528,00	0,00	494,10
	3	2,74	24,00	0,00	511,46
	Rata-rata	3,50	403,00	0,00	538,30
3	1	2,17	528,00	0,00	486,58
	2	6,64	1268,00	0,00	487,58
	3	3,19	416,00	0,00	471,63
	Rata-rata	4,00	737,33	0,00	481,93
4	1	2,30	352,00	0,00	473,29
	2	3,14	860,00	0,00	501,25
	3	9,33	1540,00	0,00	493,44
	Rata-rata	4,92	917,33	0,00	489,32
5	1	9,79	928,00	0,00	1280,00
	2	8,68	1300,00	0,00	1280,00
	3	8,71	1104,00	0,00	1536,45
	Rata-rata	9,06	1110,67	0,00	1365,22
	Rata-rata Keseluruhan	4,94	710,47	0,00	668,61

Berdasarkan nilai-nilai hasil uji coba tersebut, secara umum dengan merujuk kepada nilai rata-rata keseluruhan, waktu eksekusi tergolong sangat cepat (4,94 ms), sedangkan RSS (710,47 KB) dan *used heap* (668,61 KB) tergolong hemat konsumsi memori. Namun, studi ini masih belum dapat memberikan argumentasi terhadap perolehan nilai *used heap* skenario nomor 2 yang lebih tinggi dibandingkan nomor 4 dan nomor 5 yang lebih kompleks, sehingga memerlukan investigasi lebih mendalam. Asumsi sementara disebabkan karena terdapat proses lain yang menggunakan *resource* yang sama dalam waktu singkat saat melakukan uji coba. Untuk memudahkan dalam mengetahui tren performa rata-rata setiap skenario dapat dilihat pada Gambar 10 dan Gambar 11.



Gambar 10. Waktu Eksekusi Setiap Skenario



Gambar 11. Penggunaan Memori Setiap Skenario

4. KESIMPULAN

Studi ini berhasil menerapkan fitur navigasi rute pada *360° virtual campus tour* yang dikembangkan oleh POLBAN untuk menentukkan rute terpendek menggunakan algoritma Dijkstra. Penggunaan algoritma Dijkstra didukung oleh pustaka Graphlib, sehingga dalam proses penerapannya relatif lebih mudah dengan adanya *function* yang telah didefinisikan. Fungsionalitas utama fitur tidak menemui kendala, sehingga sudah dapat digunakan pada lingkungan *production* untuk membantu pengguna menjelajahi pemandangan yang akan dituju berdasarkan rute yang terpendek yang dihasilkan. Dalam hal uji coba performa dan konsumsi memori, fitur navigasi rute ini tergolong sangat cepat waktu eksekusinya (4,94 ms) dan hemat konsumsi memori yang diukur berdasarkan RSS (710,47 KB) dan *used heap* (668,61 KB). Kedepannya, terdapat peluang pengembangan yang dapat difokuskan pada pengukuran pengalaman pengguna (*user experience*) serta pemanfaatan dengan pendekatan *pre-compute*, yang memungkinkan seluruh rute *shortest-path* dapat dihasilkan terlebih dahulu sebelum aksi pencarian untuk lebih mempercepat proses pencarian rute. Selain itu, saat ini masih terdapat keterbatasan pada tampilan antarmuka pengguna yang hanya nyaman diakses melalui perangkat Desktop, sehingga adaptasi terhadap berbagai jenis perangkat, khususnya *mobile*, menjadi tantangan dan peluang pengembangan selanjutnya.

UCAPAN TERIMA KASIH

Studi ini didanai oleh hibah kompetitif POLBAN melalui skema Penelitian Madya Utama (PMU), dengan nomor B/2.2/PL1.R7/PG.00.03/2024. Penulis mengucapkan terima kasih kepada Faizal Abdul Hakim dan Temmy Mahesa Ridwan yang turut berkontribusi dalam mengembangkan *360° virtual campus tour* di lingkungan POLBAN.

REFERENSI

[1] R. B. Rohizan, D. M. Vistro, and M. R. Bin Puasa, "Enhanced Visitor Experience Through Campus Virtual Tour," J Phys Conf Ser, vol. 1228, no. 1, p. 12067, May 2019, doi: 10.1088/1742-6596/1228/1/012067.

[2] R. B. Figueroa, G. A. G. Mendoza, J. C. C. Fajardo, S. E. Tan, E. Yassin, and T. H. Thian, "Virtualizing a university campus tour: A pilot study on its usability and user experience, and perception," *International Journal in Information Technology in Governance, Education and Business*, vol. 2, no. 1, pp. 1–8, Nov. 2020.

[3] M. I. Iglesias, M. Jenkins, and G. Morison, "Enhanced Low-cost Web-based Virtual Tour Experience for Prospective Students," in *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, Mar. 2021, pp. 677–678. doi: 10.1109/VRW52623.2021.00221.

[4] A. D. Samala, F. Ranuharja, B. R. Fajri, Y. Indarta, and W. Agustiarmi, "ViCT—Virtual Campus Tour Environment with Spherical Panorama: A Preliminary Exploration," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 16, no. 16, pp. 205–225, Aug. 2022, doi: 10.3991/ijim.v16i16.32889.

[5] A. and K. J.-H. Hendricks Stefan and Shaker, "Design of a VR-Based Campus Tour Platform with a User-Friendly Scene Asset Management System," in *Intelligent Human Computer Interaction*, M. and K. J. and T. U. S. and S. M. and S. D. Kim Jong-Hoon and Singh, Ed., Cham: Springer International Publishing, 2022, pp. 337–348.

[6] A. S. binti Azizo, F. bin Mohamed, C. V. Siang, and M. I. M. Isham, "Virtual Reality 360 UTM Campus Tour with Voice Commands," in *2020 6th International Conference on Interactive Digital Media (ICIDM)*, Dec. 2020, pp. 1–6. doi: 10.1109/ICIDM51048.2020.9339665.

[7] B. Maulik, A. P. Nayak, S. U, S. Alok, and D. K. N, "Design and Implementation of Virtual Tour Guide App," in *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, 2022, pp. 1–6. doi: 10.1109/ICACTA54488.2022.9752804.

[8] E. Paskahlia Gunawan and C. Tho, "Development of an application for tourism route recommendations with the Dijkstra algorithm," in *2021 International Conference on Information Management and Technology (ICIMTech)*, IEEE, Aug. 2021.

[9] Risald, A. E. Mirino, and Suyoto, "Best routes selection using Dijkstra and Floyd-Warshall algorithm," in *2017 11th International Conference on Information Communication Technology and System (ICTS)*, IEEE, Oct. 2017.

[10] A. Candra, M. A. Budiman, and K. Hartanto, "Dijkstra's and A-Star in Finding the Shortest Path: a Tutorial," *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, pp. 28–32, 2020, doi: 10.1109/DATABIA50434.2020.9190342.
[11] W. Feng, "Improvement of Dijkstra in Embedded-GIS route analysis," *Computer Engineering and Applications*, 2008.
[12] M. Noto and H. Sato, "A method for the shortest path search by extended Dijkstra algorithm," *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. "cybernetics evolving to systems, humans, organizations, and their complex interactions" (cat. no.0*, vol. 3, pp. 2316–2320 vol.3, 2000, doi: 10.1109/ICSMC.2000.886462.
[13] N. Biswas, D. Banerjee, and S. Bhattacharya, "Minimising the duration of a system-controlled virtual reality tour," in *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, IEEE, Oct. 2022. doi: 10.1109/ismar-adjunct57072.2022.00124.
[14] Marzipano, "Marzipano is a 360° media viewer for the modern web," Marzipano Docs. Accessed: Apr. 30, 2025. [Online]. Available: https://www.marzipano.net/docs.html
[15] Node.js, "About Node.js®," Node.js. Accessed: Apr. 30, 2025. [Online]. Available: https://nodejs.org/en/about
[16] OpenJS Foundation, "Node.js web application framework," ExpressJS. Accessed: Apr. 30, 2025. [Online]. Available: https://expressjs.com/
[17] Evan You, "Vue.js: The Progressive JavaScript Framework," VueJS. Accessed: Apr. 30, 2025. [Online]. Available: https://vuejs.org/
[18] Tailwind Labs Inc, "Tailwind CSS: Rapidly build modern websites without ever leaving your HTML," TailwindCSS. Accessed: Apr. 30, 2025. [Online]. Available: https://tailwindcss.com/
[19] DagreJS, "A directed multi-graph library for JavaScript," GitHub Wiki. Accessed: Apr. 30, 2025. [Online]. Available: https://github.com/dagrejs/graphlib/wiki
[20] OpenLayers, "OpenLayers: A high-performance, feature-packed library for all your mapping needs," OpenLayers. Accessed: Apr. 30, 2025. [Online]. Available: https://openlayers.org/
[21] C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3rd Edition. Prentice Hall PTR, 2005.
[22] J. L. Whitten and L. D. Bentley, *Systems Analysis and Design Methods*, 7th Edition. McGraw-Hill Irwin, 2007.
[23] Node.js, "Performance measurement APIs," Node.js Documentation. Accessed: Apr. 30, 2025. [Online]. Available: https://nodejs.org/api/perf_hooks.html#performance-measurement-apis