



Analisis Performa Algoritma Dijkstra untuk Penentuan Jalur Terpendek pada Aplikasi Mobile Berbasis Location-Based Service

Ennurul Hasanah^{1*}, Rizkayeni Marta², Dedy Irfan³, Yulia Fatmi⁴

¹⁻⁴ Pendidikan Teknik Informatika, Departemen Teknik Elektronika, Fakultas Teknik
Universitas Negeri Padang

*Penulis Korespondensi: ennurulhasanah1635@gmail.com

Abstract. *The development of Location- Based Service (Pound) technology has increased the demand for systems capable of determining travel routes quickly and accurately in mobile applications. One of the most widely used approaches to the shortest path problem is the Dijkstra technique. This study aims to assess how well the Dijkstra algorithm finds the shortest path in mobile applications that use location-based services. The research method includes literature review, algorithm design, implementation in a mobile application, and testing using several route-search scenarios. The analysis was conducted based on the algorithm's ability to generate the shortest route, computational time, and consistency of the search results. The findings show that the Dijkstra method can precisely identify the ideal path on road networks with positive edge weights and process quickly enough for mobile app navigation. Therefore, for Location-Based Service-based mobile apps that provide route determination, the Dijkstra algorithm can be a helpful solution.*

Keywords: *Dijkstra Algorithm, Shortest Path, Location-Based Service, Mobile Application, Route Optimization*

Abstrak. Perkembangan teknologi *Location-Based Service* (LBS) telah mendorong meningkatnya kebutuhan akan sistem yang mampu menentukan jalur perjalanan secara cepat dan akurat pada aplikasi mobile. Salah satu tata cara yang banyak digunakan buat menuntaskan kasus pencarian jalan terpendek merupakan algoritma Dijkstra. Riset ini bertujuan buat menganalisis kinerja algoritma Dijkstra dalam memastikan jalan terpendek pada aplikasi mobile berbasis *Location- Based Service*. Tata cara riset yang digunakan meliputi riset literatur, perancangan algoritma, implementasi pada aplikasi *mobile*, serta pengujian terhadap beberapa skenario pencarian rute. Analisis dilakukan berdasarkan kemampuan algoritma dalam menghasilkan jalur terpendek, waktu komputasi, dan konsistensi hasil pencarian. Hasil riset menampilkan kalau algoritma Dijkstra sanggup memastikan rute maksimal secara akurat pada jaringan jalur yang mempunyai bobot positif dan membagikan waktu pemrosesan yang relatif kilat buat kebutuhan navigasi pada aplikasi mobile. Dengan demikian, algoritma Dijkstra bisa jadi pemecahan yang efisien dalam menunjang layanan penentuan jalan pada aplikasi berbasis *Location- Based Service*.

Kata kunci: Algoritma Dijkstra, Jalur Terpendek, *Location-Based Service*, Aplikasi *Mobile*, Optimasi Rute

1. LATAR BELAKANG

Perkembangan teknologi informasi telah mendorong pemanfaatan aplikasi mobile dalam berbagai bidang, termasuk layanan transportasi dan navigasi. *Location-Based Service* (LBS), teknologi yang banyak digunakan, memungkinkan pengguna mendapatkan informasi berdasarkan lokasi secara real time. Penerapan teknologi tersebut memberikan kemudahan bagi pengguna dalam menentukan rute perjalanan secara lebih

cepat, efektif, dan efisien sehingga mampu meningkatkan kualitas layanan transportasi berbasis mobile.

Dalam aplikasi navigasi, penentuan jalur terpendek merupakan salah satu aspek penting untuk membantu pengguna mencapai tujuan melalui rute yang optimal. Ketepatan dalam menentukan jalur berpengaruh terhadap waktu tempuh, efisiensi perjalanan, serta kualitas pelayanan yang diberikan kepada pengguna. Oleh karena itu, diperlukan metode yang dapat menghitung jalur terpendek dengan akurat berdasarkan data lokasi yang tersedia.

Algoritma ini ialah algoritma pencarian jalur terpendek dan banyak dipakai karena mampu menemukan rute optimal pada graf berbobot positif. Algoritma ini bekerja dengan membandingkan bobot setiap lintasan hingga diperoleh jalur dengan total jarak paling kecil dari titik awal menuju titik tujuan. Dengan kemampuan tersebut, algoritma Dijkstra dapat cantumkan di aplikasi mobile berbasis Location-Based Service untuk membantu proses penentuan rute perjalanan secara efektif.

Penelitian ini menerapkan algoritma Dijkstra pada pengguna mobile berbasis Location-Based Service dengan menentukan jalur terpendek menuju lokasi tujuan. Analisis dilakukan untuk mengetahui kemampuan algoritma dalam menghasilkan rute yang optimal serta mendukung proses navigasi dengan cepat hingga akurat sesuai kebutuhan pengguna.

Penelitian ini menganalisis kinerja algoritma Dijkstra untuk menemukan jalur terpendek pada aplikasi mobile berbasis lokasi berbasis layanan (LBS). Analisis dilakukan melalui implementasi algoritma pada aplikasi mobile serta pengujian terhadap beberapa skenario pencarian rute untuk mengevaluasi kemampuan algoritma dalam menghasilkan jalur optimal, waktu komputasi, dan konsistensi hasil pencarian. Akibatnya, hasil penelitian diharapkan dapat memberikan gambaran tentang efektivitas algoritma Dijkstra sebagai cara untuk menemukan jalur terpendek. Hasil ini juga dapat menjadi referensi untuk pengembangan aplikasi navigasi berbasis ponsel yang dapat memberikan informasi rute secara cepat, akurat, dan efisien.

2. METODE PENELITIAN

Metode penelitian dan pengembangan (R&D) dan pendekatan pengembangan aplikasi mobile berbasis lokasi berbasis layanan (LBS) digunakan dalam penelitian ini. Algoritma Dijkstra digunakan untuk menemukan jalur terpendek. Identifikasi kebutuhan, perancangan algoritma, implementasi aplikasi, pengujian algoritma, dan evaluasi hasil pengujian adalah semua tahapan penelitian.

Tahap identifikasi kebutuhan dilakukan untuk mengetahui kebutuhan sistem dalam menentukan jalur terpendek berdasarkan lokasi pengguna dan lokasi tujuan. Data yang digunakan berupa koordinat lokasi, jaringan jalan, serta bobot jarak antar titik yang direpresentasikan dalam bentuk graf.

Tahap perancangan dilakukan dengan menyusun struktur graf, menentukan hubungan antar simpul (node), serta merancang alur kerja algoritma Dijkstra dalam mencari lintasan dengan total bobot terkecil. Selanjutnya dilakukan implementasi algoritma pada aplikasi mobile berbasis Flutter yang terintegrasi dengan layanan peta digital sehingga mampu menampilkan hasil pencarian jalur secara visual.

Pada tahap pengujian, berbagai skenario pencarian rute digunakan untuk mencapai tujuan dan lokasi awal yang berbeda. Tujuan dari pengujian ini adalah untuk mengevaluasi kemampuan algoritma Dijkstra untuk menghasilkan jalur terpendek, waktu pemrosesan yang paling singkat, dan konsistensi hasil pencarian. Selain itu, aplikasi diuji melalui Test User Acceptance (UAT) dan System Usability Scale (SUS) untuk memastikan bahwa semua fungsi memenuhi persyaratan pengguna. Analisis kebutuhan, perancangan sistem, penerapan, pengujian, dan evaluasi adalah tahapan penelitian.

1. Analisis Kebutuhan

Pada titik ini, literatur penelitian dan analisis proses navigasi digunakan untuk menentukan kebutuhan sistem. Data yang digunakan termasuk koordinat lokasi pengguna, koordinat tujuan, jaringan jalan, dan bobot jarak antar simpul dalam grafik berbobot. Perancangan Sistem

2. Perancangan sistem meliputi pembuatan model graf, perancangan alur algoritma Dijkstra, desain antarmuka aplikasi, serta perancangan basis data yang digunakan untuk menyimpan informasi lokasi dan jaringan jalan. Aplikasi dikembangkan

menggunakan framework Flutter dengan integrasi Google Maps API sebagai media visualisasi peta.

3. Implementasi Algoritma Dijkstra

Proses pencarian jalur dimulai dengan simpul awal, atau sumber node, yang diberi nilai jarak sebesar 0, sedangkan semua simpul lainnya diberi nilai tak hingga (∞), menandakan bahwa jarak menuju simpul tersebut belum diketahui. Selanjutnya, algoritma memilih simpul dengan nilai jarak terkecil dan belum pernah dikunjungi untuk diproses. Dari simpul tersebut, algoritma menghitung jarak ke seluruh simpul tetangganya dengan membandingkan jarak yang telah tersimpan sebelumnya dengan hasil penjumlahan jarak dari simpul saat ini dan bobot sisi yang menghubungkan kedua simpul. Proses pembaruan nilai jarak dilakukan menggunakan:

$$D(v) = \min(D(v), D(u) + w(u, v))$$

di mana $D(v)$ merupakan jarak minimum menuju simpul v , $D(u)$ adalah jarak menuju simpul u , dan $w(u, v)$ merupakan bobot sisi yang menghubungkan simpul u dengan simpul v . Tahapan ini dilakukan secara berulang hingga seluruh simpul telah diproses atau simpul tujuan berhasil ditemukan, sehingga diperoleh jalur dengan total bobot atau jarak paling kecil. Hasil akhir berupa jalur dengan total bobot terkecil yang kemudian divisualisasikan pada peta digital beserta informasi jarak dan estimasi waktu tempuh.

4. Pengujian Sistem

Beberapa skenario pencarian rute, masing-masing dengan lokasi awal dan tujuan yang berbeda, digunakan untuk menguji sistem. Evaluasi dilakukan untuk mengetahui kinerja algoritma Dijkstra berdasarkan beberapa parameter, yaitu akurasi jalur yang dihasilkan, total jarak tempuh, waktu komputasi algoritma, serta konsistensi hasil pencarian pada setiap skenario pengujian.

5. Teknik Analisis Data

Data hasil pengujian dianalisis secara deskriptif dengan membandingkan hasil pencarian rute pada setiap skenario. Efektivitas algoritma dinilai berdasarkan kemampuan menghasilkan jalur terpendek secara konsisten, kecepatan proses

komputasi, serta hasil evaluasi usability melalui skor SUS. Hasil analisis digunakan untuk menilai kelayakan penerapan algoritma Dijkstra pada aplikasi mobile berbasis LBS.

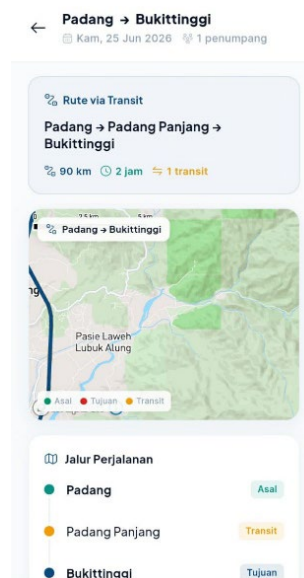
3. HASIL DAN PEMBAHASAN

Tujuan penelitian ini adalah mempelajari algoritma Dijkstra dalam menemukan jalur terpendek untuk aplikasi mobile berbasis lokasi berbasis layanan (LBS). Implementasi algoritma dilakukan dengan memanfaatkan data lokasi pengguna dan lokasi tujuan yang kemudian diproses untuk memperoleh rute dengan total bobot terkecil. Hasil implementasi selanjutnya dianalisis berdasarkan proses pencarian rute, hasil jalur yang diperoleh, serta pengujian terhadap beberapa skenario.

A. Implementasi Sistem

1. Menjalankan Penetapan Lokasi Awal dan Tujuan

Proses menentukan lokasi awal dan tujuan, yang akan digunakan sebagai dasar pencarian jalur, dilakukan sebelum implementasi algoritma Dijkstra. Lokasi awal diperoleh melalui teknologi Location-Based Service (LBS) yang memanfaatkan koordinat geografis perangkat pengguna, sedangkan lokasi tujuan dipilih secara langsung melalui tampilan peta digital. Informasi koordinat tersebut menjadi data masukan yang akan diproses oleh sistem.

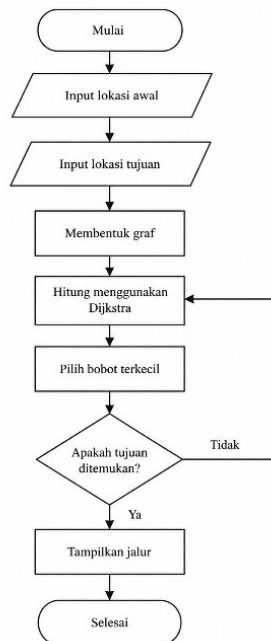


Gambar 1. Penetapan Lokasi Awal dan Lokasi Tujuan

Proses perhitungan algoritma Dijkstra memasukkan data lokasi awal dan tujuan. Sistem akan menggunakan data tersebut untuk memulai pencarian jalur dengan mempertimbangkan bobot setiap lintasan yang tersedia. Proses ini akan berlanjut hingga menemukan rute dengan bobot paling kecil secara keseluruhan.

1. Proses Perhitungan Algoritma Dijkstra

Setelah memperoleh data lokasi, algoritma Dijkstra mulai melakukan proses pencarian jalur terpendek. Algoritma menghitung bobot dari setiap lintasan yang dapat dilalui, kemudian pada setiap iterasi memilih simpul dengan nilai bobot terkecil hingga mencapai lokasi tujuan.

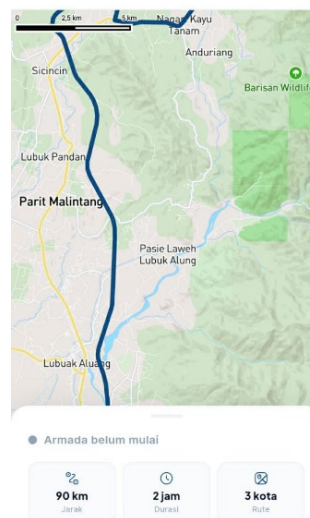


Gambar 2. Proses Perhitungan Algoritma Dijkstra

Proses tersebut berlangsung secara berulang hingga seluruh kemungkinan jalur yang relevan telah diperiksa atau lokasi tujuan berhasil ditemukan. Pada setiap iterasi, algoritma selalu memilih lintasan yang memiliki nilai bobot paling kecil sehingga hasil yang diperoleh merupakan jalur dengan total jarak minimum. Daripada memilih jalur secara manual, mekanisme ini memungkinkan sistem membuat rute perjalanan yang lebih optimal dan membantu pengguna menemukan jalur yang lebih efisien.

2. Implementasi Hasil Jalur Terpendek

Setelah perhitungan selesai, sistem menampilkan hasil pencarian jalur pada peta digital. Jalur yang ditampilkan merupakan hasil akhir dari proses perhitungan algoritma Dijkstra yang telah memilih lintasan dengan total bobot paling kecil. Karena jalur ditampilkan secara jelas dari lokasi awal hingga lokasi tujuan, visualisasi tersebut memudahkan pengguna untuk memahami arah perjalanan.



Gambar 3. Implementasi Jalur Terpendek

Selain menampilkan rute perjalanan, sistem juga memberikan informasi tambahan berupa total jarak tempuh dan estimasi waktu perjalanan. Saat merencanakan perjalanan, orang dapat menggunakan informasi ini untuk menghemat waktu tempuh. Hasil implementasi menunjukkan bahwa algoritma Dijkstra dapat membuat rute terbaik dari data graf yang digunakan sistem.

B. Hasil Uji Sistem

1. Analisis Hasil Pengujian Algoritma

Pengujian dilakukan menggunakan beberapa skenario pencarian jalur dengan kombinasi lokasi awal dan lokasi tujuan yang berbeda. Setiap skenario memiliki tujuan untuk menentukan kemampuan algoritma Dijkstra untuk menemukan lintasan dengan bobot total paling kecil berdasarkan jaringan jalan yang tersedia. Hasil dari setiap pengujian kemudian dibandingkan untuk melihat konsistensi algoritma dalam menghasilkan jalur terpendek.

Setiap pengujian membandingkan alternatif rute yang tersedia dengan rute yang dipilih oleh algoritma. Parameter yang diamati meliputi jalur yang dihasilkan, total jarak tempuh, estimasi durasi perjalanan, serta keberhasilan algoritma dalam menentukan rute yang paling optimal.

Tabel 1. Hasil Pengujian Algoritma Dijkstra

No	Lokasi Awal	Lokasi Tujuan	Alternatif Rute (Lintas Jalur)	Rute Terpilih (Dijkstra)	Jarak (km)	Durasi (estimasi)	Hasil
1	Padang	Bukittinggi	1) Padang – Sicincin – Bukittinggi (95 km) 2) Padang – Padang Panjang – Bukittinggi (90 km) 3) Padang – Solok – Bukittinggi (120 km)	Padang – Padang Panjang – Lubuk Alung – Bukittinggi	90	2 jam	Berhasil
2	Padang	Padang Panjang	1) Padang – Sicincin – Padang Panjang (62 km) 2) Padang – Lubuk Alung – Padang Panjang (58 km)	Padang – Lubuk Alung – Padang Panjang	58	1 jam 20 menit	Berhasil
3	Bukittinggi	Payakumbuh	1) Bukittinggi – Baso – Payakumbuh (65 km) 2) Bukittinggi – Padang Panjang – Payakumbuh (70 km)	Bukittinggi – Baso – Payakumbuh	62	1 jam 30 menit	Berhasil
4	Padang	Solok	1) Padang – Indarung – Solok (78 km) 2) Padang – Sicincin – Solok (82 km)	Padang – Indarung – Solok	75	1 jam 45 menit	Berhasil
5	Pariaman	Bukittinggi	1) Pariaman – Sicincin – Padang Panjang – Bukittinggi (112 km) 2) Pariaman – Padang – Bukittinggi (130 km)	Pariaman – Sicincin – Padang Panjang – Bukittinggi	110	2 jam 35 menit	Berhasil

Tabel 1. Analisis Hasil Pengujian Algoritma Dijkstra

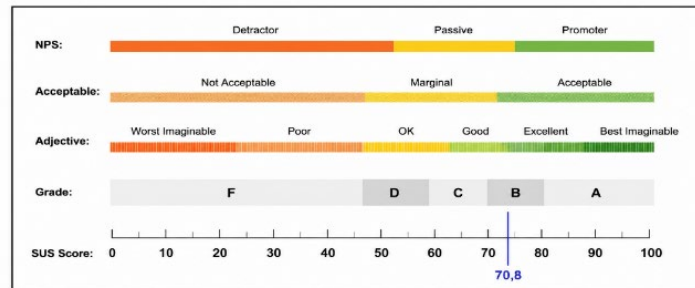
Berdasarkan hasil tes yang ditunjukkan dalam Tabel 1, algoritma Dijkstra berhasil menentukan jalur dengan total bobot paling kecil pada setiap skenario yang diuji. Rute yang dipilih memiliki jarak yang lebih pendek dibandingkan alternatif rute lainnya sehingga mampu memberikan perjalanan yang lebih efisien. Selain menghasilkan jalur yang optimal, algoritma juga memberikan estimasi durasi perjalanan yang dapat dijadikan sebagai informasi pendukung bagi pengguna dalam menentukan rute.

Hasil menunjukkan bahwa algoritma Dijkstra dapat mengatasi masalah pencarian jalur terpendek pada aplikasi mobile berbasis lokasi berbasis layanan (LBS).

2. Hasil System Usability Scale (SUS)

Sistem kemudahan penggunaan skala (SUS) diuji setelah algoritma Dijkstra diterapkan. Perhitungan skor SUS dilakukan dengan mengolah jawaban responden sesuai aturan perhitungan yang telah ditetapkan. Nilai akhir berada pada rentang 0 hingga 100, kemudian diinterpretasikan ke dalam beberapa kategori, seperti Acceptability Range, Grade Scale, dan Adjective Rating. Berdasarkan hasil penilaian

pengguna, interpretasi ini digunakan untuk menentukan apakah suatu sistem memiliki tingkat usability yang buruk, cukup, baik, atau sangat baik.



Gambar 4. Hasil System Usability Scale (SUS)

Seperti yang ditunjukkan pada Gambar 4, hasil pengujian menunjukkan skor SUS sebesar 70,8, yang berada dalam kategori Acceptable dan mendapatkan Grade B. Nilai tersebut juga berada dalam Rating Adjective "Baik". Hasil ini menunjukkan bahwa aplikasi itu mudah digunakan, yang memungkinkan pengguna menggunakan fitur penentuan jalur dengan mudah dan nyaman. Hasil ini menunjukkan bahwa algoritma Dijkstra dapat diterapkan secara optimal untuk jalur terpendek dan aplikasi ponsel dengan antarmuka pengguna yang mudah digunakan. Akibatnya, aplikasi ini dianggap cocok untuk digunakan sebagai alat penentuan jalur berbasis lokasi berbasis layanan (LBS).

4. KESIMPULAN DAN SARAN

Kesimpulan

Hasil penelitian menunjukkan bahwa algoritma Dijkstra dapat diterapkan dengan baik pada aplikasi mobile berbasis layanan lokasi (LBS) untuk menemukan jalur terpendek dari lokasi awal ke lokasi tujuan. Proses pencarian jalur dilakukan dengan merepresentasikan jaringan jalan ke dalam bentuk graf berbobot, kemudian algoritma menghitung total bobot setiap lintasan hingga diperoleh rute dengan jarak paling kecil. Hasil implementasi menunjukkan bahwa algoritma Dijkstra dapat secara konsisten menghasilkan rute terbaik pada setiap skenario pengujian, sehingga dapat membantu pengguna menemukan rute perjalanan yang lebih efisien.

Berdasarkan hasil analisis pengujian, algoritma ini berhasil menemukan rute terpendek dengan membandingkan berbagai rute alternatif yang tersedia. Jalur yang

dipilih memiliki bobot total yang lebih rendah daripada jalur lainnya untuk memberikan efisiensi terhadap jarak tempuh dan estimasi waktu perjalanan. Ini menunjukkan bahwa algoritma Dijkstra berhasil menyelesaikan masalah pencarian jalur terpendek pada jaringan jalan dengan bobot positif dengan tingkat akurasi yang tinggi.

Penelitian ini juga menguji kemudahan penggunaan aplikasi melalui metode Sistem Usability Scale (SUS). Hasil pengujian menunjukkan bahwa aplikasi tidak hanya mampu menjalankan algoritma Dijkstra dengan baik, tetapi juga memiliki skor 70,8, termasuk dalam kategori Acceptable, memperoleh Grade B, dan memiliki rating kata-kata "Baik".

Secara keseluruhan, penelitian ini menunjukkan bahwa algoritma Dijkstra dapat digunakan pada aplikasi mobile berbasis lokasi berbasis layanan (LBS) untuk menemukan jalur terpendek dengan cepat. Selain itu, aplikasi yang dikembangkan sangat mudah digunakan, sehingga dapat memenuhi kebutuhan navigasi dan menemukan rute perjalanan secara lebih efektif.

Saran

Hasil penelitian menunjukkan bahwa pengembangan lebih lanjut untuk mengetahui seberapa jauh algoritma Dijkstra berkembang dan algoritma pencarian jalur lainnya berbeda dalam hal akurasi dan efisiensi dalam menemukan jalur terpendek. Rute yang dibangun juga dapat disesuaikan dengan kondisi jalan yang berubah dengan menggunakan data lalu lintas real-time.

Pengujian juga dapat dilakukan pada wilayah yang lebih luas serta menggunakan lebih banyak skenario pencarian rute sehingga hasil analisis menjadi lebih komprehensif. Dengan pengembangan tersebut, diharapkan penelitian selanjutnya dapat menghasilkan sistem penentuan jalur yang lebih optimal, akurat, dan mampu memberikan layanan navigasi yang lebih baik bagi pengguna aplikasi mobile berbasis Location-Based Service (LBS).

DAFTAR REFERENSI

- Brooke, J. (1996). *SUS: A quick and dirty usability scale*. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271. <https://doi.org/10.1007/BF01386390>
- Elmasri, R., & Navathe, S. B. (2017). *Fundamentals of database systems* (7th ed.). Pearson.
- Google. (2024). *Dart documentation*. <https://dart.dev>
- Google. (2024). *Flutter documentation*. <https://docs.flutter.dev>
- Goodchild, M. F. (2018). Reimagining the history of GIS. *Annals of GIS*, 24(1), 1–8. <https://doi.org/10.1080/19475683.2018.1424737>
- Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B: Planning and Design*, 37(4), 682–703.
- Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic information science and systems* (4th ed.). John Wiley & Sons.
- Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann.
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
- Skiena, S. S. (2020). *The algorithm design manual* (3rd ed.). Springer.
- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human-computer interaction* (6th ed.). Pearson.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database system concepts* (7th ed.). McGraw-Hill.
- Tanenbaum, A. S., & Bos, H. (2015). *Modern operating systems* (4th ed.). Pearson.
- ISO. (2018). *ISO 9241-11:2018 Ergonomics of human-system interaction—Usability: Definitions and concepts*. International Organization for Standardization.
- OpenStreetMap Foundation. (2024). *OpenStreetMap*. <https://www.openstreetmap.org>
- Google Developers. (2024). *Google Maps SDK for Flutter*. <https://developers.google.com/maps/flutter-package>
- Kumar, A., & Gupta, R. (2021). Shortest path optimization using Dijkstra algorithm for mobile navigation systems. *International Journal of Advanced Computer Science and Applications*, 12(4), 215–222.

- Singh, P., & Sharma, V. (2022). Performance analysis of shortest path algorithms in location-based services. *International Journal of Computer Applications*, 184(12), 18–26.
- Zhou, X., & Golledge, R. G. (2017). *Spatial decision support systems*. CRC Press.
- Chang, K. T. (2021). *Introduction to geographic information systems* (9th ed.). McGraw-Hill.
- Burrough, P. A., McDonnell, R. A., & Lloyd, C. D. (2015). *Principles of geographical information systems*. Oxford University Press.
- De Smith, M. J., Goodchild, M. F., & Longley, P. A. (2020). *Geospatial analysis* (6th ed.). Winchelsea Press.
- W3C. (2023). *Geolocation API Specification*. <https://www.w3.org/TR/geolocation/>