Jurnal Kecerdasan Buatan dan Teknologi Informasi
Vol. 5, No. 2, May 2026, pp. 155-166
e-ISSN: 2964-2922, p-ISSN: 2963-6191
DOI : https://doi.org/10.69916/jkbti.v5i2.443

SHORTEST ROUTE SEARCH TO ACCOMMODATIONS NEAR MANDALIKA CIRCUIT USING DIJKSTRA'S ALGORITHM AND ANDROID-BASED LOCATION-BASED SERVICE

Moch. Syahrir*[1], Ahmad Subandi Azmi[2], Kurniadin Abd.Latif[3], Pahrul Irfan[4]

[1,2,3]Departement of Computer Science, Bumigora University, Mataram, Indonesia
[4]Department of Informatics Engineering, University Of Mataram, Mataram, Indonesia
Email: [1]moch.syahrir@universitasbumigora.ac.id, [2]bandiazmi31@gmail.com, [3] kurniadin@universitasbumigora.ac.id, [4]irfan@staff.unram.ac.id

(Received: April 16, 2026; Revised: April 27, 2026; Published: May 4, 2026)

Abstract

The development of mobile technology, particularly on the Android platform, has created significant opportunities for real-time, location-based applications. One important implementation is the use of Location Based Service (LBS) in the tourism sector to help tourists efficiently find strategic locations. This study focuses on developing an Android-based LBS application that integrates the Dijkstra Algorithm to determine the shortest route to accommodations around the Mandalika Circuit area, Kuta Beach, Lombok, a leading destination for MotoGP events in Indonesia. The system development adopts the waterfall model, consisting of requirement analysis, system design, implementation, and testing. In the analysis phase, user needs related to accommodation information and route navigation are identified. The design phase includes system architecture, user interface, and digital map integration. Implementation is carried out by developing an Android application capable of accessing real-time location data and processing route calculations using the Dijkstra Algorithm to produce the most efficient path. The resulting application displays the distribution of nearby accommodations, provides travel distance information, and offers optimal route guidance that can be directly accessed by users. System testing shows that the application runs according to the defined functional requirements. Additionally, evaluation using a Likert-scale questionnaire indicates a user satisfaction level of 84%, reflecting good acceptance and usability. In conclusion, this research successfully implements LBS technology combined with the Dijkstra Algorithm in a mobile application, providing practical solutions for tourists visiting the Mandalika Circuit area.

Keywords: location-based service; dijkstra algorithm; waterfall; likert scale; hotel.

1. INTRODUCTION

The construction of the Mandalika Circuit in the Mandalika Special Economic Zone (SEZ) has made this region a strategic world-class tourism destination. The hosting of international events such as the Mandalika MotoGP has further enhanced the region's appeal, driving a surge in both domestic and international tourist visits. According to data from the West Nusa Tenggara Provincial Government, the number of tourist visits to Lombok reached 1,376,295 in 2022, a significant increase compared to previous years[1]. This increase in tourist numbers has a direct impact on the need for supporting facilities, particularly accommodation around the Mandalika area. However, tourists often face challenges in finding optimal accommodation, both in terms of distance, travel time, and ease of access. This problem is compounded when tourists are unfamiliar with the area's geography. Therefore, a technology-based solution is needed that can provide location information and route navigation efficiently and accurately.

One approach is the use of Location-Based Service (LBS) technology, a service that utilizes users' geographic location information via mobile devices. LBS enables the provision of relevant information based on the user's real-time position, making it highly suitable for application in location search and navigation applications. In this context, integrating LBS with route finding algorithms is crucial to produce a system that is not only informative but also optimal in determining travel routes[2]. A commonly used algorithm for shortest route finding is Dijkstra's Algorithm. This algorithm works by calculating the path with the minimum weight in a non-negative weighted graph, thus generating the shortest route between two points. The advantage of Dijkstra's algorithm lies in its ability to provide optimal solutions deterministically, making it widely used in navigation and digital mapping systems[3].

Several previous studies have implemented Dijkstra's algorithm in various contexts. The study, entitled "Determining the Shortest Route to Shopping Centers in Jakarta Using the Dijkstra Algorithm"[4], demonstrated that this algorithm is effective in determining the fastest route in dense urban areas. Meanwhile, the study, "A GIS-Based Android Tourist Guide Application for Searching Nearby Tourist Attractions Using the Dijkstra Algorithm," successfully integrated geographic information systems with mobile devices to help users find nearby tourist attractions[5],[6]. In addition, several studies have also combined LBS with digital map services such as Google Maps API and GraphHopper to improve route search accuracy and performance.

Although these studies have demonstrated positive results, there are several limitations that constitute research gaps. First, most studies focus on general urban areas and have not specifically targeted international event-based tourism areas like Mandalika. Second, few studies have specifically examined finding the shortest route to accommodation, a key tourist need. Third, the integration of LBS, Dijkstra's algorithm, and the specific needs of tourists in the Mandalika Special Economic Zone (SEZ) has not been comprehensively explored[7]. Based on these challenges, this study aims to develop an Android-based application that utilizes LBS technology and the Dijkstra algorithm to determine the shortest route to accommodation around the Mandalika Circuit. This application is designed using digital map services and a navigation system that provides real-time information on location, distance, and optimal routes. With this application, it is hoped that tourists will find it easier to find suitable accommodation, thereby enhancing their comfort and experience during their visit to Mandalika[8].

Furthermore, this research is also expected to contribute to the development of location-based technology, particularly in the application of the Dijkstra algorithm in the tourism sector. Academically, this research provides added value through a contextual approach to national priority tourism areas, while enriching the literature related to LBS integration and route search algorithms in mobile applications[9].

2. RESEARCH METHODS

The methodology applied in this research is the Waterfall development method. The waterfall method is an approach commonly used by systems analysts. The basic principle of the waterfall method is that system development is carried out sequentially and linearly. This means that if one step is not completed, the next step cannot begin. Likewise, if the second step is not completed, the third step cannot begin, and so on. Automatically, the next step can only be executed if the previous steps have been completed[10].

3.1. Use Case Diagram

A use case diagram is a visual representation that models the behavior of a system and explains the interactions between one or more actors within the system. This diagram aims to identify the functions within a system and determine who has the right to use those functions[11].

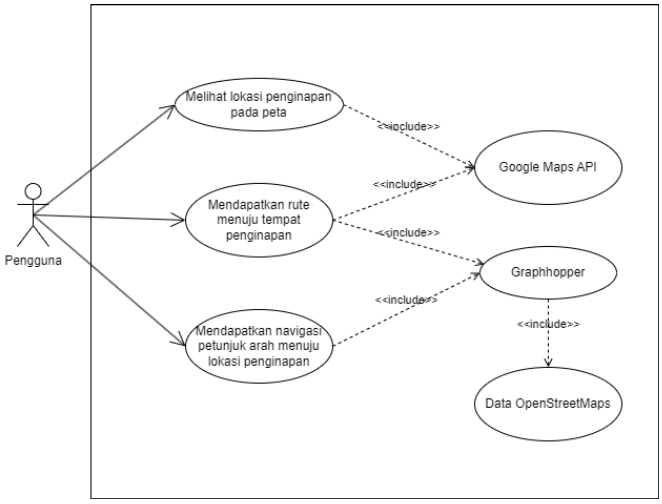

Figure 1. Use Case Diagram

2.2. Dijkstra's Algorithm

Dijkstra's algorithm is used to find the shortest path to the accommodation in this study. This algorithm is well-known in graph theory and is useful for finding the shortest path between two weighted vertices in a graph. The following is an example of applying Dijkstra's algorithm to find the shortest path between a starting point and a destination point[12].

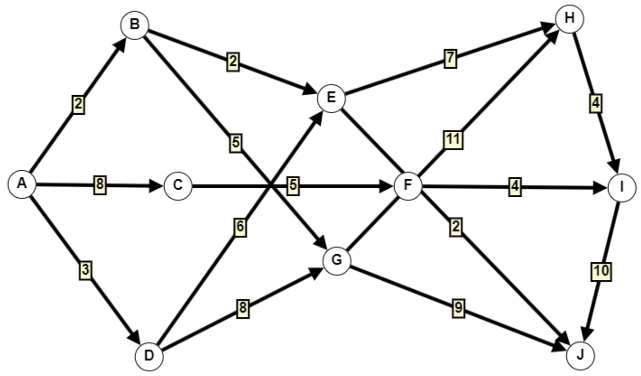

Figure 2. Weighted Graph

The graph above represents a road map. Next, we aim to find the shortest path from starting point A to ending point J, where A represents the user's location and J represents the location of the accommodation. The weights on each edge of the graph represent the distance between the roads.

The steps of Dijkstra's algorithm are as follows:

a. Initialization:
1. Dist[A] = 0 (because A is the starting point)
2. Dist[B] = ∞
3. Dist[C] = ∞
4. Dist[D] = ∞
5. Dist[E] = ∞
6. Dist[F] = ∞
7. Dist[G] = ∞
8. Dist[H] = ∞
9. Dist[I] = ∞
10. Dist[J] = ∞

b. Start Iteration:

1 st iteration:

V	A	B	C	D	E	F	G	H	I	J
A	$\underline{0_A}$	2_A	8_A	3_A	∞	∞	∞	∞	∞	∞

In the first iteration, the starting point is A, and point A has three neighbors, namely B, C, and D. Then, the distance values from point A to B, C, and D have been obtained. Point A is marked as the starting point and has the shortest weight by giving an underline. This marked point will not be iterated again.

2 st iteration:

v	A	B	C	D	E	F	G	H	I	J
A	**$\underline{0_A}$**	2_A	8_A	3_A	∞	∞	∞	∞	∞	∞
B		**$\underline{2_A}$**	8_A	3_A	4_B	∞	7_B	∞	∞	∞

From the 1st iteration, it can be seen that the point with the shortest distance is point B. So, in the 2nd iteration, starting from point B. Point B has two neighbors, namely E and G. From this iteration, the values of points E and G have been obtained. However, in the previous iteration, there was a shortest distance, namely point D. So, the shortest distance will be updated and the next iteration starts from point D.

3 st iteration:

v	A	B	C	D	E	F	G	H	I	J

A	**0_A**	2_A	8_A	3_A	∞	∞	∞	∞	∞	∞
B		**2_A**	8_A	3_A	4_B	∞	7_B	∞	∞	∞
D			8_A	**3_A**	9_D	∞	11_D	∞	∞	∞

From the 2nd iteration, we look for the smallest value, which is the value of D. So, the 3rd iteration starts from point D. Point D has two direct neighbors, namely E and G. However, their values are not smaller than point B which is related to points E and G in the previous iteration. So, in the next iteration, the values of points E and G used are the values from the results of the 2nd iteration.

4 st iteration:

v	A	B	C	D	E	F	G	H	I	J
A	**0_A**	2_A	8_A	3_A	∞	∞	∞	∞	∞	∞
B		**2_A**	8_A	3_A	4_B	∞	7_B	∞	∞	∞
D			8_A	**3_A**	9_D	∞	11_D	∞	∞	∞
E			8_A		**4_B**	∞	7_B	11_E	∞	6_E

The result of the 3rd iteration is not updated because the result is not smaller than the previous iteration. So, in the 4th iteration, starting from point E. Point E is directly connected to points H and J. Then, the shortest distance value is 6, so the next iteration starts from point J. From the 4th iteration, the shortest distance from point E to point J is obtained. When the end point or destination point has been marked, we have found the shortest path from the starting point A to the ending point J with a distance of 6. In this example, the shortest path from point A to point J is A->B->E->J.

2.3. Location-Based Service (LBS)

Location-Based Service (LBS) is not a system in itself, but rather a crucial service that utilizes embedded systems to support GSM systems[13] Location-Based Service (LBS) is an information service accessible via mobile devices, equipped with the ability to determine the user's location and provide information about available services based on that location[14].

2.4. Graphs

A graph is formed from a number of vertices connected by arcs. The notation for representing a graph is (G, E), where G is the set of vertices (vertices), and E is the set of arcs (edges) with specific attributes at each vertex. These attributes include the number of vertices, the number of arcs, and the length of the arc connecting vertices i and j, denoted as d(i, j). Graphs can be used to represent networks, and some examples of network models represented as graphs involve the design of oil pipelines, physical networks such as roads, railways, aircraft routes, power lines, and the like[6][15].

2.5. Android

Android is an open-source software operating system, allowing for open distribution so that users can create new applications for the platform. The Android platform provides developers with the opportunity to create applications that can run on a variety of mobile devices. Initially, Google Inc. acquired Android Inc., a start-up company developing software for mobile phones. To advance Android development, the Open Handset Alliance was formed, comprising 34 companies across various sectors such as hardware, software, and telecommunications, including Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, and Nvidia[16].

2.6. Shortest Path Problem

The shortest path problem (SPP) refers to the challenge of finding the shortest path between two or more vertices in a weighted graph, where the total weight of the edges traversed is minimal. The SPP can also be considered an optimization problem involving a weighted graph, where the weights can represent distances between cities, message delivery times, development costs, and the like[17].

2.7. Google Maps API

The Google Maps API refers to a set of application service programming functions and web-based mapping technology provided by Google Maps, enabling the integration of Google Maps into websites or applications. The Google Maps API consists of a collection of commands, functions, classes, and protocols that enable interaction between different software. Using an API key allows access to application interfaces that require permission to display Google Maps. In general, the Google Maps API is implemented in computer-based information systems that process and store geographic data or information[18]

2.8. Graphhopper

GraphHopper is a routing engine that functions as an open-source library for route planning, estimating travel times on road networks, and solving static vehicle routing problems. With high efficiency, GraphHopper implements various routing algorithms, including unidirectional Dijkstra, one-to-many Dijkstra, and unidirectional and bidirectional A*. Chosen for its extensibility and unmodified elevation profile management, GraphHopper also supports the Contraction Hierarchies (CH) algorithm to accelerate Dijkstra calculations, providing routes with the lowest weight values. While it does not require the input of travel time matrices for multiple locations, GraphHopper excels in its high-speed performance, even for long-distance routes exceeding hundreds of kilometers[19]

2.9. OpenStreetMap (OSM)

OSM is referred to as the Wikipedia of the world map; it is an internationally distributed, volunteer-run association whose individuals work to create a common digital map of the world. Its goal is to build and maintain a free, editable map database from around the world, so that users do not have to pay for geodata and are subject to copyright restrictions and licensing commitments. OSM has been one of the most successful Voluntary Information Graphics (VGI) projects in recent years. It is also frequently cited within the Geographic Information Systems (GIS) community and has coordinated the creation, production, and distribution of free mapping resources to support humanitarian relief efforts in many locations around the world since 2009.[15]. In addition, OSM has been shown to have more complete geodata in certain areas and is more local and semantically accurate compared to other datasets. Therefore, this shows how important the use of the OSM API is for this research in achieving its goals. The OSM mapping process is structured in a way as described by, where users can register and log in, create, edit, or delete maps, access the entire dataset, and retrieve the history of the entire dataset for free so that users can trace back every action taken[20].

2.10. Black-Box Testing

Software testing from a functional specification perspective is a testing process that does not involve checking the internal design or program code of the software. Its primary focus is to ensure that the software's functions, inputs, and outputs meet the specified specification requirements[21]

2.11. Questionnaires

A questionnaire is a data collection tool used to obtain information from respondents by providing a series of structured written questions. The goal is to gather respondents' responses regarding various variables being studied[22]

3. RESULTS AND DISCUSSION

3.1. SplashScreen Page

Figure 1. SplashScreen Page

The splash screen is the initial interface displayed when a user launches the application, serving as a brief introduction before accessing the main features. In this application, the splash screen is designed with a simple and clean layout to provide a clear and professional first impression. It prominently displays the text “Find Hotel,” which represents the core function of the application, namely helping users find suitable accommodation quickly and efficiently. Alongside the text, there is an icon positioned on the right side, which visually supports the concept of hotel searching or navigation. Although it appears simple, the splash screen has an important role in improving the overall user experience. It acts as a transition screen while the system prepares essential processes such as initializing the application, loading necessary data, and activating location-based services. The use of a minimalist design helps ensure that users are not overwhelmed with too much information at the beginning, while still clearly communicating the purpose of the application. Overall, the splash screen functions not only as an opening display but also as a medium to introduce the application’s identity and set user expectations regarding its features and usability.

3.2. Main Page

Figure 2. Main Page

The main page is displayed after the user passes the splash screen and serves as the core interface of the application, where users can interact with its primary features. This page is designed to provide both visual and textual information in an integrated manner to help users easily find nearby accommodations. It consists of two main components, namely the MapView and the RecyclerView, which work together to enhance usability and user experience.The MapView plays a central role by displaying a digital map that shows the user’s current location along with markers representing the locations of available accommodations. These markers are strategically placed based on geographic coordinates obtained through location-based services (LBS), allowing users to visually understand the spatial distribution of hotels or lodging options around them. Users can interact with the map by zooming, panning, or selecting markers to get a clearer view of specific locations. In addition to the map, the main page also features a RecyclerView that presents a structured list of nearby accommodations. Each item in the list contains basic information such as the name of the accommodation, its distance from the user’s current location, and possibly additional details like ratings or brief descriptions. This list format allows users to quickly compare multiple options without having to rely solely on the map.

Furthermore, each accommodation item is equipped with a button that enables users to view the route to the selected destination. When this button is pressed, the application utilizes routing functionality—based on algorithms such as Dijkstra, to calculate and display the shortest path from the user’s current position to the chosen accommodation. This feature provides practical navigation assistance, making it easier for users to reach their destination efficiently. Overall, the main page is designed to combine map-based visualization with list-based information, creating a user-friendly interface that supports both exploration and decision-making. By integrating these components, the application ensures that users can not only discover nearby accommodations but also access clear and actionable navigation guidance.

3.3. Route Page

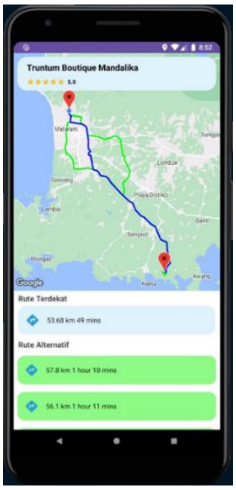

Figure 3. Route Page

This page is displayed when the user clicks the route button on the main page, and it serves as a detailed navigation preview interface. It provides users with a more comprehensive view of their journey before they begin traveling to the selected accommodation. The page prominently features a MapView that visualizes the planned route from the user's current location to the chosen hotel. This route is clearly highlighted on the map, allowing users to easily understand the direction, path, and overall distance they will travel. In addition to the visual route display, the page also presents essential information about the selected hotel in a clear and structured manner. This includes the name of the accommodation, the estimated distance from the user's current position, and the estimated travel time required to reach the destination. These details help users make quick decisions and better plan their trip based on time and proximity considerations. To enhance interactivity and functionality, the page includes two main action buttons: "Go" and "Share." The "Go" button provides a seamless transition to a real-time navigation mode, where the application actively guides the user step by step toward the destination, potentially integrating turn-by-turn directions for improved usability. Meanwhile, the "Share" button allows users to easily share the hotel's location with others through various platforms, such as messaging apps or social media, making it convenient for coordinating trips or providing recommendations. Overall, this page is designed not only as an informational interface but also as a decision-support and action-oriented feature. By combining route visualization, key travel information, and interactive controls, it enhances the user experience and ensures that users can navigate to their chosen accommodation with confidence, efficiency, and convenience.

3.4. Navigation Page

Figure 4 Navigation Page

This navigation page serves as an advanced guidance interface that actively assists users during their journey to the selected destination. It is designed to provide a real-time and interactive navigation experience, ensuring that users can travel efficiently and confidently. The page prominently displays a MapView that visualizes the user's current position along with the route that has been calculated. The route is clearly highlighted, making it easy for users to follow the path from their starting point to the destination. In addition to the map visualization, the navigation page also delivers step-by-step directions that guide users to the next segment of the route. These directions are dynamically updated based on the user's movement, allowing the system to adjust and provide accurate guidance even if the user changes direction or deviates from the planned path. This feature enhances reliability and ensures that users remain on the optimal route at all times. To further improve usability, the page provides important travel information in real time, including the estimated travel time, the remaining distance to the destination, and the estimated time of arrival (ETA). This information helps users manage their journey more effectively, enabling better planning and time management. Moreover, the navigation interface is designed with a user-friendly and intuitive layout, minimizing distractions while emphasizing essential information. Visual cues such as route lines, markers, and directional indicators are combined with concise textual instructions to create a seamless navigation experience. Overall, this page not only functions as a digital map but also as a smart navigation assistant that enhances convenience, accuracy, and user confidence throughout the trip.

3.5. Black Box Testing

The testing was conducted by the author using his personal device, the Xiaomi Redmi S2. The following table records the results of the black-box testing.

Table 1. Black-Box Testing Table

Test Case	Description	Input	Expected Output	se Observation Results	Remarks
Test case 1	Route Search from User Location to Accommodation	Starting coordinates, destination coordinates	Best route to accommodation and estimated time	The application can provide route search results from the user's location to the destination (accommodation)	[✓] Accepted [ ] Rejected
Test case 2	Searching for accommodation around the Mandalika Circuit	Mandalika Circuit coordinates, search radius limits	List of accommodations, basic accommodation information, and distances from the Mandalika Circuit to accommodations within a specified radius.	The application can provide information on accommodations around the Mandalika Circuit.	[✓] Accepted [ ] Rejected
Test case 3	Using GPS services	GPS service activation	The application can access the user's location with reasonable accuracy.	The application requests permission to use the GPS on the user's device. If the user allows it, the application will obtain the user's location.	[✓] Accepted [ ] Rejected
Test case 4	Using Location-Based Services (LBS)	The pull service requires a distance of 1.5 km from the Mandalika Circuit (coordinate point).	Getting accommodation information	The application provides information about accommodations around the Mandalika Circuit.	[✓] Accepted [ ] Rejected

Test case 5	Route search failed	Starting coordinates (location unreachable), destination coordinates (accommodation).	Message indicating route search failed	User's location is out of range (outside Lombok Island), the message "Your location is out of range" appears.	[✓] Accepted [ ] Rejected
Test case 6	Application performance	Starting coordinates (distant location), destination coordinates (lodging)	Route search results in a reasonable time and without errors	The application is capable of performing Dijkstra calculations and then displaying the resulting route very efficiently.	[✓] Accepted [ ] Rejected
Test case 7	Pressing the navigation button displays navigation	Starting coordinates (user location), destination coordinates (lodging)	Displays navigation directions to the destination (lodging), distance traveled, travel time, and estimated time of arrival.	The application provides navigation based on accurate route search results.	[✓] Accepted [ ] Rejected
Test case 8	Displaying alternative routes	Starting coordinates (user location), destination coordinates (accommodation).	Displays alternative routes on the map as a comparison to the main route.	The application provides alternative routes in addition to the main route	[✓] Accepted [ ] Rejected

3.6. Questionnaire

From the questionnaire, the following responses were obtained:

SS = Strongly Agree
S = Agree
N = Neutral/Uncertain
TS = Disagree
STS = Strongly Disagree

No	**Question**	**SS**	**S**	**N**	**ST**	**STS**	**Total**
1	Is the interface of this route search application easy to understand and use?	15	16	0	0	0	31
2	How well does this application provide information about hotel locations around the Mandalika Circuit?	16	14	1	0	0	31
3	How well does this application provide directions to accommodations?	17	13	1	0	0	31
4	Does this application help you find the best route to your accommodation?	16	12	3	0	0	31

5	Does this application help you find accommodations around the Mandalika Circuit?	15	13	3	0	0	31
6	Does this application meet your expectations?	10	18	0	3	0	31
7	How satisfied are you with the functionality of the Route Search application for Accommodations Around the Mandalika Circuit?	16	11	2	2	0	31
Total		**105**	**97**	**10**	**5**	**0**	**217**

From the questionnaire results table above, the level of application suitability can be calculated using the following formula:

Total score strongly agree and agree = (total respondens SS × 5) + (total responden S × 4)

$$(105 \times 5) + (97 \times 4) = 913$$

Total ideal score = number of respondents × total questions × 5

$$31 \times 7 \times 5 = 1085$$

$$\text{N} = \frac{\text{Total score strongly agree and agree}}{\text{total ideal score}} \times 100\%$$

If the total agree score is 913 and the total ideal score is 1085 then:

$$N = \frac{913}{1085} \times 100\% = 84\%$$

From the results of the questionnaire calculations, as many as 84% of respondents stated that they strongly agreed and agreed with the performance of the route search application using the Dijkstra algorithm.

3.7. Discussion

The results of this study indicate that the integration of Location-Based Service (LBS) and Dijkstra's Algorithm in an Android application provides a practical solution for route optimization in the tourism sector. The system is able to deliver accurate route calculations and real-time navigation, which are essential features for tourists unfamiliar with the Mandalika area. Compared to previous studies that focused on general urban navigation, this research specifically addresses the need for accommodation search in a tourism-focused region, thereby filling an important research gap. The use of Dijkstra's Algorithm proves effective in generating optimal routes with consistent performance, as demonstrated by the successful black-box testing results.

In addition, the user interface design, which combines MapView and RecyclerView, contributes to improved usability by allowing users to access both visual and textual information simultaneously. The questionnaire results, showing an 84% satisfaction rate, suggest that users perceive the application as helpful, easy to use, and relevant to their needs. However, several limitations are identified. The application relies on static route calculations without considering dynamic factors such as traffic conditions, road closures, or real-time events, which may affect route accuracy in real-world scenarios. Furthermore, the study is limited to the Mandalika area and does not yet support broader geographic scalability.

Despite these limitations, the findings highlight the potential of combining LBS and graph-based algorithms in supporting smart tourism solutions. This research contributes both practically, by providing a usable application, and academically, by demonstrating the effectiveness of Dijkstra's Algorithm in a specific tourism context. Future improvements can focus on enhancing system intelligence, expanding coverage, and integrating additional real-time data sources to further optimize performance and user experience.

4. CONCLUSION

This study successfully developed an Android-based application that integrates Location-Based Service (LBS) technology with Dijkstra's Algorithm to determine the shortest route to accommodations around the Mandalika Circuit area. The application is capable of displaying the distribution of nearby accommodations, providing distance and travel time information, and offering optimal route navigation in real time. Based on the results of black-box testing, all system functionalities operated in accordance with the defined requirements, indicating that the application is reliable and performs efficiently in various scenarios. Furthermore, the evaluation results obtained through a questionnaire show a user satisfaction level of 84%, which reflects a high level of acceptance and usability among users. These findings indicate that the application is effective in assisting tourists in finding suitable accommodations and navigating efficiently within the Mandalika area. Overall, this research demonstrates that the integration of LBS and Dijkstra's Algorithm can provide a practical and effective solution to support tourism activities, particularly in areas hosting international events such as Mandalika.

For future development, it is recommended to enhance the application by integrating more advanced routing algorithms, such as A* or machine learning-based approaches, to improve route optimization under dynamic conditions such as traffic congestion. Additionally, incorporating real-time traffic data and user preferences, such as budget, accommodation ratings, and facility filters, would further improve the relevance and personalization of the recommendations. Expanding the coverage area beyond the Mandalika region and integrating additional data sources, such as online booking systems, could also increase the application's usefulness. From a technical perspective, improving the user interface design and adding offline map capabilities would enhance accessibility and user experience, especially in areas with limited internet connectivity. Lastly, further research can explore the integration of this system with broader smart tourism frameworks to support digital transformation in the tourism sector.

REFERENCES

[1] I. W. Yasa, "ANALISIS RISIKO PEMBANGUNAN SIRKUIT MOTOR GP KAWASAN KEK MANDALIKA LOMBOK TENGAH," 2023.

[2] M. Dimyathi, E. Santoso, and R. K. Dewi, "Sistem Informasi Presensi Menggunakan Teknologi Location Based Service (LBS)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 6, pp. 2024–2031, 2018.

[3] N. H. Adi, M. Giatman, W. Simatupang, A. Afrina, and R. Watrianthos, "Penerapan Metode Dijkstra Pada Jalur Distribusi LPG Untuk Penentuan Jarak Terpendek," *Build. Informatics, Technol. Sci.*, vol. 3, no. 3, pp. 235–243, 2021, doi: 10.47065/bits.v3i3.1052.

[4] J. Setiawan *et al.*, "PERBELANJAAN DI JAKARTA MENGGUNAKAN ALGORITMA DIJKSTRA," vol. 21, no. 3, pp. 156–165, 2019.

[5] R. Perayoga, P. Hendradi, and A. Setiawan, "Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek Objek Wisata," vol. 4, no. 3, 2023, doi: 10.30865/klik.v4i3.1495.

[6] H. Fithra *et al.*, "Real-time Public Transportation Tracking System Application Using Android-Based Dijkstra Algorithm," *Int. J. Informatics Vis.*, vol. 10, no. 1, pp. 154–159, 2026, doi: 10.62527/joiv.10.1.2322.

[7] F. Multidimensional *et al.*, "Foundations of Multidimensional and Metric Data Structures The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling".

[8] A. N. Cahyawan, N. Faizah, and W. Nurcahyo, "Development of an Android-Based Geographic Information System for Maneungteung Tourism Area in Cirebon Regency Using Location-Based Service (LBS) Method with Android Studio Version 2024.1.2," *J. Mob. Technol.*, vol. 3, no. 1, pp. 39–48, 2025, doi: 10.59431/jms.v3i1.558.

[9] S. Steiniger, M. Neun, and A. Edwardes, "Foundations of Location Based Services," pp. 1–28.

[10] M. Rumondang Tampubolon, Zainal Rahman Malik, Reni Reina Nurul Ainun Nissa, Yezika Oktarmila, Rikcy Agustius Sinaga, and Putri Mentari Endraswari, "Print and Photocopy Service System Design Using Mobile Based Waterfall Method," *BITJournal Bangka Inf. Technol. J.*, vol. 1, no. 1, pp. 1–9, 2024, doi: 10.33019/5b21yb67.

[11] D. Hindarto, "Architecture and High Performance Computing Indonesian Culinary Application System Design with UML Method Journal of Computer Networks , Architecture and High Performance Computing," *J. Comput. Networks ,* vol. 5, no. 2, pp. 612–622, 2023.

[12] E. W. Dijkstra, "Dijkstra1959.pdf," 1959. [Online]. Available: http://www.bioinfo.org.cn/~dbu/AlgorithmCourses/Lectures/Dijkstra1959.pdf

[13] N. Hijriani, V. A. Nintyas, and A. Ibrahim, "Aplikasi Sistem Informasi Pemetaan Lokasi Universitas Sriwijaya Dengan Memanfaatkan Google Maps Berbasis Android," *JSI J. Sist. Inf.*, vol. 13, no. 1, pp. 2216–2220, 2024, doi: 10.18495/jsi.v13i1.38.

[14] A. Herwanto, F. Ilmu, K. Universitas, and E. Unggul, "Analisa Perancangan Penjualan Rumah Berbasis".

[15] F. Y. Nurhasanah, W. Gata, D. Riana, M. Jamil, and S. F. Saputra, "Shortest Path Finding Using Dijkstra's Algorithm," *PIKSEL Penelit. Ilmu Komput. Sist. Embed. Log.*, vol. 9, no. 1, pp. 89–102, 2021, doi: 10.33558/piksel.v9i1.2365.

[16] S. E. (2021). Westermann, K., Woud, M. L., Cwik, J. C., Graz, C., Nyhuis, P., Margraf, J., & Blackwell, "LITERATUR REVIEW: PERANCANGAN APLIKASI MOBILE LEARNING TENSES BERBASIS ANDROID," p. 6, 2021.

[17] Y. P. Hutapea, C. E. J. C. Montolalu, and H. A. H. Komalig, "Algoritma Dijkstra untuk Penentuan Lintasan Terpendek Pada Kasus Tujuh Hotel di Kota Manado Menuju Bandara Sam Ratulangi Manado," *J. Mat. dan Apl.*, vol. 9, no. 2, pp. 158–163, 2021.

[18] Y. Fito *et al.*, "Implementasi Google Maps API Berbasis Website," *Res. Fac. Eng. Palangka Raya Univ.*, no.

September, pp. 1–6, 2021.
[19] A. W. Sari, P. W. L. Bimantari, and N. P. Sakhia, *Development of Phantom Radiology Using Eggshells Powder as Bone Genu Material*, vol. 746 LNEE. 2021. doi: 10.1007/978-981-33-6926-9_48.
[20] M. Minghini and E. Commission, "CH A PT ER 3 A Review of OpenStreetMap Data," no. September, pp. 37–59, 2017.
[21] P. K. Ayuningtyas, D. Atmodjo WP, and P. Rachmadi, "Performance And Functional Testing With The Black Box Testing Method," *Int. J. Progress. Sci. Technol.*, vol. 39, no. 2, p. 212, 2023, doi: 10.52155/ijpsat.v39.2.5471.
[22] R. Bihu and ABSTRACT:, "QUESTIONNAIRE SURVEY METHODOLOGY IN EDUCATIONAL AND SOCIAL SCIENCE STUDIES," vol. 9, no. 3, p. 6, 2021.