

Decision Support System for Determining Priority Routes for Safety Vehicles for Emergency Response in Densely Populated Areas Using GIS-Based Dijkstra Algorithm

Firmansyah¹, Firahmi Rizky²

Program Studi Teknologi Informasi Fakultas Ilmu Komputer Dan Teknologi Informasi
Universitas Muhammadiyah Sumatera Utara

Corresponding Author e-mail : firminnsyah@gmail.com, firahmirizky@umsu.ac.id

Article History:

Received: 13-10-2026

Revised: 26-10-2026

Accepted: 26-10-2026

Keywords: Dijkstra's Algorithm, Safety Vehicle, Emergency Handling, Geographic Information System, Decision Support System

Abstract : Traffic density and limited information on alternative routes can hinder the mobility of safety vehicles in handling emergencies in densely populated areas. This study aims to develop a Geographic Information System (GIS)-based Decision Support System for Safety Vehicle Priority Route Determination using the Dijkstra algorithm in Medan Tembung District. The study uses a system development method through the stages of problem identification, data collection, analysis, design, implementation, and testing. Data were obtained through field observations and secondary OpenStreetMap data. The study population includes the road network in Medan Tembung District, while the sample consists of the road network and location points in a simulation of the journey from Medan Haji Hospital to the emergency location. The analysis was carried out by representing the road network as a distance-weighted graph and comparing the results of manual calculations with the system output. The results show that the manual calculation produces a distance of 3.885 km, while the system displays a distance of 3.97 km with an estimated travel time of 6 minutes. Black Box testing shows that all tested features function as expected. This system can help visualize locations and determine routes based on distance, but does not take into account real-time traffic conditions.

How to Cite : Firmansyah, Firahmi Rizky. (2026). The title. Ambidextrous Journal of Innovation Efficiency and Technology in Organizations. 5(3). pp 595-607
<https://doi.org/10.61536/ambidextrous.v5i03.952>



<https://doi.org/10.61536/ambidextrous.v5i03.952>

This is an open-access article under the [CC-BY-SA License](https://creativecommons.org/licenses/by-sa/4.0/).



Introduction

Areas with high population density face various mobility, traffic, and public service challenges, particularly in emergency response. Busy and narrow roads and a high number of vehicles can hinder the mobility of emergency vehicles, such as ambulances, fire trucks, and police vehicles, in reaching the scene of an incident. Delays in emergency response can impact public safety and increase the risk of loss of life and property. Wang et al. (2013) explain that strategies for determining routes and prioritizing emergency vehicles can help reduce travel delays while minimizing the impact on general traffic flow. This situation is a significant concern in Medan Tembung District, Medan City, which has dense residential areas, educational institutions, and centers of economic activity. The high volume of vehicles in the area, particularly during rush hour, has the potential to cause traffic congestion that hinders the movement of emergency vehicles to the scene of an emergency.

The problem of determining safety vehicle routes in densely populated areas is related to limited information about the road network, changing traffic conditions, and the suboptimal use of digital technology to support decision-making. In practice, route determination can still rely on driver experience and manual information, so available alternative routes are not necessarily systematically considered. Inappropriate route selection can extend travel times and reduce the effectiveness of emergency response. Similar problems are also found in the public service sector in urban areas, which have not fully utilized spatial data-based decision-making support technology (Hendi Muhammad & Yuniansyah, n.d.). Research by Oh and Ashuri (2023) shows that the use of route-finding algorithms and spatial data can help analyze emergency vehicle journeys and identify factors and road sections related to emergency response times. Therefore, an approach capable of integrating geographic information and road network analysis is needed to help determine safety vehicle routes in a more targeted manner.

One technology that can be utilized to support these needs is a Geographic Information System (GIS), which enables the management and presentation of geographic information in the form of road networks and incident locations through interactive digital maps. The use of GIS can support visual and integrated route analysis, thus aiding the decision-making process in emergency response. Previous research has shown that the integration of GIS and the Dijkstra algorithm can produce the shortest route in the context of evacuation route planning. Pramudya and Subiyanto (2015), for example, applied the Dijkstra algorithm through network analysis to plan evacuation routes for landslides in Semarang City. Another study by Yulia and Triase (2024) discussed the use of the Dijkstra algorithm in a geographic information system for mapping and searching for nearby locations in Medan City. However, the application of this technology in the context of determining safety vehicle routes still needs to be specifically directed at the needs of emergency response in densely populated areas, particularly Medan Tembung District. Thus, a system is needed that not only presents road network information but also is able to provide route recommendations based on available route calculations.

In determining the shortest route on a road network, the Dijkstra algorithm can be used to calculate the minimum distance between the starting point and the destination point through interconnected nodes in a weighted graph with non-negative values. This algorithm allows for systematic path finding, thus supporting the determination of safety vehicle routes (Berutu et al., 2025). The application of the Dijkstra algorithm in geographic information systems has also been used to determine the shortest path based on point representation and road networks, as demonstrated by Pramudita and Safitri (2019). However, the determination of priority routes in this study also considers road conditions, distance, and traffic congestion levels as supporting criteria. Therefore, a GIS-based decision support system is needed that integrates road network data with the Dijkstra algorithm to generate route recommendations to the location of an emergency incident. This system is expected to help officers consider alternative routes in a more structured manner and present the calculation results in the form of an easy-to-understand digital map.

This study aims to design a GIS-based decision support system to determine priority routes for safety vehicles in Medan Tembung District, applying the Dijkstra algorithm in finding the shortest route, and generating route recommendations visualized through digital maps. This research is important to support the speed and accuracy of decision-making in handling emergencies in densely populated areas. The novelty of this research lies in the application of GIS integration and the Dijkstra algorithm specifically directed at determining priority routes for safety vehicles in Medan Tembung District. This system is expected to support a more systematic and efficient route determination process, thereby

increasing the effectiveness of emergency handling and the quality of public services in the field of safety.

Method

This study uses a system development method to design a Decision Support System for Determining Priority Routes for Safety Vehicles for Emergency Response in Densely Populated Areas Using the Dijkstra Algorithm Based on a Geographic Information System (GIS). The system is developed as a web-based application that helps users determine routes to emergency locations in Medan Tembung District. The development process is carried out through the stages of problem identification, data collection, data analysis, system design, implementation, and testing. These stages are in line with the software engineering approach that places requirements analysis, design, implementation, and testing as important parts of the system development process (Pressman & Maxim, 2020).

Dijkstra's algorithm is used to find the shortest route based on weighted distances in a road network represented as a graph. This algorithm determines the minimum distance from a starting point to a destination point through nodes connected with non-negative weights (Dijkstra, 1959). Meanwhile, GIS is used to manage and visualize geographic information and route search results in the form of digital maps.

The research instruments include hardware in the form of a laptop with an Intel Core i5 processor, 8 GB RAM, 512 GB SSD storage, and the Windows 11 operating system. The software used consists of Visual Studio Code for writing program code, XAMPP for running a local server, PHP for application development, and MySQL for database management.

Data collection was conducted through field observations and the use of secondary data from OpenStreetMap, as well as distance and route information used as a comparison in system testing. OpenStreetMap is an open geographic data source that includes road network information and various geographic elements collected and updated by contributors (OpenStreetMap Foundation, n.d.). The data collected includes safety vehicle starting points, intersection points, emergency incident locations, road networks, distances between points, as well as information on road conditions and traffic congestion levels.

The data analysis technique is performed by processing the road network into a graph consisting of nodes representing location points and edges connecting them. The distance between nodes is used as the weight for each edge in the Dijkstra algorithm calculation. The calculation process begins by setting the distance from the starting point to zero and the distance from the other points to infinity, then updating the distance estimate through the connected points until a path to the incident location is obtained (Dijkstra, 1959).

Road conditions and congestion levels are used as supporting information to understand the road network, while the primary calculation of Dijkstra's algorithm uses distance weights. The manual calculation results are then compared with the route and distance displayed by the system to determine whether the route search results are consistent.

The research population includes the road network in Medan Tembung District which is used as the basis for determining priority routes for safety vehicles. The research sample consists of the road network and location points used in the route search simulation from Medan Haji Hospital to the emergency location with coordinates 3.619653, 98.692814. The points used are represented through nodes A to H, which include the starting point, intersection or road section, and the location of the incident. The data was selected based on the needs of the manual calculation process of the Dijkstra algorithm and system testing in determining the route to the emergency location.

The research procedure began by identifying the problems of determining safety vehicle routes in Medan Tembung District, specifically mobility constraints due to traffic congestion, limited information on alternative routes, and the suboptimal use of spatial information in route determination. The next stage was collecting primary data through field observations and secondary data from OpenStreetMap, along with route information for comparison. The road network data was then processed into a graph consisting of nodes and edges, with the distance between points as weights for the Dijkstra algorithm calculation.

The next step is to perform manual calculations to determine the shortest route from the starting point to the emergency location. The system is then designed using Unified Modeling Language (UML) diagrams, including Use Case Diagrams, Activity Diagrams, Sequence Diagrams, and Class Diagrams, to describe user interactions, activity flows, process sequences, and system structure. UML can be used to model system requirements, behavior, interactions, and structure during software analysis and design (Unhelkar, 2017).

Implementation was carried out by developing a web-based application that provides login features, a dashboard, node data management, location recording and emergency reports, GIS maps, and incident history. In the final stage, the system was tested by comparing the route and distance results obtained through manual calculations with the results displayed in the application. The test results were used to determine the suitability of the Dijkstra algorithm application in supporting the determination of priority routes for safety vehicles to emergency incident locations.

Results and Discussion

Implementation of Dijkstra's Algorithm

The Dijkstra algorithm is implemented to determine the shortest route from the initial location of the safety vehicle to the emergency scene. In the developed system, the road network is represented by nodes and edges. Nodes are used as points on the road network, while edges are connections between nodes, each with a distance value as a weight. This data is then used by the Dijkstra algorithm to find the path with the shortest total distance.

1. Road Network Formation

The initial step in the route discovery process is to construct a road network based on node and edge data. This data is retrieved from a database and then used to determine the relationships between nodes.

```
if (
    isset($graph[$awal]) &&
    isset($graph[$tujuan])
) {
    $graph[$awal][] = [
        'tujuan' =>
            $tujuan,
        'jarak' =>
            $jarak
    ];
}
```

Figure 1. Road Network Formation Program

The road network is formed by connecting the starting node with the destination node based on available edge data. Each connection between nodes is assigned a distance value, which is used as a weight in the network. The system also creates two-way connections so that road sections can be used from the starting node to the destination node or vice versa.

2. Determining the Starting Point and Destination Point

Once the road network is established, the system determines the starting point and the destination point. The starting point is the location of the safety vehicle, while the destination point is the selected emergency location.

```

$nodeTerdekat = null;
$jarakKeLokasi = INF;

foreach (
    $nodes as $id => $node
) {

    $jarakGPS = hitungJarakGPS(
        $latitudeDarurat,
        $longitudeDarurat,
        $node['latitude'],
        $node['longitude']
    );

    if (
        $jarakGPS < $jarakKeLokasi
    ) {

        $jarakKeLokasi =
            $jarakGPS;

        $nodeTerdekat =
            $id;

    }

}

```

Figure 2. Node Determination Program

This section shows the process of determining the closest node to the incident location. This process is performed by calculating the distance between the emergency location coordinates and each available node. Next, each distance is compared, and the node with the shortest distance is selected as the closest node. This node is then used to connect the incident location to the road network in the route search process.

3. Dijkstra's Algorithm Process

Once the starting and destination points are determined, the system begins the route search process using Dijkstra's algorithm. In the initial stage, each node is assigned an infinite initial distance, while the starting node is assigned a distance of 0. The system also records the nodes visited and the previous nodes used to construct the route.

```

while (true) {

    $nodeSekarang = null;
    $jarakTerkecil = INF;

    foreach (
        $jarak as $id => $nilai
    ) {

        if (
            !$dikunjungi[$id] &&
            $nilai < $jarakTerkecil
        ) {

            $jarakTerkecil =
                $nilai;

            $nodeSekarang =
                $id;

        }

    }

}

```

Figure 3. Algorithm Process

The code shows the route search process using Dijkstra's algorithm. The system searches for the node with the smallest distance value and has not yet been visited. This node is then selected as the next node to be processed. This process is repeated to determine the shortest path to the destination.

4. Determining Dijkstra's Result Path

After Dijkstra's algorithm completes, the system generates a route based on the previous nodes stored during the calculation. The system traces the nodes from the destination back to the starting point, then reverses their order to obtain a route from the starting point to the incident location. The system then calculates the total route distance and saves the result for display on the results page.

```
$route = [];
$current =
    $nodeVirtual;

while (
    $current != null
) {

    $route[] =
        $current;

    if (
        !isset(
            $sebelumnya[$current]
        )
    ) {
        break;
    }
    $current =
        $sebelumnya[$current];
}

$route =
    array_reverse(
        $route
    );

$totalJarak =
    $jarak[$nodeVirtual];
```

Figure 4. Route Formation

The code is used to generate a route based on the results of Dijkstra's algorithm calculations. The system traces the nodes from the destination to the starting point based on the previous nodes, then reverses the order to create a route from the starting point to the incident location. Once the route is generated, the system calculates the total distance obtained from the calculations.

Digital Map Interface Implementation

The system uses Leaflet.js to display a digital map as a visualization of the location and the resulting route. The digital map displays the location of the incident and the path obtained from the route search process using Dijkstra's algorithm. This map allows users to more clearly see the predetermined route through a geographic view.

```
var map = L.map('map').setView([3.6200, 98.7300], 14);

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  attribution: '&copy; OpenStreetMap contributors'
}).addTo(map);
```

Figure 5. Implementation of Digital Maps

The digital map implementation in the system uses the Leaflet library. The map uses OpenStreetMap as the base map used to display location information and the road network in the system.

Interface View

The user interface is the part that acts as a link between the user and the system. At this stage, the previously created design is implemented into a usable page. The system interface is kept simple and easy to understand so that users can easily operate each available function.

1. Login Page

Figure 6. Login Form

The login page is used by users to access the system. Users must enter their registered username and password to access the system's features. On the login page, the system provides access based on the user's agency: Police, Fire Department, and Ambulance. Each user will be directed and granted access based on the agency they use to log in, allowing the information and reports displayed to be tailored to each agency's needs.

2. Dashboard Page

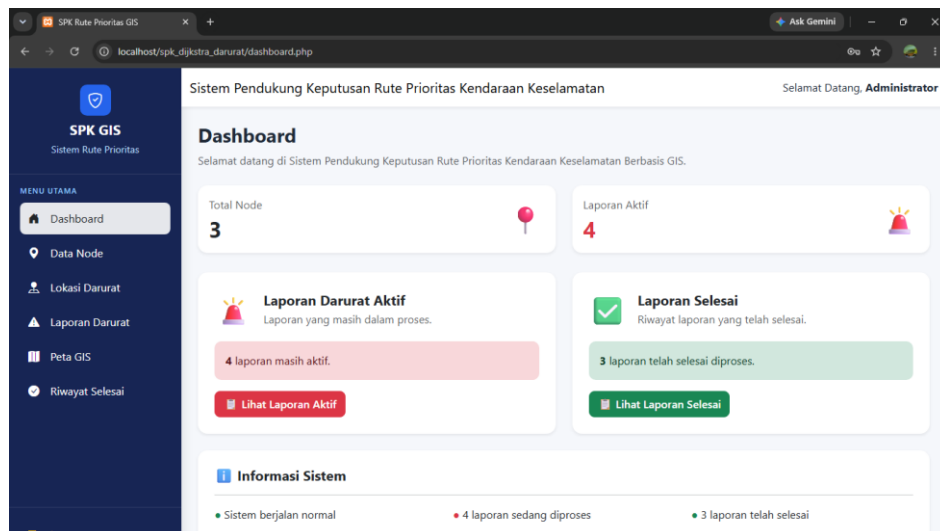


Figure 7. Dashboard

The dashboard is the main page a user sees after successfully logging into the system. This page displays brief information about the system and several main menus. The dashboard serves as the initial page for accessing the system's available features.

3. Node Data Page

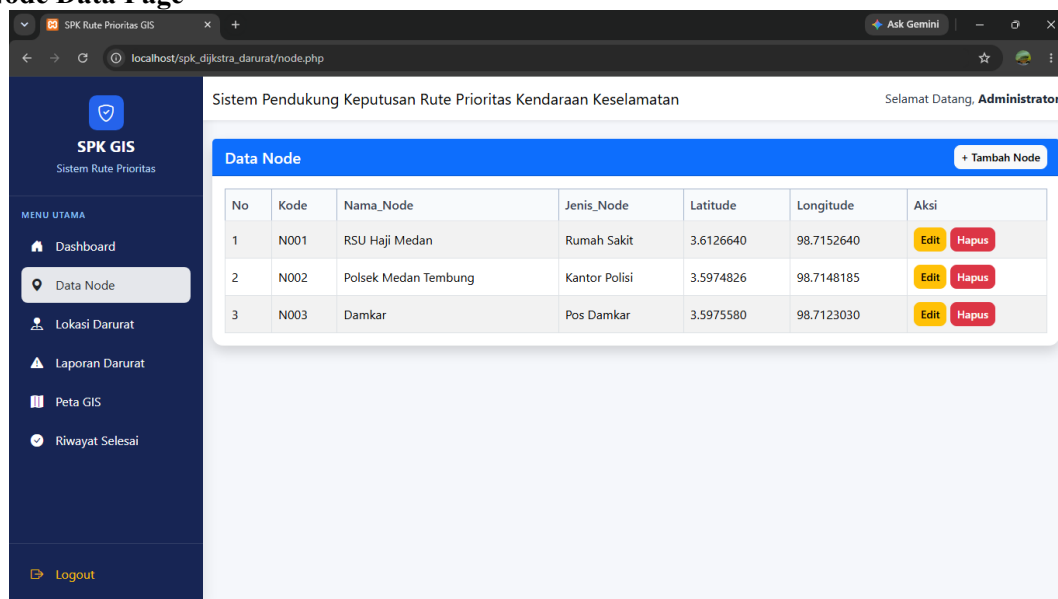


Figure 8. Data Node

The Node Data page is used to manage road network point data used in the route search process. On this page, users can view stored node data and manage the data according to system requirements.

4. Emergency Location Page

Sistem Pendukung Keputusan Rute Prioritas Kendaraan Keselamatan

Selamat Datang, Administrator

Tambah Lokasi Darurat

Kode Darurat
Contoh: D001

Nama Lokasi
Contoh: Jalan Tuasan

Jenis Kejadian
-- Pilih Jenis Kejadian --

Lokasi Terdeteksi
Belum mendeteksi lokasi Gunakan Lokasi Saya

Klik tombol jika ingin menggunakan lokasi GPS otomatis.

Latitude
Contoh: 3.611897

Longitude
Contoh: 98.6969843

Bisa diketik manual atau diisi otomatis.

Cari Node Terdekat Node belum dihitung.

Figure 9. Emergency Location

The Emergency Locations page is used to enter data about emergency incidents. On this page, users can enter information about the location of the incident, the type of incident, the required agency, a description, and the location coordinates.

5. Emergency Report Page

Sistem Pendukung Keputusan Rute Prioritas Kendaraan Keselamatan

Selamat Datang, Administrator

Laporan Darurat

Daftar laporan darurat yang masuk dan belum selesai diproses.

Data Laporan Masuk

No	Kode Laporan	Nama Lokasi	Jenis Kejadian	Keterangan	Koordinat	Status	Aksi
1	DR-20260819124547	Lokasi GPS	Orang Sakit	sakit	Lat: 3.6196530 Long: 98.6928140	Aktif	Lihat Lokasi Selesai
2	DR-20260819122245	Lokasi GPS	Tindak Kriminal	pencurian sepeda motor	Lat: 3.6226220 Long: 98.6973200	Aktif	Lihat Lokasi Selesai
3	DR-20260819122111	Lokasi GPS	Tindak Kriminal	pencurian	Lat: 3.6226220 Long: 98.6973200	Aktif	Lihat Lokasi Selesai
4	DR-20260815113844	Lokasi GPS	Kecelakaan	tabrakan	Lat: 3.6111454 Long: 98.6969315	Aktif	Lihat Lokasi Selesai

Logout

Figure 10. Emergency Report

The Reports page is used to view incident reports that have been entered into the system. On this page, users can view available report information according to the agency they logged in to. The system restricts report access by agency, so Fire, Police, and Ambulance officers can only view reports addressed to their respective agencies. Active reports can be further processed by the relevant officer, including determining the route of the safety vehicle to the incident location. Once a report has been processed, the officer can change the report status to completed so that the report is no longer displayed as an active report.

6. GIS Map Page

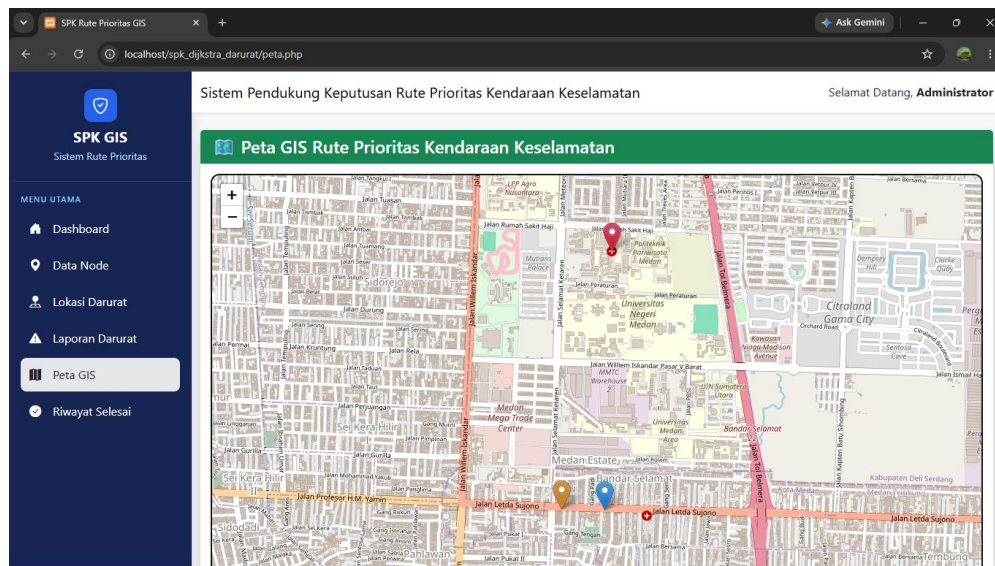


Figure 11. GIS Map

The GIS Map page displays the location of the incident and the route search results. The map allows users to see the location in real time and identify the safety vehicle's route. This page displays the incident location, the vehicle's starting point, and the route generated by the route search process using Dijkstra's algorithm.

7. History Page

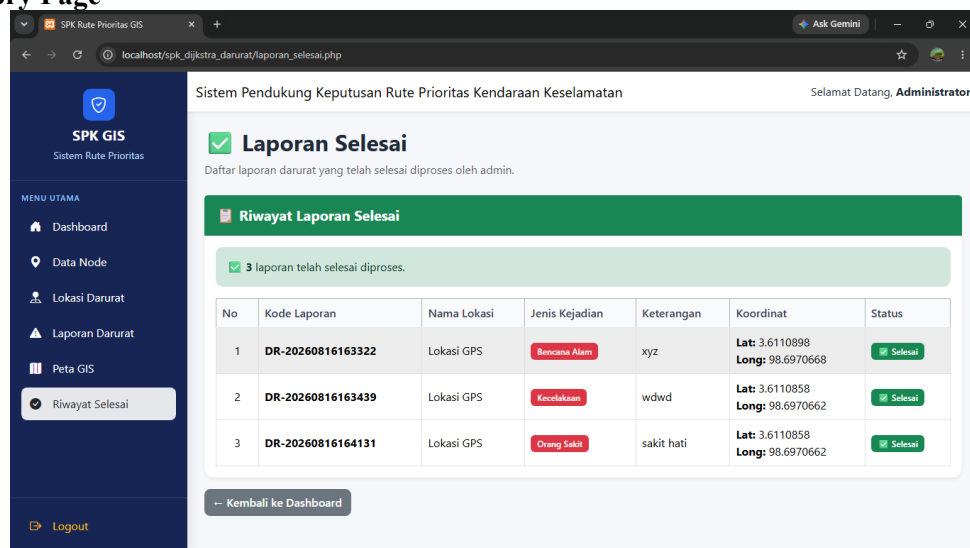


Figure 12. History

The Completion History page displays reports that have been processed. Reports that were previously active will change their status to Completed after processing is completed. This page is used to save and view completed reports, clearly distinguishing active and completed reports.

Interface Testing

Interface testing is performed to ensure that every page on the system can be used according to its designed function. Testing is performed by trying each available menu and feature, then determining whether the system's output matches expectations. The method used in this testing is Black Box Testing. Testing is conducted from the user's perspective by providing input to the system without observing the program code running within it.

Black box testing

Black Box Testing is performed to determine whether each function within a system operates as expected. Testing is performed by providing input to the system and then observing the resulting output. This testing does not examine the processes or program code used within the system, but only the functions that can be used by users. This testing focuses on the functional aspects of the system.

Table 1. Black box testing

NO	Features tested	Test scenario	Desired results	Results
1	Login Page	Open the system login page	The login page displays the username, password fields and the Login button.	Succeed
2	Dashboard Page	Log in and go to the main page	The dashboard displays system information, number of nodes, active reports, completed reports, and a navigation menu.	Succeed
3	Data Node	Open the Data Node menu	The system displays node data in the form of code, node name, node type, latitude, longitude, and Edit and Delete buttons.	Succeed
4	Emergency Location	Open the Emergency Location menu	The system displays a form for entering the emergency code, location name, type of incident, location, latitude, and longitude.	Succeed
5	Emergency Report	Open the Emergency Report menu	The system displays a list of incoming reports along with the report code, location, type of incident, description, coordinates, status and action.	Succeed
6	GIS Map	Open the GIS Map menu	The system displays a digital map and the location of the incident in the form of a marker.	Succeed
7	Completed History	Open the Completed History menu	The system displays a list of reports that have been processed along with information on location, type of incident, description, coordinates and status.	Succeed

Blackbox Testing Results

Based on the results of the interface testing and Black Box Testing that have been carried out, it can be seen that the GIS-based Safety Vehicle Priority Route Decision Support System using the Dijkstra algorithm has run according to the functions and needs that have been designed. Testing was carried out on several main features, namely login, dashboard, node data, emergency location, emergency report, GIS Map, and completion history. The test results show that each feature can receive input and produce output as expected, starting from the process of logging into the system, managing node data and incident locations, displaying emergency reports, to visualizing locations on the GIS Map and storing the history of completed reports. Based on these results, all tested features were declared successful, so it can be concluded that the interface and main functions of the system have run well and can be used according to the needs that have been determined in the study.

1. System Advantages

Based on the results of the tests conducted, the system has several advantages. The system can assist users in reporting emergency incidents and determining the location of the incident based on the provided coordinates. The system can also display the incident location on a GIS map, making it easier to locate the incident. Furthermore, the system can determine and display priority routes from the starting point to the incident location using Dijkstra's algorithm. The displayed route results also provide information on distance and estimated travel time, helping users determine the correct route.

2. System Deficiencies

Despite its advantages, the system developed still has several shortcomings. The system still relies on road network data in the form of available nodes and edges, so changes or additions to the road network need to be updated in the system. Using GIS maps also requires an internet connection for proper map display. Furthermore, the accuracy of location coordinates still depends on the accuracy of the user's device's GPS. The system also does not utilize real-time traffic conditions, so conditions such as congestion, accidents, or road closures that occur after the network data is generated cannot be directly taken into account in route determination.

Conclusion

This research produces a Decision Support System for Determining Priority Routes for Safety Vehicles for Emergency Response in Densely Populated Areas Using the Dijkstra Algorithm Based on Geographic Information System (GIS) in Medan Tembung District. The system is developed in the form of a web-based application that integrates road network data management, location recording and emergency reports, and route visualization through digital maps using Leaflet.js with OpenStreetMap as a base map. Manual calculations on the trip simulation from RSU Haji Medan to the incident location resulted in a distance of 3.885 km, while the system displayed a distance of 3.97 km with an estimated travel time of 6 minutes. The results of Black Box Testing show that all tested features, including login, dashboard, node data, emergency locations, emergency reports, GIS maps, and completion history, function according to the expected results. These findings indicate that the system can help users obtain incident location information and route recommendations based on road network distance calculations.

However, this study has limitations because route determination still depends on the availability and update of node and edge data, the accuracy of GPS coordinates, and an internet connection to display digital maps. Furthermore, the system does not account for changes in real-time traffic conditions, so congestion, accidents, and road closures cannot be directly calculated in route searches. Future research is recommended to develop the integration of the latest traffic data and periodically update the road network so that route recommendations are more in line with field conditions. Practically, this system can be used as an aid for ambulance personnel, firefighters, and police in identifying incident locations and determining routes based on available distances. Thus, the system is expected to support the decision-making process and improve the effectiveness of emergency response in Medan Tembung District.

Bibliography

- Berutu, I., Auzi, S., Ashillah, S., & Harliana, P. (2025). Integrasi algoritma Dijkstra pada aplikasi QGIS untuk simulasi rute tercepat di Medan: Studi kasus rute dari Universitas Negeri Medan ke Rumah Sakit Royal Prima. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(1), 453–461. <https://doi.org/10.36040/jati.v9i1.12296>
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271. <https://doi.org/10.1007/BF01386390>
- Oh, H. J., & Ashuri, B. (2023). Enriching GPS data for expanding interpretation of emergency vehicles using a pathfinding algorithm and spatial data harvesting methods. *Sustainable Cities and Society*, 95, Article 104600. <https://doi.org/10.1016/j.scs.2023.104600>
- OpenStreetMap Foundation. (n.d.). Copyright and license. OpenStreetMap. <https://www.openstreetmap.org/copyright>
- Pramudita, R., & Safitri, N. (2019). Algoritma Dijkstra untuk menentukan jalur terpendek dalam sistem informasi geografis. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 7(1), 33–42. <https://doi.org/10.33558/piksel.v7i1.1662>
- Pramudya, R. A., & Subiyanto, S. (2015). Penggunaan algoritma Dijkstra dalam perencanaan rute evakuasi bencana longsor di Kota Semarang. *Geoplanning: Journal of Geomatics and Planning*, 2(2), 93–102. <https://doi.org/10.14710/geoplanning.2.2.93-102>
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw Hill.
- Unhelkar, B. (2017). *Software engineering with UML*. Auerbach Publications. <https://doi.org/10.1201/9781351235181>
- Wang, J., Ma, W., & Yang, X. (2013). Development of degree-of-priority based control strategy for emergency vehicle preemption operation. *Discrete Dynamics in Nature and Society*, 2013, Article 283207. <https://doi.org/10.1155/2013/283207>
- Yulia, A., & Triase, T. (2024). Sistem informasi geografis pemetaan dan pencarian lokasi terdekat tempat sampah sementara di Kota Medan menggunakan algoritma Dijkstra. *Jurnal Algoritma*, 21(2), 252–263. <https://doi.org/10.33364/algoritma/v.21-2.2139>

Setiawan, N. B., Muhammad, A. H., & Yuniansyah. (n.d.). Aplikasi metode Dijkstra pada sistem pengambilan keputusan pemilihan jalur transportasi menggunakan teknologi GIS.

