Jurnal Aplikasi Sains Data
Journal homepage: https://jasid.upnjatim.ac.id/index.php/jasid

Dijkstra's Algorithm for Optimizing Humanitarian Aid Distribution Routes to Flood Victims in Cerme District, Gresik

Erika Fatimatul Hidayanti[1], Desi Tristianti[2], Datia Putri Nabila Br Tarigan[3], Dwi Arman Prasetya[4], Tresna Maulana Fahrudin[5]

[1,2,3,4] Department of Data Science, UPN "Veteran" Jawa Timur, Indonesia

[5] Department of Information and Communication Systems , Okayama University, Okayama 700-8530, Japan

[1]22083010081@student.upnjatim.ac.il, [2]22083010037@student.upnjatim.ac.id, [3]22083010049@student.upnjatim.ac.id, [4]arman.prasetya.sada@upnjatim.ac.id), [5]tresnamf@s.okayama-u.ac.jp

ABSTRACT

This study presents the development and analysis of a system designed to optimize the distribution routes of social aid during flood emergencies in the Cerme District, Gresik Regency. The primary objective is to ensure that logistical operations, particularly the delivery of aid to affected villages, are carried out in the most efficient and timely manner. To achieve this, Dijkstra's Algorithm is employed due to its well-established reliability in computing the shortest path between nodes in a weighted graph. The graph used in this research is constructed based on real-world spatial data, with each node representing a village and the edges representing actual road distances obtained from mapping services. The system is implemented using an Object-Oriented Programming (OOP) paradigm in Python, which ensures modularity and scalability of the codebase. For graph modeling and shortest path computation, the NetworkX library is utilized, while the graphical user interface (GUI) is built using Tkinter to provide an interactive and user-friendly experience. The application enables users to select starting and destination points from dropdown menus, compute the shortest route dynamically, and visualize it on an interactive graph complete with route details and distances. Experimental trials were conducted by simulating various flood scenarios, and the results demonstrated that the system successfully identified optimal aid routes with minimized travel distances. These outcomes confirm the practicality and effectiveness of the proposed method. Moreover, the ability to update the graph dynamically allows the system to adapt to changes in road accessibility due to flooding. This makes the tool highly applicable in real-world disaster response scenarios. In conclusion, the developed application offers a valuable solution for both local government agencies and humanitarian volunteers, helping to improve coordination, reduce delivery time, and ensure that aid reaches flood-affected communities as efficiently as possible.

Keywords: Dijkstra's Algorithm, Shortest Route, Flood, Gresik

I. INTRODUCTION

Indonesia, an archipelago, is situated in the tropics between 95° and 141° East Longitude and 6° North and 11° South Latitude. Its location, bisected by the equator, results in high rainfall intensity, year-round sun exposure, seasonal transitions, and vulnerability to natural phenomena that increase disaster potential. Disasters, whether natural or human-induced, can lead to property damage, environmental degradation, and loss of life. These events generally pose threats and disruptions to the survival of humans and other living beings [1].

* *Corresponding author.*
E-mail address: arman.prasetya.sada@upnjatim.ac.id
10.33005/jasid.v2i01.32

According to data released by the Indonesian Geoportal Data Bencana Indonesia, from January 1 to May 26, 2025, there have been 779 flood incidents across various regions in Indonesia. A report from the East Java Provincial Disaster Management Agency (Pusdalops BPBD) indicates that 15 regencies in East Java experienced flooding. These include Madiun, Nganjuk, Ngawi, Magetan, Sidoarjo, Kediri, Bojonegoro, Tuban, Probolinggo, Gresik, Pacitan, Trenggalek, Ponorogo, Lamongan, and Blitar.

Gresik Regency is identified as a flood-prone area, as detailed in the East Java Central Bureau of Statistics tables. Within Gresik, Cerme District specifically exhibits a notably high flood risk. This assessment was confirmed by the Secretary of the Gresik Regency Disaster Management Agency (BPBD), who stated that Cerme District is categorized as an area with significant flood vulnerability. This conclusion is based on direct observations conducted by the Gresik Regency BPBD [2].

When floods strike, accessibility becomes severely limited, impeding efforts to deliver social aid (bansos) to affected residents. Poorly planned aid distribution can lead to delays, inequitable allocation, and even difficulty reaching the most critical locations. To address these issues, an efficient and rapid strategy for determining bansos distribution routes is crucial, especially during flood emergencies. Dijkstra's Algorithm is a widely used shortest path algorithm for route optimization problems. This algorithm can find the path with the minimum weight between two points in a graph, making it highly relevant for determining aid distribution routes.

Dijkstra's Algorithm has been extensively applied in recent years to determine disaster evacuation routes. For instance, Nurfadila et al. [3] successfully utilized it to identify the closest tsunami evacuation paths for the South Jember Coast, pinpointing optimal routes from various affected areas. Similarly, Safutra et al. [4] employed Dijkstra's Algorithm for planning fire evacuation routes at RSUD I Lagaligo. Furthermore, Azizah and Kusumasari [5] implemented the algorithm to find the shortest routes to landslide locations.

This research aims to analyze the utilization of Dijkstra's Algorithm in determining social aid (bansos) distribution routes for flood victims in Cerme District, Gresik Regency. By applying this algorithm, we anticipate a simplified process for identifying the shortest routes, ultimately supporting the availability of an efficient application for use in emergency situations.

II. RESEARCH METHOD

In this study, the shortest route was determined through several structured stages designed in the form of a systematic workflow. Initial data was obtained through the use of Google Earth, where coordinates were marked for each village in the Cerme District. These points formed the basis for building a graph structure, which represented the geographical relationship between villages based on actual distance. Subsequent stages included preparing the graph data structure, creating and representing weighted graphs, applying the shortest path search algorithm, and visualizing the final results in a format that can be accessed and analyzed through an application or software. The following is the research workflow:

Figure 1. Research flowchart

A. Preparing Data Structures

The data structure used in this study is a graph data structure, which consists of nodes (vertices) and edges. Each node in the graph represents a location entity, namely a village, while the edges describe the direct relationships or paths between these villages. The graph used is categorized as an undirected weighted graph, where each edge has a weight value representing the distance between nodes, but the direction of the path is considered bidirectional (two villages can be directly reached without a specific direction). This structure was chosen because it can represent the real geographical relationships between villages in the Cerme subdistrict without specific directional restrictions, and allows for analysis of the shortest path based on the weight of the distance between nodes. The graph will be created based on the representation of the following map.

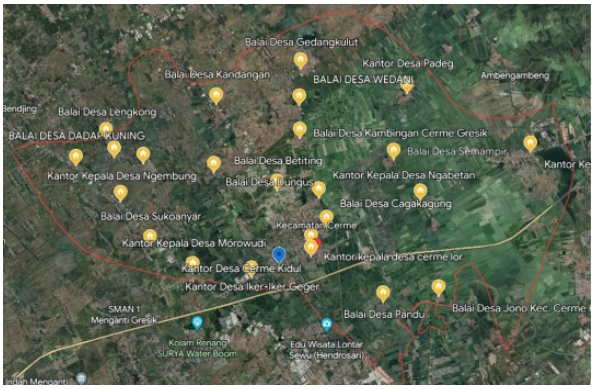

Figure 2. Map of Villages in Cerme Sub-district

B. Create and Represent Graphs

After the graph structure is determined, the next step is to represent the graph in a form that can be processed computationally. In this study, an adjacency list representation is used, in which each node has a list of other nodes that are directly connected to it, along with a weight (distance) value for each connection. The adjacency list representation was chosen because it is efficient for storing and accessing graph information that is sparse (not fully connected). The data used in the adjacency list is organized based on the Cerme subdistrict map, which has been converted into a graph as shown in Figure 3. In the graph, each village is labeled with a letter and connected by lines indicating the routes between villages along with distance information in kilometers. This adjacency list is also used as input in programming to support graph visualization and shortest path search using the Dijkstra algorithm.

This representation allows the system to calculate and determine the path from the 'Center' point to the destination village, by first finding the nearest points based on distance, then constructing the shortest route through the available paths.

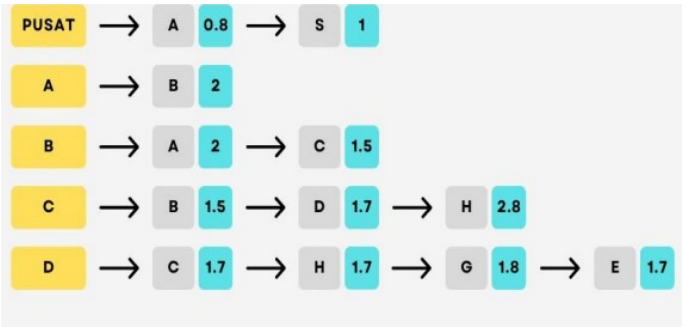

Figure 3. Pieces of undirected weighted graph representation using adjacency list

C. Finding the Shortest Route

The calculation and distance determination process begins with the input of a 'Center' point and a destination point. The system then proceeds to search for the nearest point and calculate the distance between points until it reaches the destination point. After the distance data between points is collected and represented in the form of a weighted graph, the shortest route is determined by applying the Dijkstra Algorithm. This algorithm is used because of its ability to efficiently and accurately solve the shortest path problem on weighted graphs, provided that the graph weights are non-negative. The process begins by initializing the distances of all nodes from the starting point (center) to infinity, except for the center node, which is set to zero. Each node is considered unvisited, and an iterative process is performed to find the node with the minimum distance that has not yet been visited.

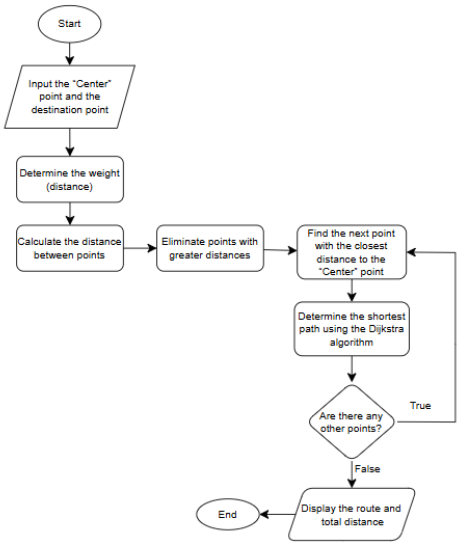

Figure 4. Flowchart of the shortest distance determination process

At each iteration, the node with the smallest distance value is selected for inspection, and the distances to all its neighboring nodes are updated if a shorter path through that node is found. After all neighboring nodes have been inspected, the node is marked as visited to prevent it from being inspected again. This process continues until all nodes have been processed or until the destination node is reached. After the calculation is complete, the shortest path connecting the center point to the destination point is determined through a backtracking process. This search is performed by following the trail of previously recorded nodes during the algorithm process, starting from the destination node and returning to the initial node. The result of this process is the sequence of nodes forming the shortest route along with the total distance traveled. This information can then be used as a basis for route decision-making, whether in the context of navigation systems, transportation management, or logistics planning.

D. Visualize Result with Apps

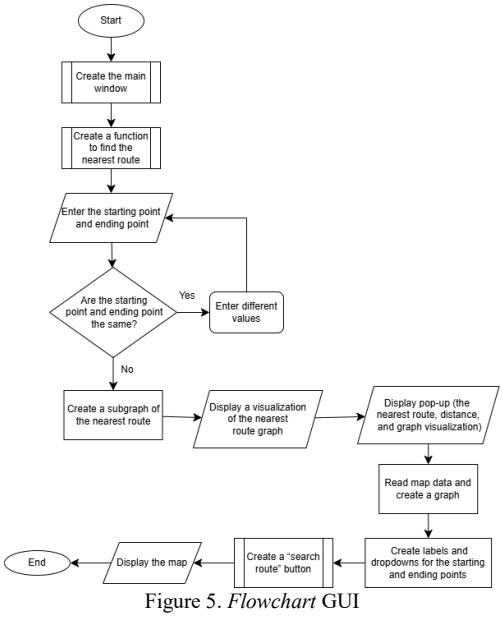

Figure 5. *Flowchart* GUI

The visualization of the short route calculation results is displayed by developing a GUI with the creation of the main application window, followed by the implementation of logic functions that run the route search algorithm. This interface allows users to select the starting point and end point through the dropdown components provided. The system then validates the input. If the starting and ending points are the same, the user is prompted to re-enter different points. After successful validation, the

system constructs a subgraph based on the shortest route and visually displays it through an undirected weighted graph.

The graph visualization is accompanied by interactive information in the form of pop-ups displaying details such as distance, points passed through, and total route length. This process is supported by spatial data reading and a graph constructed beforehand from Google Earth data, where each point represents the location of a village within the Cerme subdistrict.The interface also provides a “find the nearest route” action button that triggers the visualization and recalculation process when the input changes. Finally, an interactive map is displayed to illustrate the optimal route, making it easier for users to understand the geographical representation of the route obtained.

III. RESULT AND DISCUSSION

In this study, a weighted graph was successfully constructed based on spatial data extracted from Google Earth and Google Maps. Each node in the graph represents a village in the Cerme sub-district, while the edges connecting the nodes represent the road distances in kilometers between villages. The edge weights are derived from the actual travel distances obtained via Google Maps, making this a weighted undirected graph model.

The visual representation of this graph is shown in Figure 4, where the distance unit is in kilometers. An interactive interface was developed to support route visualization, which includes pop-ups that provide essential information such as the total distance, route path, and specific nodes (villages) traversed. A dedicated "Find the Nearest Route" button enables users to recompute and re-visualize the optimal route dynamically whenever the input or destination changes. This interactive system helps users particularly logistics planners and disaster response coordinators understand the geographical context of the shortest paths.

The system uses Dijkstra’s algorithm to calculate the shortest route from the distribution center (Puskesmas) to the target village nodes. Several trials were conducted to evaluate the shortest paths to different destinations. The following are the results:

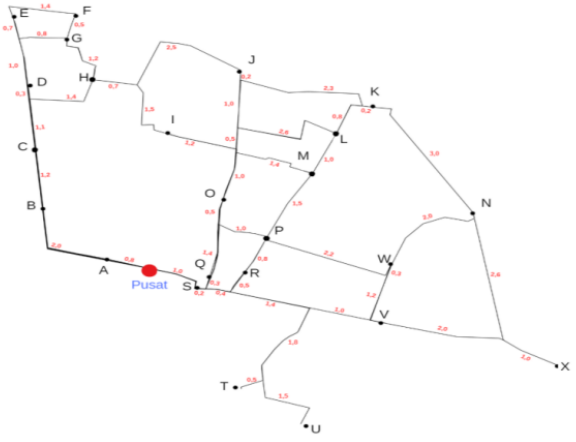

Figure 6. Representation Graph of Cerme Sub-district (Distance Unit in Kilometers)

Explanation of Selected Routes:

Point	Location
Center	Puskesmas
A	Desa Iker-Iker Geger
B	Desa Morowudi
C	Desa Sukoanyar

L	Desa Wedani
M	Desa Kambingan
N	Desa Padeg
O	Desa Betiting
P	Desa Ngabetan

D	Desa Ngembung	Q	Desa Cerme Lor
E	Desa Dadap Kuning	R	Desa Cagak Agung
F	Desa Lengkong	S	Desa Cerme Kidul
G	Desa Dooro	T	Desa Pandu
H	Desa Dampaan	U	Desa Jono
I	Desa Dungus	V	Desa Tambak Beras
J	Desa Kandangan	W	Desa Semampir
K	Desa Gedangkulut	X	Desa Banjasari

Table 1. List of villages in Cerme sub-district

After conducting several trials, multiple shortest route options were identified for several villages from the central point.

Percobaan Ke-	Rute Tercepat	Jarak yang Ditempuh
1.	Pusat ke G (A,B,C,D,G)	7.8 km
2.	Pusat ke F (A,B,C,D,G,F)	8.3 km
3.	Pusat ke P (S,R,P)	2.9 km
4.	Pusat ke K (S,R,P,M,L,K)	6.4 km

Table 2. Shortest Route Search Results

a. Route from Puskesmas to Desa Dooro:
   Path: Puskesmas → Iker-Iker → Morowudi → Sukoanyar → Ngembung → Dooro
   Distance: 7.8 km
b. Route from Puskesmas to Desa Lengkong:
   Path: Puskesmas → Iker-Iker → Morowudi → Sukoanyar → Ngembung → Dooro → Lengkong
   Distance: 8.3 km
c. Route from Puskesmas to Desa Ngabetan:
   Path: Puskesmas → Cerme Kidul → Cagak Agung → Ngabetan
   Distance: 2.9 km
d. Route from Puskesmas to Desa Gedangkulut:
   Path: Puskesmas → Cerme Kidul → Cagak Agung → Ngabetan → Kambingan → Wedani → Gedangkulut
   Distance: 6.4 km

These findings confirm that the shortest route can be identified by selecting the neighboring node with the smallest edge weight iteratively until the destination is reached. The cumulative distance of the selected edges represents the total cost of the path. The use of a graph-based approach enables efficient computation and flexible route planning. Since the graph can be dynamically updated with real-time spatial data or adjusted weights (e.g., in case of road closures due to flooding), the system is adaptable for disaster response scenarios.

To facilitate user interaction and practical implementation, a graphical user interface (GUI) was developed as illustrated. This interface allows users to easily select the starting point (*Titik Awal*) and

destination (*Titik Akhir*) from dropdown menus. Upon pressing the "Cari Rute Terdekat" (Find Nearest Route) button, the application executes the shortest path algorithm and visually displays the optimal route on the map using node and edge representations.

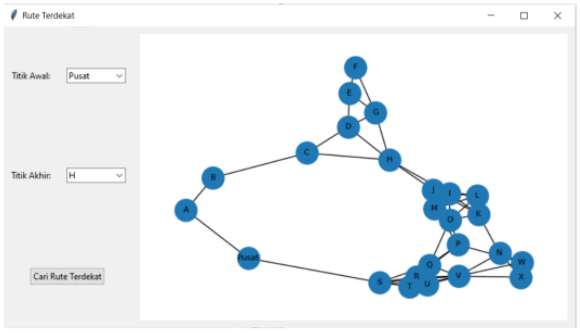

Figure 7. Nearest Route Application

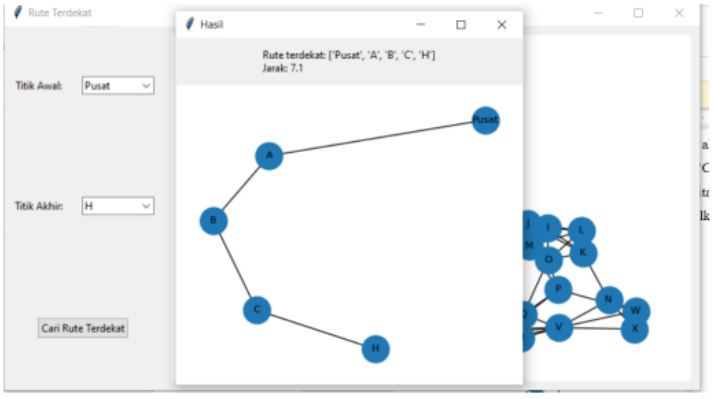

Figure 7. Nearest Route Application

The GUI also includes a pop-up result window that summarizes the computed route in a sequential list of village nodes, along with the total distance covered. This real-time visualization enhances the user experience by providing clear feedback and supporting rapid decision-making, particularly valuable during emergency logistics or disaster mitigation planning.

IV. CONCLUSION

This project successfully developed a system capable of determining the shortest delivery routes for distributing social aid during flood emergencies in Cerme Sub-district. The implementation of Dijkstra's algorithm proved effective in optimizing route selection from the central distribution point (Puskesmas) to various affected village offices. The system addresses the primary challenge in disaster logistics route efficiency by providing accurate and dynamic shortest path calculations based on real geographical data. Dijkstra's algorithm, by prioritizing nodes with the least cumulative distance, ensures that limited resources such as transportation and time are utilized optimally. Furthermore, the use of an object-oriented programming (OOP) paradigm for code implementation has enabled a well-structured and modular script, allowing for faster execution and easier maintenance. The combination of spatial data integration, graph theory, and interactive visualization enhances both the usability and practical value of this system for local governments and volunteers. This method has the potential to be scaled and adapted to other regions or types of emergencies, making it a valuable tool for disaster preparedness and response planning.

V. REFERENCES

[1] A. Fauzia, D. A. Pawestri, U. Wahrudin, S. Rahmawati, S. Himayah, and Nandi, "Analisis Penentuan Lokasi Evakuasi Bencana Banjir dengan Sistem Informasi Geografis dan Metode Simple Additive Wighting (Studi Kasus: Kecamatan Cileungsi)," *Jurnal Pendidikan Geografi Undiksha*, vol. 9, no. 2, 2021.

[2] Badan Nasional Penanggulangan Bencana, *Geoportal Data Bencana Indonesia*. [Online]. Available: https://gis.bnpb.go.id/. Accessed on: May 31, 2023.

[3] M. C. Bunaen, H. Pratiwi, and Y. F. Riti, "Penerapan Algoritma Dijkstra untuk Menentukan Rute Terpendek dari Pusat Kota Surabaya Ke Tempat Bersejarah," *Jurnal Teknologi dan Sistem Informasi Bisnis*, vol. 4, no. 1, 2022.

[4] E. Kusuma, Jefri, and H. Agung, "Aplikasi Perhitungan dan Visualisasi Jarak Terpendek Berdasarkan Data Coordinate dengan Algoritma Dijkstra dalam Kasus Pengantaran Barang Di Kawasan Jabodetabek," *Jurnal SISFOKOM*, vol. 8, no. 1, 2019. [5] K. D. Wijaya, *Penentuan Alternatif Rute Evakuasi Banjir Kecamatan Cerme Kabupaten Gresik*, Thesis, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.

[6] E. Ismantohadi and Iryanto, "Penerapan Algoritma Dijkstra untuk Penentuan Jalur Terbaik Evakuasi Tsunami – Studi Kasus: Kelurahan Sanur Bali," *Jurnal Teknologi Terapan*, vol. 4, no. 2, 2018.

[7] GeeksforGeeks, *Graph Data Structure and Algorithms*. [Online]. Available: https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/. Accessed on: May 22, 2023.

[8] W. Bismi, W. Gata, A. Anton, and T. Asra, "Penerapan Algoritma Hybrid Dalam Menentukan Rute Terpendek Antara Cabang Kampus," *Jurnal Sistem Komputer*, vol. 13, no. 1, 2020.

[9] Efanntyo and A. R. Mitra, "Perancangan Aplikasi Sistem Pengenalan Wajah dengan Metode Convolutional Neural Network (CNN) untuk Pencatatan Kehadiran Karyawan," *Jurnal Instrumentasi dan Teknologi Informatika (JITI)*, vol. 3, no. 1, 2021.

[10] M. Romzi and B. Kurniawan, "Pembelajaran Pemrograman Python dengan Pendekatan Logika Algoritma," *Jurnal Teknik Informatika Mahakarya (JTIM)*, vol. 3, no. 2, 2020.

[11] R. M. R. Clinton and R. Sengkey, "Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini-Komputer Raspberry Pi," *Jurnal Teknik Elektro dan Komputer*, vol. 8, no. 3, 2019.

[12] B. Folaimam, Rosihan, and A. Mubarak, "Implementasi Algoritma Dijkstra untuk Penentuan Jalur Terpendek pada Aplikasi Evakuasi Bencana untuk Penyandang Disabilitas," *Jurnal Informatika dan Komputer (JIKO)*, vol. 2, no. 2, 2018.