International Journal
of Applied Information Technology

http://journals.telkomuniversity.ac.id/ijait/

Android-Based Geolocation Technology on a Blood Donation System (BDS) Using the Dijkstra Algorithm

Syaiful Ahdan [a, *], Setiawansyah [b]

[a,b] *Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia*
syaifulahdan@teknokrat.ac.id, setiawansyah@teknokrat.ac.id

A R T I C L E I N F O

Received October 14th, 2020
Revised dd January 5th, 2021
Accepted January 5th, 2021
Available online July 5th, 2021

Keywords
Blood Donation System, Geolocation Technology, Android Geolocation

A B S T R A C T

Blood transfusion is an activity that is often carried out in order to save human life. The availability of appropriate bloodstock is crucial for use during special medical conditions. Therefore, the blood collection center must ensure availability; therefore, it needs to recruit active blood donors, difficult patients to find blood stocks, which will be discussed in this study. The contribution of this research is to design an information system that can find blood donors in the Bandar Lampung area with geolocation technology and use the Dijkstra algorithm to determine the closest route, with a system developed on the Android platform. The system can recommend donors to patients who need blood according to the patient's qualifications. Results obtained are an android-based mobile application that can search for geolocation-based blood donors using the Dijkstra algorithm, which can map the location of the nearest blood donor based on the patient's location. Based on data obtained from the results of system testing on functionality and usability, the system can map the location of blood donors and provide blood donor recommendations for patients in need. System testing is carried out using the ISO 9126 standard using the usability aspect, which consists of four test characteristics: the understanding aspect, the learning aspect, the ability aspect, and the operability aspect, and the attractiveness aspect.

Acknowledgment

Thanks to Universitas Teknokrat Indonesia for providing facilities in the implementation of this research.

* Corresponding author at:
Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia
Jl. ZA Pagaralam No. 9-11, Labuhan Ratu, Kedaton, Kota Bandar Lampung, 35132
Indonesia
E-mail address: syaifulahdan@teknokrat.ac.id

ORCID ID:
First Author: 0000-0002-4696-4230

https://doi.org/10.25124/ijait.v5i01.3317
Paper_reg_number IJAIT000050101 2021

1. Introduction

One of the main activities carried out by paramedics is blood transfusion and providing services for related products. Blood transfusion activities are carried out when the treatment process is in a serious condition, for example, surgical activities [1]. Blood transfusion activities are often carried out in order to save human life. Hospitals and emergency clinics must always have direct access to blood needs, and appropriate bloodstock must be available for use in times of special medical conditions.[2], [3].

The availability of appropriate bloodstock is crucial for use during special medical conditions. Therefore, the blood collection center must ensure availability. In addition, it needs to recruit active blood donors. Previous studies have stated that it is essential to understand several factors that can motivate potential blood donors to donate blood. Therefore it is necessary to conduct socialization to recruit prospective donors and determine the appropriate target [4].

Blood donation and transfusion service centers are among the most complex management systems in the health sector. Management of service quality in blood transfusion services (BTS) starts from a safe blood donor recruitment (BDR) system. Previous studies [5] discussed the recruitment process for blood donors. It can be concluded that the recruitment process for blood donors based on cellphones and geolocation aims to facilitate the information retrieval process and a management system that can guarantee the quality and quality and increase efficiency in operational management. Cellular phones can also be implemented as health care applications used to save lives and provide more user comfort. A blood donation system (BDS) is an information system that allows donors and blood donation service centers to communicate and coordinate efficiently. The blood donation process does not take much time and effort for both prospective donors and medical staff. Blood shortages can be seen in advertisements on social networks about searching for blood stocks for patients' urgent blood transfusion needs.

Therefore, blood transfusion centers and blood donors are urgently needed [6]. In previous studies [7], it has been suggested that blood donation services are a complex process and time-consuming to find multiple donors with blood type compatibility with the patient. Therefore developing an Android-based blood donor application can be a solution used to build connections between applicants and donors anytime and anywhere, which aims to help applicants broadcast messages throughout the voluntary blood donor network managed by the application.

Our research aims to solve the problem of searching for blood donors in the city of Lampung by using a search model based on the closest location of blood seekers. The blood donor search system is developed using geolocation technology and mobile computing. The community can utilize the proposed system for finding blood donors that will be developed by installing it on a smartphone so that the wider community can use it if they want to contribute to providing blood to patients.

2. Research Model

Research is supported by geolocation technology integrated with smartphone devices so that applications that involve functions such as remote monitoring and tracking can be designed based on geolocation. The proposed research is a blood donor search system used on smartphones with Android applications by utilizing geolocation technology and adding search optimization using an algorithm.

2.1. Geolocation Technology

A geographical information system is a system used to enter, store, recall, process, analyze and generate geographic or geospatial reference data, to support decision making in planning [8]. The latest generation of wireless communication network technology and smartphones impacts easy access anytime and anywhere. With easy access, users can communicate with each other, set up their network, and share geolocation information using global position system (GPS) technology. Geolocation technology is also used in frameGeoSocial, which was developed for case studies to create social networks of friends that aim to share information about places already visited [9].

Currently, geolocation-based applications have been widely developed according to the context of the problem and the objectives of system development to provide more added value from an application [10]. Previous research has developed an application using geolocation technology, namely an Android-based heart rate monitoring application by sending messages in the form of alarm sounds through notifications such as SMS and electronic mail. With geolocation technology, patient health monitoring can be monitored in real-time [11]. Previous research has developed an application using geolocation technology, namely an Android-based heart rate monitoring application, by sending messages in the form of alarm sounds through notifications such as SMS and electronic mail.

2.2. Android

Android is a software stack for mobile devices that includes the main operating system, middleware, and applications. The Android SDK provides the tools and APIs needed for application development on an Android-based platform using the Java programming language [12].

2.3. Dijkstra

Dijkstra's algorithm, proposed in 1959, is a graphical search algorithm that can be used to solve the shortest path problem. The shortest path algorithm focuses on the route length parameter and calculates the shortest route and the fastest path algorithm focuses on the path with the fastest travel time [13][14]. Dijkstra's algorithm is widely used for the development of geolocation-based applications. One of the studies in previous research [15] suggests that Dijkstra's algorithm was once used for finding food ingredients by providing the shortest path to users by using the greedy approach to solve problems.

2.4. Dijkstra's Algorithm in the Blood Donor Search System

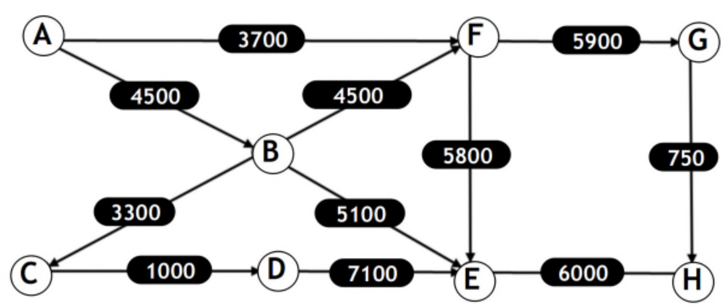

Figure 1 Search Route for Blood Donors

Topology (Figure 1), the distance value at each connected node is obtained based on the distance between the nodes' coordinates based on measurements obtained from Google Maps. Each distance is measured in square meters m^2. Calculations of Distance and Coordinates can be seen in Table 1 and Table 2.

Table 1 Node Description

Node	Description	Coordinate
A	Jl. Teuku Umar	5°23'44.6"S 105°15'41.7"E
B	Jl. ZA Pagaralam	5°22'25.0"S 105°14'29.3"E
C	Jl. Sultan Haji	5°22'33.2"S 105°15'57.4"E
D	Jl. Kota Sepang	5°22'09.8"S 105°16'08.6"E
E	Jl. Imam Bonjol	5°23'39.6"S 105°14'00.7"E
F	Jl. Urip Sumoharjo	5°23'24.0"S 105°16'29.8"E
G	Jl. Anggrek	5°25'08.3"S 105°15'54.2"E
H	H: Jl.Melati	5°25'27.8"S 105°15'54.6"E

Table 2 Distance Between Nodes

		Connected to Node		
Node	Coordinate	Node	Coordinate	Distance
A	5°23'44.6"S 105°15'41.7"E	F	5°23'24.0"S 105°16'29.8"E	$3700m^2$
		B	5°22'25.0"S 105°14'29.3"E	$4500m^2$
B	5°22'25.0"S 105°14'29.3"E	A	5°23'44.6"S 105°15'41.7"E	$4500m^2$
		F	5°23'24.0"S 105°16'29.8"E	$4500m^2$
		C	5°22'33.2"S 105°15'57.4"E	$3300m^2$
		E	5°23'39.6"S 105°14'00.7"E	$5100m^2$
C	5°22'33.2"S 105°15'57.4"E	B	5°22'25.0"S 105°14'29.3"E	$3300m^2$
		D	5°22'09.8"S 105°16'08.6"E	$1000m^2$
D	5°22'09.8"S 105°16'08.6"E	C	5°22'33.2"S 105°15'57.4"E	$1000m^2$
		E	5°23'39.6"S 105°14'00.7"E	$7100m^2$
E	5°23'39.6"S 105°14'00.7"E	B	5°22'25.0"S 105°14'29.3"E	$5100m^2$
		F	5°23'24.0"S 105°16'29.8"E	$5800m^2$
		H	5°25'27.8"S 105°15'54.6"E	$6000m^2$
F	5°23'24.0"S 105°16'29.8"E	A	5°23'44.6"S 105°15'41.7"E	$3700m^2$
		B	5°22'25.0"S 105°14'29.3"E	$4500m^2$
		E	5°23'39.6"S 105°14'00.7"E	$5800m^2$
		G	5°25'08.3"S 105°15'54.2"E	$5900m^2$
G	5°25'08.3"S 105°15'54.2"E	F	5°23'24.0"S 105°16'29.8"E	$5900m^2$
		H	5°25'27.8"S 105°15'54.6"E	$750m^2$
H	5°25'27.8"S 105°15'54.6"E	E	5°23'39.6"S 105°14'00.7"E	$6000m^2$
		G	5°25'08.3"S 105°15'54.2"E	$750m^2$

Step 1 starts from the initial node, namely A, the destination node E, each edge connected between nodes has been given a value. Dijkstra performs calculations on neighboring nodes that are directly connected to the departure node of node A. Table 3 describes Dijkstra's stage 1; node A is the initial route, and node E is the destination route. Dijkstra selects nodes that are directly connected to node A, namely, B and F. A is directly connected to B with a value of = 4500, to node F with a value of 3700. Only two nodes are directly connected to node A, and then stage 1 has been completed.

Table 3 Step 1 of Dijkstra's Node A

Iteration	V	Node: Min = (dist[node], prev[node]) iteration							
		a	b	c	d	e	f	g	h
		(0,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0
1	a	0	(4500,a)1	(∞,a)1	(∞,a)1	(∞,a)1	(3700,a)1	(∞,a)1	(∞,a)1

Step 2 starts from node F is placed in the initial position in step 2 because F has a shorter distance when compared to B. Table 4 describes Dijkstra's second step, node F as the initial route and node E as the destination route. Dijkstra selects nodes directly connected to node F, namely node B, node E, and node G. node F is directly connected to node B = 4500, node F = 5800. Node G = 5900, stage 2 has been completed, and nodes that are not directly connected with node F, namely: node H, node D, node C, the Dijkstra algorithm steps for step 2 can be seen in Table 4.

Table 4 Step 2 of Dijkstra's Node F

iteration	V	Node : Min = (dist[node], prev[node]) iteration							
		a	b	c	d	e	f	g	h
		(0,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0
2	f	(3700,a)1	(4500,a)1	(∞,a)1	(∞,a)1	(9500,f)2	(3700,a)1	(9600,f)2	(∞,a)1

Step 3 starts from node B with a value of 4500, and node B is placed in the initial position in step 3. Table 5 describes stage 3 Dijkstra, node B as the initial route, and node E as the destination route. The result is, Dijkstra selects nodes that are directly connected to node B, namely: node C, node F, and node E. node B are directly connected to node F = 3700. node B is directly connected to node E = 9500, and node B is directly connected to node F = 3700. node C = 7800 because only three nodes are directly connected to node B, then Dijkstra 3 has been completed, and nodes that are not directly connected to node B are node H, node D, and node G.

Table 5 Step 3 of Dijkstra's Node B

iteration	V	Node : Min = (dist[node], prev[node]) iteration							
		a	b	c	d	e	f	g	h
		(0,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0
3	b	(3700,a)1	(4500,a)1	(7800,b)2	(∞,a)1	(9500,f)2	(3700,a)1	(9600,f)2	(∞,a)1

Step 4 starts from node C because route (B-C) has a shorter distance = 3300 when compared to route (B-E) = 5100, and route (B-F) = 4500, so node C is placed in the initial position at stage 4. Table 6 describes stage 4 of Dijkstra, node C as the starting route and node E as the destination route. The results obtained are that the Dijkstra algorithm selects nodes directly connected to node C, namely node D, node D is directly connected to node C = 8800, and node B is directly connected to node C = 9500. Node B is directly connected to node C = 4500 because only two nodes are directly connected to node C, stage 4 has been completed, and nodes that are not directly connected to node C are node A, node F, node G, node H, node E.

Table 6 Step 4 of Dijkstra's Node C

iteration	V	Node : Min = (dist[node], prev[node]) iteration							
		a	b	c	d	e	f	g	h
		(0,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0
4	c	(3700,a)1	(4500,a)1	(7800,b)2	(8800,c)3	(9500,f)2	(3700,a)1	(9600,f)2	(∞,a)1

Step 5 starts from node D = 8800 (A-B-C) because D is the only node directly connected to node C, node D is the starting route, and node E is the destination route. The results of Dijkstra choose a node that is directly connected to node D, namely node E, node E is connected directly to node D with a value of = 15900 because there is only one node that is directly connected to node D, so step 5 is complete. Nodes that are not directly connected to it are: node A, B, node F, node G, node H, Dijkstra's steps for stage 5 can be seen in Table 7.

Table 7 Step 5 of Dijkstra's Node D

iteration	V	Node : Min = (dist[node], prev[node]) iteration							
		a	b	c	d	e	f	g	h
		(0,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0
5	d	(3700,a)1	(4500,a)1	(7800,b)2	(8800,c)3	(9500,f)2	(3700,a)1	(9600,f)2	(∞,a)1

Step 6 starts from node D is placed in the initial position in stage 6, and node E is the destination node because there is only one destination node. Stage 6 has been completed, and stage 6 is the final stage of finding the closest route with a value of = 9500. Dijkstra has not been completed calculating all existing nodes. Some of the nodes that have not been calculated are node G and node H. Dijkstra will recalculate in step 7 and stage 8. Dijkstra's step for stage 6 can be seen in Table 8.

Table 8 Step 6 of Dijkstra's Node E

iteration	V	Node : Min = (dist[node], prev[node]) iteration							
		a	b	c	d	e	f	g	h
		(0,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0
6	e	(3700,a)1	(4500,a)1	(7800,b)2	8800,c)3	9500,f)2	(9500,f)2	(9600,f)2	(16350,h)4

Step 7 starts calculating the route, in the previous stage, node G and node E. node G is i placed from several nodes that are not yet connected directly to node H. which is directly connected to node G, namely H = 10350, Dijkstra's step for stage 7 can be seen in Table 9.

Table 9 Step 7 of Dijkstra's Node G

iteration	V	Node : Min = (dist[node], prev[node]) iteration							
		a	b	c	d	e	f	g	h
		(0,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0
7	g	(3700,a)1	(4500,a)1	(7800,b)2	8800,c)3	(9500,f)2	(9500,f)2	(9600,f)2	(10350,h)3

Step 8 starts with several nodes for which route calculations have not been carried out in the previous stage, and namely, node H is placed in a position directly connected to node E. The result is Dijkstra choosing a node that is directly connected to H, namely E = 16350. The value obtained is not the best value because the best value is 9500 by route (A-F-E). Dijkstra in the graph (Figure 1) can be written the following pseudo-code and in detail can be seen in Figure 2. Dijkstra's step for stage 8 can be seen in Table 10.

Table 10 Step 8 of Dijkstra's Node H

Iteration	V	Node : Min = (dist[node], prev[node]) iteration							
		a	b	c	d	e	f	g	h
		(0,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0	(∞,-)0
8	h	(3700,a)1	(4500,a)1	(7800,b)2	8800,c)3	(9500,f)2	(9500,f)2	(9600,f)2	(10350,h)3

```
BEGIN
  d(v[1]) ← 0
  FOR i = 2,..,n DO
    d(v[i]) ← ∞, parent(v[i]) ← NULL
  WHILE queue ≠ Ø DO
    u = queue.extractMin()
    FOR ALL (u,w) ∈ E DO
      dist ← d(u) + l(u,w)
      IF w ∈ queue AND d(w) > dist DO
        d(w) = dist, parent(w) = (u,w)
      ELSE IF parent(w) == NULL THEN
        d(w) = dist, parent(w) = (u,w)
        queue.insert(w,dist)
END
```

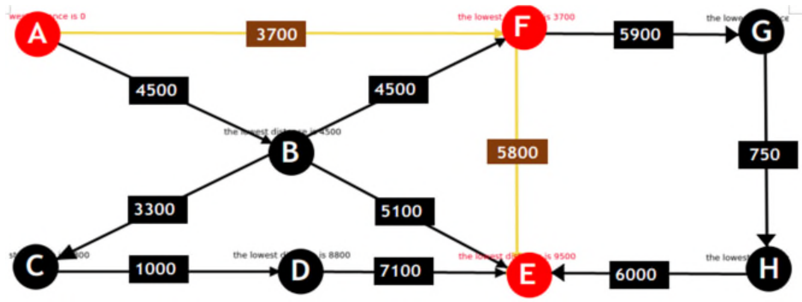

Figure 2 Shortest Path Search using Dijkstra's Algorithm

Adjacency matrix (Figure 3) is a matrix which states the existence of an edge that connects the source vertex to the destination vertex. We are currently using the case, let A = (B, F) be a graph with n vertices, n ≥ 1. The adjacency matrix G is a two-dimensional matrix of n x n size. If the matrix is called A = [aij], then aij = 1 if vertices i and j are neighbors, otherwise aij = 0 if vertices i and j are not neighbors. In other words, if there is an edge connecting the original vertex and the destination vertex, then the matrix will be filled with a value of 1 and vice versa. If there is no edge connecting the original vertex and destination vertex, the matrix will be filled with a value of 0. A directed graph or graph has a direction, the row in the 30 adjacency matrix is represented as the origin vertex, and the column is represented as the destination vertex. The minimum distance can be seen in Figure 4.

$$\begin{pmatrix} 0 & 4500 & 0 & 0 & 0 & 3700 & 0 & 0 \\ 0 & 0 & 3300 & 0 & 5100 & 4500 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5800 & 0 & 5900 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 750 \\ 0 & 0 & 0 & 0 & 6000 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \infty & 4500 & 7800 & 8800 & 9500 & 3700 & 9600 & 10350 \\ \infty & \infty & 3300 & 4300 & 5100 & 4500 & 10400 & 11150 \\ \infty & \infty & \infty & 1000 & 8100 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 7100 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 5800 & \infty & 5900 & 6650 \\ \infty & \infty & \infty & \infty & 6750 & \infty & \infty & 750 \\ \infty & \infty & \infty & \infty & 6000 & \infty & \infty & \infty \end{pmatrix}$$

Figure 3 Adjacency Matrix

Figure 4 Minimum Distance

The incidence matrix in this case study represents the relationship between vertices and edges. Let G = (V, E) be a graph with n vertices and m edges. The incidence matrix G is a two-dimensional matrix measuring n x m. Rows show vertex labels, while columns show edge labels. If the matrix is called A = [a_{ij}], then a_{ij} = 1 if vertex i is incidence with edge j, otherwise a_{ij} = 0 if vertex i is not incidence with edge j. Side-by-side matrices can be used to represent graphs containing multiple edges. If the graph in Figure 1 is represented in the incidence matrix, a form of incidence matrix will be obtained, as shown in Figure 5.

$$\begin{pmatrix} 4500 & 3700 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4500 & 0 & 0 & 3300 & 0 & 0 & 5100 & 4500 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3300 & 1000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1000 & 7100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -7100 & -5100 & 0 & -5800 & 0 & -6000 \\ 0 & -3700 & 5900 & 0 & 0 & 0 & 0 & -4500 & 5800 & 0 & 0 \\ 0 & 0 & -5900 & 0 & 0 & 0 & 0 & 0 & 0 & 750 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -750 & 6000 \end{pmatrix}$$

Figure 5 Incidence Matrix

The incidence matrix of Figure 5 is intended to facilitate calculating the degrees of each vertex. A vertex of degree n means that there are n edges that connect it to other vertices. The incidence matrix is 1 if vertex i corresponds to edge j.

2.5. Determination of the Shortest Distance

Determination of the shortest distance in Figure 2 is determined using the Dijkstra algorithm and obtains the shortest route A - F - G, with a distance of 9500m^2. The calculation of the solution can be seen in Table 11 as follows:

Table 11 Step 7 of Dijkstra's Node G

Searching for a Route (A - E)	Distance (m^2)
A→ B	4500 m^2
A→ F	3700 m^2
A→ F→ E	9500 m^2
A→ F→ G	9600 m^2
A→ B→ C	7800 m^2
A→ B→ F	9000 m^2
A→ B→ E	9600 m^2
A→ B→ C→ D	8800 m^2
A→ F→ G→ H	10350m^2
A→ B→ C→ D→ E	15900m^2
A→ F→ G→ H→ E	16350m^2

Based on Table 11, the closest route to the user's position (A) to the intended blood donor (E) is 9500m2. The calculation of Dijkstra's algorithm in the geographic information system of blood donors is by searching the distance between two coordinate points, namely the user and the donor. Then calculating the closest distance search using the blood donor geographic information system that has used the Dijkstra algorithm.

3. System Design

System design is a stage for modeling and communicating the system using diagrams and text. The following is the design of the geographic information system interface for permanent blood donors in Bandar Lampung using the Dijkstra algorithm: a use-case diagram describing the expected functionality of a system (Figure 6 & Figure 7). A use case can represent the interaction between actors and the system. The following is an explanation of the geographic system use case diagram of the closest blood donor, namely (1) Donor seekers enter the application, (2) Admin can enter donor data, (3) Blood seekers can view donor location maps,

(4) Donor seekers can view donor location data, (5) Donor seekers can submit donor requests.

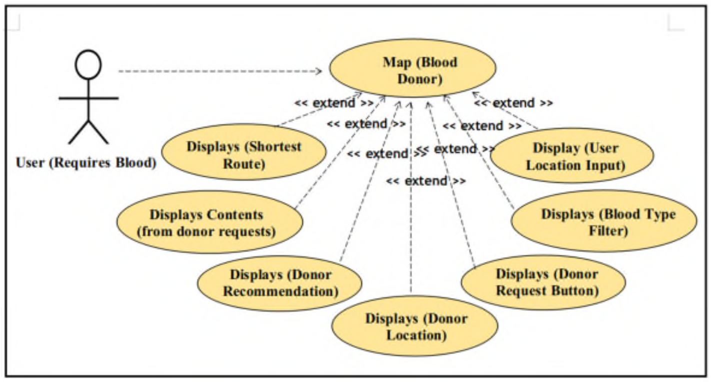

Figure 6 Use Case diagrams System (User)

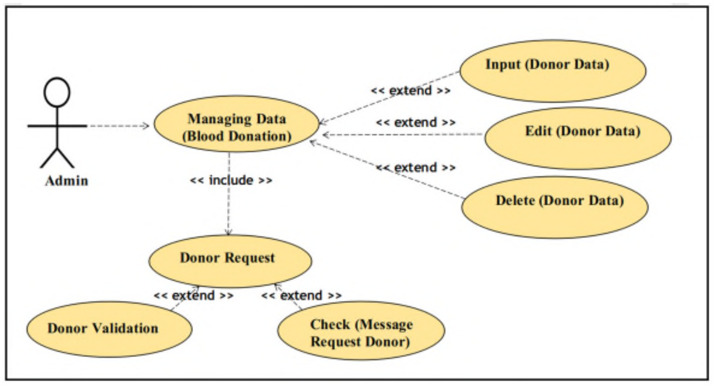

Figure 7 Use Case diagrams System (Admin)

Data obtained from PMI Bandar Lampung about blood donors is used as a basis for analysis to create a blood donor information system in Bandar Lampung using the Dijkstra algorithm. The result is a blood donor search system that can provide for the nearest permanent blood donor based on the request location. The implementation of the Dijkstra algorithm in the system is presented in the following explanation.

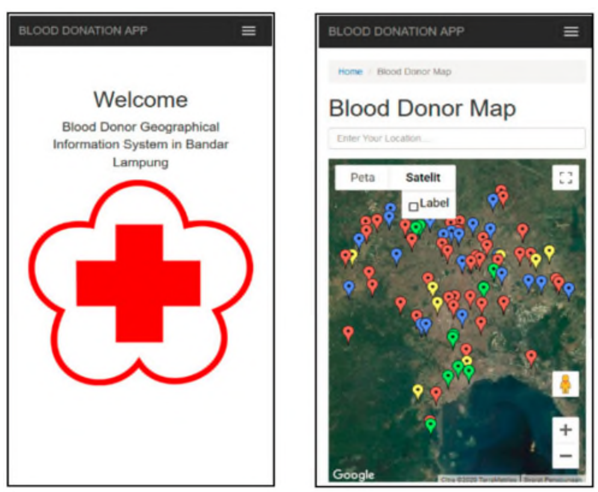

Figure 8 Main Menu and Display of Blood Donor Mapping

The main menu and application features can be used after the user logs in to the provided area (Figure 8). The user will provide the input username and password in the provided edit box and press the login button to run the authentication process. If the authentication process is correct, the user can use the available features. Figure 8 shows the location of the permanent donor mapping in Bandar Lampung using the Dijkstra Algorithm. All fixed blood groups that have been input through the system consisting of blood types A, B, AB, and O. Blood type A is marked in red, blood type B is marked in B, blood group AB is marked in yellow, and blood type O is marked in green. To find blood donors that match the required blood group, patients seeking blood donors must enter the search location in the provided feature. In addition, the system can also display recommendations from the closest donors. To make a blood donor request, the user can access the blood donation request section by selecting a coordinate point that matches the predetermined identity color. Then the user fills in the blood donor request form, such as inputting order data, contact information that can be contacted, address, and content message.

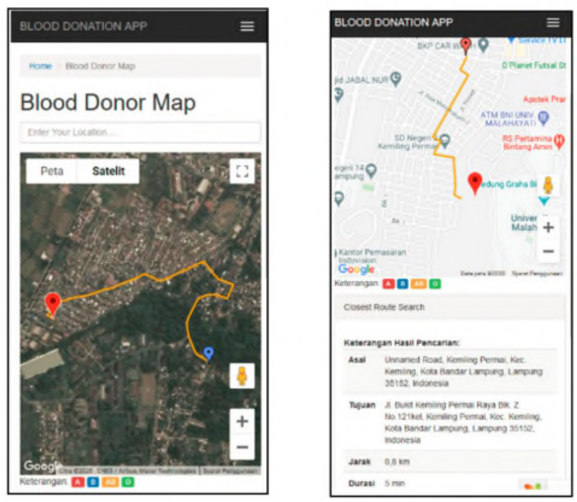

Figure 9 Main Menu and Display of Blood Donor Mapping

The blood donation request input from Figure 9 consists of a username (Require Blood), phone number, address of the customer, and the message's contents, which will be forwarded by pressing the submit button. After the patient fills in the form correctly and adequately, the PMI staff will notify the patient's blood requests.

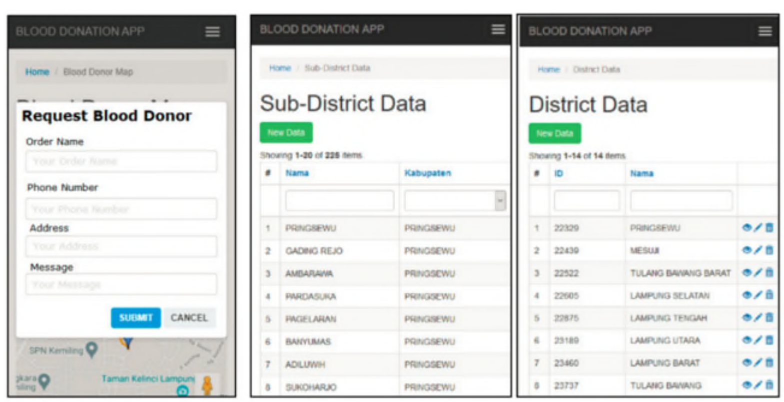

Figure 10 Display of Blood Donation Request

The display of donor request input can be seen in Figure 10, which the patient has ordered. The admin is responsible for validating request from donor seekers, can be seen in Figure 11.

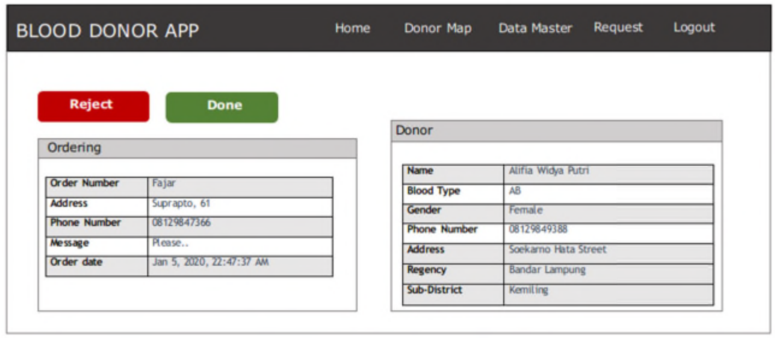

Figure 11 Display of Blood Donation Request Validation (Admin)

4. Testing

Ensuring quality in the software system requires monitoring and management and supports stringent standards. The software industry will check products and services that meet the requirements [16]. The International Organization for Standardization/ International Electronic Commission, or called ISO 9126. The testing is to minimize errors and inconsistencies system and suit what is expected by the user. Testing is carried out on a blood donor search system using ISO 9126. Testing will be carried out by giving a questionnaire to several respondents who function as users and some respondents who act as system administrators. Of the several respondents who filled out, the questionnaire will conduct a test that aims to test the quality of the blood donor search system. Respondents' answers to be processed are respondents who provide valid answers to the questionnaire. The following formula will measure respondents' responses to the quality of applications.

$$Result = \frac{Actual\ Score}{Ideal\ Score}\ x\ 100\%$$

The actual score is the answer of all respondents to the questionnaire that has been submitted. The ideal score is the highest score or weight or all respondents are assumed to choose the answer with the highest score (Table 12).

Table 12 Criteria for Percentage of Respondents

Total score (%)	Criteria
20,00% – 36,00%	Not Good
36,01% – 52,00%	Poor
52,01% – 68,00%	Enough
68,01% – 84,00%	Good
84,01% – 100%	Very Good

The testing on the information system is to search for blood donors using the ISO 9126 test standard, testing two of the six characteristics of ISO 9126, namely usability and functionality. The usability aspect aims to test the ease of using the system, and the functionality aspect aims to test the system's usability. The usability aspect test is carried out on the community (blood donor seekers). The respondents who were tested were five respondents. The respondents tried the geographic information system application of blood donors. Then the respondents filled out a questionnaire. The number of questions in the questionnaire consisted of 18 questions with four sub-aspects, namely understand-ability, learn-ability, operability, and attractiveness using a scale of Strongly Agree (SS) = 5, Agree (S) = 4, Doubt (R) = 3, Disagree (TS) = 2, Strongly Disagree (STS) = 1. The results of the Usability test can be seen in Figure 12.

Table 13 Usability Testing Result Data

Respondentt	understand-ability		Learn-ability					Operability			Attractiveness							
	Statement																	
	1	2	1	2	3	4	5	1	2	3	1	2	3	4	5	6	7	8
1	4	4	5	5	5	5	4	5	5	5	4	5	5	5	5	5	5	5
2	3	5	4	3	2	4	3	3	4	2	3	3	4	4	3	1	3	5
3	5	5	4	5	4	3	4	5	3	5	3	4	3	4	4	3	3	5
4	5	4	5	5	5	4	4	5	5	4	3	5	4	5	3	5	5	4
5	4	5	4	4	5	5	5	4	4	5	5	5	4	5	5	3	5	5

Table 14 Usability Testing Result Data Calculation

Respondent	*Understandability*	*Learnability*	*Operability*	*Attractiveness*
1	8	24	15	44
2	8	16	9	26
3	10	20	13	29
4	9	23	14	34
5	9	23	13	37
Actual Score	44	108	64	170
Ideal Score	50	125	75	200
Percentage	88 %	86.40%	85,33 %	85 %

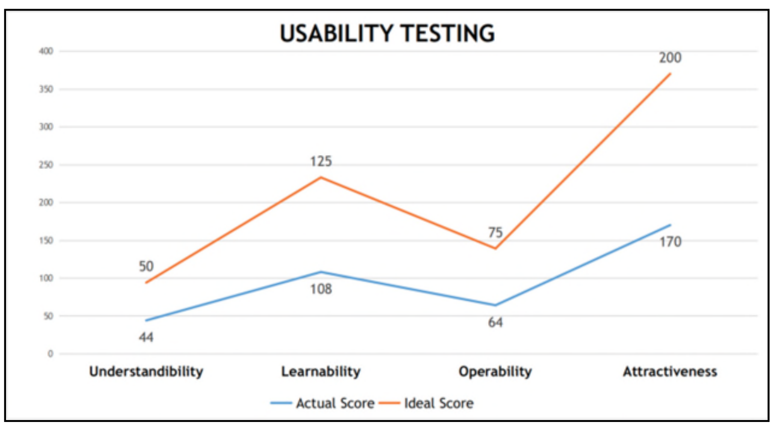

Figure 12 Usability Testing Score

After comparing with the range of score interpretation criteria, based on the graph obtained from the calculation of the actual score and the ideal score in Table 13 and Table 14, the feasibility of each sub-aspect of usability from the five respondents was as follows: understandability value of 44, learn ability value of 108, operability value of 64, attractiveness value of 170. The percentage for testing the overall usability aspects of the test data.

$$\% \, usability = \frac{Total\ Score\ of\ Test\ Results}{Highest\ Score}\ x\ 100\%$$

$$= \frac{386}{450}\ x\ 100\% = 85.78\%$$

Based on the data processing results on the test of the geographic information system of donor seekers using ISO 9126 on the usability aspect, it states that the software has a value of 88% understand-ability, 86.4% learn-ability, 85.33% operability, and 85% attractiveness. Overall percentage processing from the usability aspect of the geographic information system of blood donors still gets 85%. From the calculation of the usability percentage that has been done, as in Figure 13, it can be concluded that the geographic information system application of blood donors is declared very feasible to use.

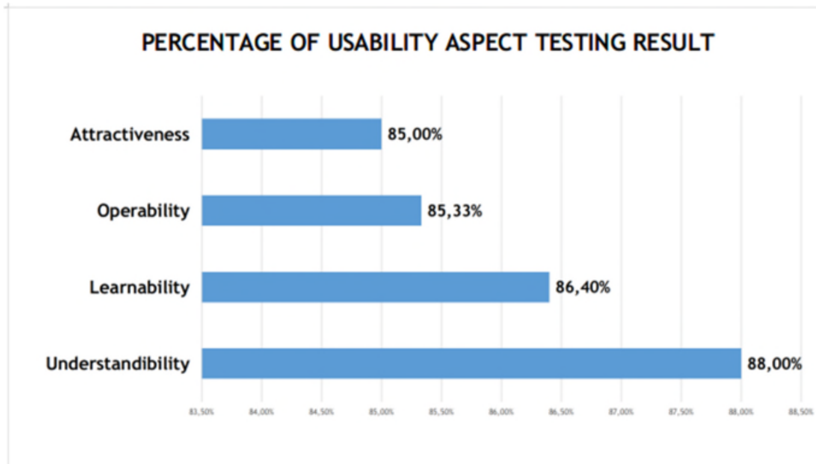

Figure 13 Percentage of Usability Test Results

In testing, the functionality of the questionnaire is filled in by examiners who have expertise in the field of software engineering to find out whether the functions in the application can run correctly (success or failure). The results can be seen in Table 15.

Table 15 Functionality Aspect Test Results

No	Statement	Result	
		Success	Failed
1.	Login menu	2	0
2.	Displays the Main Page	2	0
3.	Add blood donor button	2	0
4.	Save button on the menu	2	0
5.	Name input form	2	0
6.	Blood Type Form	2	0
7.	Gender Form	2	0
8.	Telephone Number Form	2	0
9.	Address Form	2	0
10.	Regency ID Form	2	0
11.	District Form	2	0
12.	Latitude and Longitude	2	0
13.	Display Pages	2	0
14.	Displays the Donor Map	2	0
15.	Display District / City	2	0
16.	Display the District	2	0
17	Displays the Donor	2	0
18	Displays the donor search	2	0
19	Showing a map of added	2	0
20	View button, update	2	0
21	Menu Request for Blood	2	0
22	Displays a Blood	2	0

No	Statement	Result	
		Success	Failed
23	Blood type A blood donor	2	0
24	Blood donor finder button	2	0
25	Blood group AB blood	2	0
26	Blood group O blood	2	0
27	All goal blood donor	2	0

Furthermore, the percentage calculation is done for testing aspects of functionality, which are as follows.

$$\% \, functionality = \frac{Total\ Score\ of\ Test\ Results}{Highest\ Score} \, x \, 100\%$$
$$= \frac{62}{62} \, x \, 100\% = 100\%$$

This study shows that in the aspect of geographic information system functionality, the search for blood donors in Bandar Lampung uses the Dijkstra algorithm to obtain a value of 100%, which means that the system can work properly and correctly. The geographic information system for permanent blood donors in Bandar Lampung using the Dijkstra algorithm was tested in the software quality test phase based on ISO 9126 (functionality and usability). Testing the geographic information system for permanent blood donors in Bandar Lampung using the Dijkstra algorithm is 100% Functionality and 85% Usability.

5. Conclusions

Based on the results of testing and analysis on the geographic information system of blood donors developed, system quality testing on the geographic information system of blood donors in Bandar Lampung includes functionality and usability aspects. Namely: the results of testing the functional aspects of the system can run well so that the software can be said to be valid. The test results obtained are presentations with usability aspects of 85.78% 100% functionality aspect. The geographic information system of searching for blood donors can be a media reference for obtaining regular blood donor information for Bandar Lampung.

Bibliography

[1] H. Javadzadeh Shahshahani, M. T. Yavari, M. Attar, and M. H. Ahmadiyeh, “Knowledge, attitude and practice study about blood donation in the urban population of Yazd, Iran, 2004,” *Transfus. Med.*, vol. 16, no. 6, pp. 403–409, 2006, DOI: 10.1111/j.1365-3148.2006.00699.x.

[2] Y. Aslani, S. Etemadyfar, and K. Noryan, “Nurses’ knowledge of blood transfusion in medical training centers of the Shahrekord University of Medical Science in 2004,” *Iran. J. Nurs. Midwifery Res.*, vol. 15, no. 3, pp. 141–144, 2010, [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/21589778.

[3] H. A. Majdabadi, M. Kahouei, S. Taslimi, and M. Langari, “Awareness of and attitude towards blood donation in students at the Semnan University of Medical Sciences,” *Electron. Physician*, vol. 10, no. 5, pp. 6821–6828, May 2018, DOI: 10.19082/6821.

[4] R. Finck, A. Ziman, M. Hoffman, M. Phan-Tang, and S. Yuan, “Motivating Factors and Potential Deterrents to Blood Donation in High School Aged Blood Donors,” *J. Blood Transfuse*, vol. 2016, p. 8624230, 2016, DOI: 10.1155/2016/8624230.

[5] M. S. Rahman, K. A. Akter, S. Hossain, A. Basak, and S. I. Ahmed, “Smart Blood Query: A Novel Mobile Phone-Based Privacy-Aware Blood Donor Recruitment and Management System for Developing Regions,” in *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, 2011, pp. 544–548, doi: 10.1109/WAINA.2011.115.

[6] A. M. Mostafa, A. E.Youssef, and G. Alshorbag, "A Framework for a Smart Social Blood Donation System Based on Mobile Cloud Computing," *Heal. Informatics - An Int. J.*, vol. 3, no. 4, pp. 1–10, 2014, DOI: 10.5121/hiij.2014.3401.

[7] M. Fahim, H. I. Cebe, J. Rasheed, and F. Kiani, "mHealth: Blood donation application using an android smartphone," in *2016 Sixth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, 2016, pp. 35–38, DOI: 10.1109/DICTAP.2016.7543997.

[8] H. Yanto, "Evakuasi, Optimalisasi Jalur Tsunami Menggunakan Sistem Informasi Geografis di kota Padang Berbasis Web," *J. Sains dan Inform.*, vol. 4, no. 12, pp. 193–202, 2018.

[9] T. M. Ramos, M. B. Damasceno, B. R. M. M. Holanda, and A. Araujo, "FrameGeoSocial mobile Ad-hoc social network based on geolocation," *2015 10th Iber. Conf. Inf. Syst. Technol. Cist. 2015*, 2015, DOI: 10.1109/CISTI.2015.7170441.

[10] A. Roxin *et al.*, "Survey of Wireless Geolocation Techniques To cite this version : A Survey," 2012.

[11] E. O. Tartan and C. Ciflikli, "An Android Application for Geolocation Based Health Monitoring, Consultancy and Alarm System," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018, vol. 02, pp. 341–344, DOI: 10.1109/COMPSAC.2018.10254.

[12] Khan, Jamil., Shahzad, Sara. (2015) : Android Architecture and Related Security Risk. Asian Journal of Technology & Management Research, Vol. 05 – Issue: 02 (Jun-Dec 2015), 14-18.

[13] N. Akpofure and N. Paul, "Anapplication of Dijkstra's Algorithm to shortest route problem," *IOSR J. Math.*, vol. 13, no. 1, pp. 20–32, 2017, DOI: 10.9790/5728-1303012032.

[14] J. C. Dela Cruz, G. V Magwili, J. P. E. Mundo, G. P. B. Gregorio, M. L. L. Lamoca, and J. A. Villaseñor, "Items-mapping and route optimization in a grocery store using Dijkstra's, Bellman-Ford and Floyd-Warshall Algorithms," in *2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 243–246, DOI: 10.1109/TENCON.2016.7847998

[15] N. Gupta, K. Mangla, A. Jha, and M. Umar, "Applying Dijkstra's Algorithm in Routing Process," *Int. J. New Technol. Res.*, vol. 2, no. 5, p. 263504, 2016.

[16] Abdelaziz, T., Elammari, M., Bani, W. (2015): Applying the ISO standard in Assessing the Quality of Software Systems. American Journal of Computer Science and Information Engineering, September 10, 2015, 2(3). 28-32.