Floyd-Warshall Application for the Shortest Route Search in a Traffic Accident Notification App

Haniah Mahmudah[1], M. Fajar Ibrahim[2], Okkie Puspitorini[3], Ari Wijayanti[4], Nur Adi Siswandari[5]

***Abstract*—In the case of an accident, a prompt response is needed to prevent accidents or provide assistance to traffic accident victims. Several ways can be carried out to address this problem. One of them is by developing several smartphone applications for accident detection and accident notification that provide aid in traffic accidents. The existing application to assist the victims merely presents the shortest route from the respondent to the victim's location without any features that help the respondent find a route to the nearest hospital and police station. Therefore, this study develops a smartphone application for accident alerts for victims' relatives and respondents, which assists in locating the victim's position and the closest hospitals and police stations. This accident notification app for smartphones utilizes open-source software and is very scalable. The outcome of this study is an Android application capable of sending accident notification broadcasts, allowing the victim's relatives and respondents to get accident notifications and drive to the accident place using the route given by the application. In addition, the developed app also provides information about the location of the nearest hospital and police station, allowing respondents nearby the location to help the victim promptly. The results of testing the application using the black box method on the Android platform indicated that 100% of the features of this application were running well. The shortest route with the Floyd-Warshall algorithm was 4.199 km, with no route deviations from the distance testing scenario. The average speed of notification delivery response from victims to respondents was 27.86 ms.**

***Keywords*—Traffic Accidents, Android Apps, Floyd-Warshall Algorithm, Black Box.**

I. INTRODUCTION

Population growth has led to an increase in the number of vehicles, ultimately leading to an increase in traffic accidents. A rapid response is needed in an accident to help the victims [1]. Reducing the number of people injured in traffic accidents requires measures to either lower the likelihood of accidents or help those involved in an accident.

Several measures can be taken to prevent traffic accidents, including providing explanations of safe driving practices and raising awareness of the importance of driver safety during a trip. Some accident prevention research has been conducted, for example, by monitoring the condition of drivers who drive safely or recklessly, such as moving, turning left, turning right, abrupt braking, and turning back at high speed. Meanwhile, research to help victims of traffic accidents includes the development of many smartphone apps for accident detection and accident notification to assist in traffic accidents [2].

Open-source software usage facilitates smartphone app development by utilizing previously installed sensors. Smartphones benefit from this because of their remarkable scalability. Some studies on smartphone applications to detect accidents include the detection of accidents in the form of hardware connected to several sensors that send notifications in the event of an accident. Another study developed an Android application that included the accident victim application and the respondent application to determine the accident location, allowing the victim to receive aid faster [3]. In addition, research that utilizes the internet of things (IoT) to detect accidents using GSM communication can transmit data from several sensors, including the MQ2 gas sensor and speed sensor using the Hall effect sensor [4]. However, the application to help accident victims only presents the shortest route from the respondent to the victim's location with no features that help the respondent find the route to the nearest hospital and police station [3], [4].

Several route determination studies used diverse websites and Android applications with various algorithm [1]‒[21]. One of utilized algorithms was the Floyd-Warshall. Dynamic programming through the calculation steps of the Floyd-Warshall algorithm has the ability of a quick and simple calculation process compared to prior studies on the search for the shortest route for the evacuation route of fire victims [5]. The results of the Floyd-Warshall algorithm research were able to provide the shortest route on each floor in a multi-story hotel building. In addition, there are studies using chart diagrams and applying the Floyd-Warshall algorithm in route weight calculation, resulting in an application that searches for the shortest route on tourist attractions [6]. The application of a futsal match finder using the Floyd-Warshall algorithm on the map has also been studied [7]. Furthermore, other studies have employed several algorithms, including A*star, Djiskstra, Bellman-Ford, and Floyd-Warshall, to determine the shortest route [8]-[10].

Helping accident victims as soon as possible is one method of reducing the number of fatalities among accident victims. If the accident victim is critically injured, immediate assistance is required, such as transporting the accident victim to the hospital or reporting to the nearest police station. It is vital to find the shortest route so that responders can promptly reach the victim's area and help accident victims. Dynamic programming and the shortest route determination algorithm can be utilized to determine the best route with the shortest time. In previous studies, the application merely indicated the accident victim's

[1,2,3,4,5] *Telecommunication Engineering Study Program, Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Jln. Raya ITS Keputih 60111 Surabaya (tel.: 031-5947280; fax: 031- 5946114; email: [1]haniah@pens.ac.id, [2]fajar@stutendt.pens.ac.id, [3]okkie@pens.ac.id, [4]ariw@pens.ac.id, [5]nuradi@pens.ac.id)*

[Received: 22 July 2021, Revision: 2 August 2022]

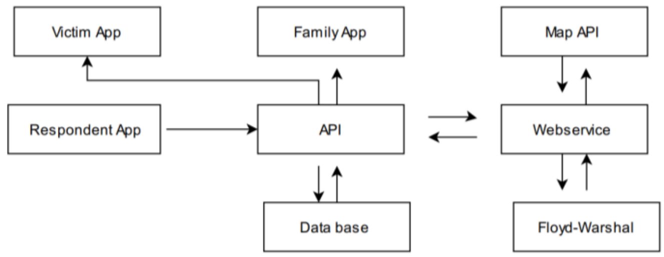

Fig. 1 Design of applications on smartphones.

location but did not provide directions to nearby medical facilities or police stations. Therefore, in this study, an application was developed for accident victims, their families, and respondents, which provided accident notifications by sending the accident location to the victim's family and several respondents near to the victim's location so that the victim's position and the shortest route to the victim's location could be determined. In this study, the shortest route was determined using the Floyd-Warshall algorithm because of its dynamic programming. Since the Floyd-Warshall algorithm can do calculations quickly and easily, it lends itself well to be applied on mobile devices. In the app, accident notification sent to the victim's family and respondents is in the form of information to search for the shortest route toward the accident location, nearest hospital and police station. The use of GPS coordinate points at the starting and ending locations is optimized to search for the shortest route. At the same time, the Floyd-Warshall algorithm was implemented in this research to determine the shortest route given a set of parameters, such as the starting and ending GPS coordinates, the distance between nodes, and the overall routes and distance between the starting and ending points. Application testing was carried out using a black box to find out the performance of the accident notification application.

II. Methodology

The development of this application was carried out using an object-oriented approach in the form of a waterfall method which included an identification phase, an analysis phase, a design phase, and a trial phase. The initial step was identifying and optimizing all alternative points for the shortest and fastest routes. The form, navigation structure, and Unified Modeling Language (UML), as well as the analysis and calculation of the optimized route traversed to reach the accident site, were planned in the analysis phase to achieve the best possible results. After the Floyd-Warshall algorithm obtained the optimal value, the application design stage was carried out. Fig. 1 depicts the application design phase, which includes applications for victims, families, and respondents.

The first phase depicted in Fig. 1 is when an accident triggers the sensors in the victim's smartphone application, which in turn, allows the victim's application to classify the accident. When the classification result indicates an accident, a warning will appear with a waiting time of 30 s. The respondent and the victim's family will be notified if the system does not get a response within the specified time frame. Notification of an accident is sent to the victim's family via the registered family

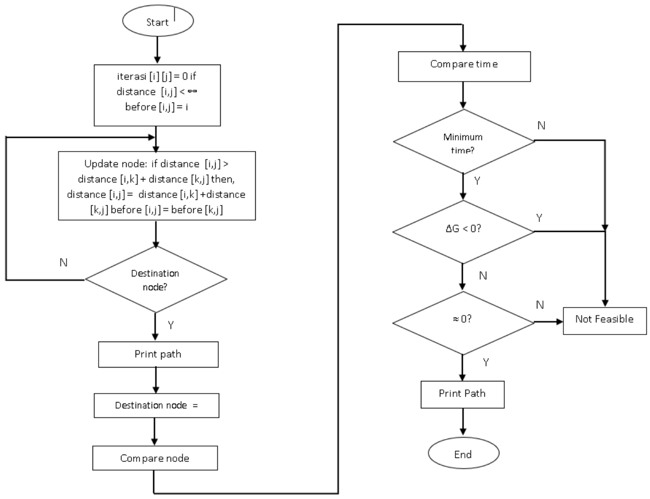

Fig. 2 Floyd-Warshall algorithm flow.

phone number. The sent notification is in the form of latitude and longitude locations and the victim's identity. The data were processed on web service for optimized route determination using the Floyd-Warshall algorithm and displayed on Maps API. The API is a bridge for data exchange between web service as a server and Android application as a client. The Floyd-Warshall algorithm calculation requires the input of the starting and the destination points in the form of latitude and longitude. Before determining the route from the respondent's location to the victim's location, the routing scenario, as in Fig. 1. was carried out.

The route and distance between linked nodes were derived from the scenario depicted in Fig. 1. The node was transformed into an $m \times m$ matrix to be used as input for the Floyd-Warshall algorithm in determining the optimized route. The flowchart of the Floyd-Warshall algorithm is shown in Fig. 2

Fig. 2 shows that the initial process was carried out by inserting the starting and destination points with distance values to $m \times m$ matrix. Subsequently, an iteration process was performed with as many nodes as possible. The result obtained from the last iteration was the shortest distance from the starting point to the destination. After obtaining the shortest distance, the mileage comparison process was carried out to obtain the most optimal route. The route was printed and displayed in the application if an optimal route had been obtained.

A. *Notification Broadcasts*

The process of sending notifications was carried out after the results of the accident classification stated the occurrence of an accident involving victims. The victim's application sent broadcast notifications to several respondents. Respondents receiving notifications were those within the radius of the victim's location. In contrast, respondents who were outside the radius would receive notifications through the parameters of the family phone number that the victim had registered. Calculation of the respondents' distance from the victim was carried out using the spherical law of cosines equation with the parameter of two points of longitude and latitude of the respondents' and victim's location and the earth's radius so that

Fig. 3 Test route scenarios.

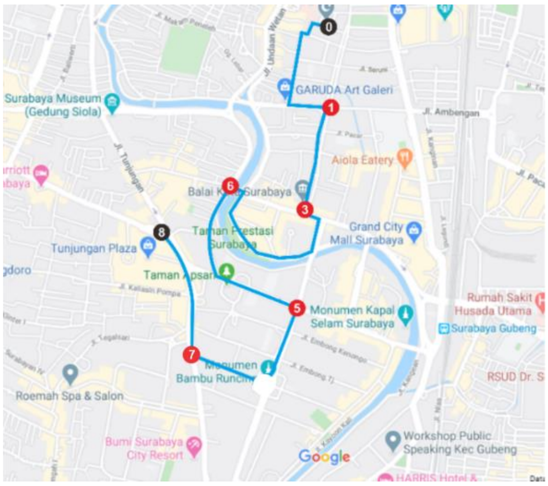

Fig. 4 First route (T0).

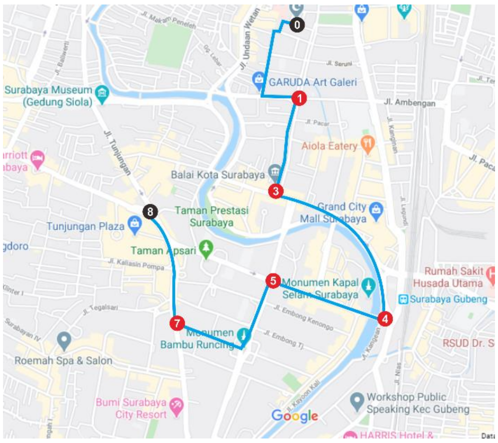

Fig. 5 Second route (T1).

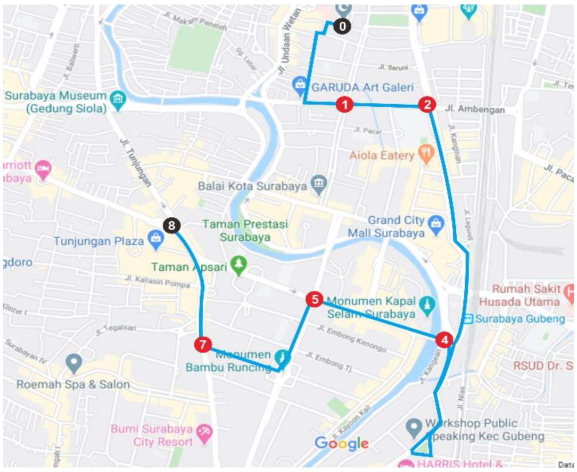

Fig. 6 Third route (T2).

the respondents' and the victim's distance was obtained. The formula for calculating the distance between the respondent and the victim's location using the spherical law of cosines equation is shown in (1).

$$d = a.\cos(\sin(A).\ \sin(B) + \cos(A).\cos(B).\cos(D\text{-}C)).R \quad (1)$$

where R is the earth's radius of 6,371 km, d is the distance in km, A is latitude 1, B is latitude 2, C is longitude 1, and D is longitude 2.

The broadcast notification was done by utilizing Firebase Cloud Messaging (FCM). FCM is a feature of Firebase. This feature can send real-time notifications and data to respondents' applications. There are two classes used in FCM, FirebaseInstanceIdService, and FirebaseMessagingService. The FirebaseInstanceIdService is used to generate a token from the FCM. *This token* is used as the identity of the respondent's device, that is, the recipient of the notification. On the other hand, FirebaseMessagingService is used to receive notifications and data sent by victims.

B. Floyd-Warshall Algorithm

The route search was carried out by dynamic calculations using PHP web programming with graph theory for the process of determining the shortest route in the application of the Floyd-Warshall algorithm. The shortest route test scenario based on distance is shown in Fig. 3, which is in the form of a map image located in central Surabaya.

It can be seen in Fig. 3 that the route that the respondent can traverse to the victim's location is as follows.

1) First Route: The first route is defined as T0. The node traversed by T0 is T0 = 0-1-3-6-5-7-8. The T0 route is shown in Fig. 4.

2) Second Route: The second route is defined as T1. The node traversed by T1 is T1 = 0-1-3-4-5-7-8. The T1 route is shown in Fig. 5.

3) Third Route: The third route is defined as T2. The node traversed by T2 is T2 = 0-1-2-4-5-7-8. The T2 route is shown in Fig. 6.

From the test scenario, as in Fig. 3, coordinate point data and distance between nodes with the starting point (T_0) and the destination point (T_k) are obtained, as in Table I. Subsequently, the data were converted into nodes, and the distance was displayed in the meter.

The calculation stage to determine the shortest route in the Floyd-Warshall algorithm employed (2). A matrix with rows is the initial location, while the column is the destination location.

$$x[i,j] \leq x[i,k] + x[k,j] \quad (2)$$

TABLE I
DISTANCE BETWEEN NODES

No.	T_0	T_k	Coordinate	Distance (m)
1	0	1	*{"nodes": ["0-1"], "coordinates": [[-7.25242724455...*	69.745
2	1	0	*{"nodes": ["1-0"], "coordinates": [[-7.25584599630...*	695.745
3	1	2	*{"nodes": ["1-2"], "coordinates": [[-7.25584599630...*	379.264
4	2	1	*{"nodes": ["2-1"], "coordinates": [[-7.25601213972...*	379.264
5	2	4	*{"nodes": ["2-4"], "coordinates": [[-7.25601213972...*	2,525.243
6	4	2	*{"nodes": ["4-2"], "coordinates": [[-7.26631210166...*	2,525.243
7	1	3	*{"nodes": ["1-3"], "coordinates": [[-7.25584599630...*	530.530
8	3	1	*{"nodes": ["3-1"], "coordinates": [[-7.26050714532...*	530.530
9	3	4	*{"nodes": ["3-4"], "coordinates": [[-7.26050714532...*	957.575
10	4	3	*{"nodes": ["4-3"], "coordinates": [[-7.26631210166...*	957.575
11	4	5	*{"nodes": ["4-5"], "coordinates": [[-7.26631210166...*	634.990
12	5	4	*{"nodes": ["5-4"], "coordinates": [[-7.26464043859...*	634.990
13	3	6	*{"nodes": ["3-6"], "coordinates": [[-7.26050714532...*	916.764
14	6	3	*{"nodes": ["6-3"], "coordinates": [[-7.25928164731...*	916.764
15	6	5	*{"nodes": ["6-5"], "coordinates": [[-7.25928164731...*	874.962
16	5	6	*{"nodes": ["5-6"], "coordinates": [[-7.26464043859...*	874.962
17	5	7	*{"nodes": ["5-7"], "coordinates": [[-7.26464043859...*	747.149
18	7	5	*{"nodes": ["7-5"], "coordinates": [[-7.26653668999...*	747.149
19	7	8	*{"nodes": ["7-8"], "coordinates": [[-7.26653668999...*	631.931
20	8	7	*{"nodes": ["8-7"], "coordinates": [[-7.26127898209...*	631.931

where

k = 0, 1, 2, 3, 4, 5, 6, 7. 8, 9 (number of iterations),
i = 0, 1, 2, 3, 4, 5, 6, 7. 8 (starting point),
j = 0, 1, 2, 3, 4, 5, 6, 7. 8 (destination point).

In (2), the selected value was the smallest value of the comparison between $x[i,j]$ and results from $x[i,k] + x[k,j]$ operation. If $x[i,j] \leq x[i,k] + x[k,j]$, value $x[k,j]$ would be selected. If $x[i,j] \geq x[i,k] + x[k,j]$, $x[i,j]$, then $x[i,j] = x[i,k] + x[k,j]$. This value exchange process was carried out through an iteration process of k times where $k = 0$ was the matrix defined at the beginning, and the first iteration was carried out starting from iteration $k = 1$ up to $k = 9$.

In the first iteration of $k = 1$, the data in the 0th row of the 0th column to the 8th column and the 0th row to the 8th-row column, and the $x[i][j]$ row, with the provisions $i = j$ with a value 0 (forming a diagonal line), the value was fixed. The value comparison iteration process was performed starting from $x[1][2]$ up to $x[8][7]$, and the smallest value was selected.

TABLE II
RESULTS OF THE *K*TH ITERATION PROCESS

The *k*th Iteration	Matrix on Data
1	$x[0][2]$, $x[0][3]$, $x[2][3]$, $x[3][2]$, $x[3][0]$, $x[2][0]$
2	$x[0][4]$, $x[1][4]$, $x[4][1]$, $x[4][0]$.
3	$x[0][4]$, $x[1][4]$, $x[2][4]$, $x[6][4]$, $x[0][6]$, $x[1][6]$, $x[2][6]$, $x[4][6]$, $x[4][0]$, $x[4][1]$, $x[4][2]$, $x[4][6]$, $x[6][0]$, $x[6][1]$, $x[6][2]$, $x[6][4]$
4	$x[0][5]$, $x[1][5]$, $x[2][5]$, $x[3][5]$, $x[5][0]$, $x[5][1]$, $x[5][2]$, $x[5][3]$
5	$x[4][6]$, $x[0][7]$, $x[1][7]$, $x[2][7]$, $x[3][7]$, $x[4][7]$, $x[6][7]$, $x[6][4]$, $x[7][0]$, $x[7][1]$, $x[7][2]$, $x[7][3]$, $x[7][4]$, $x[7][6]$
6	$x[0][7]$, $x[1][7]$, $x[2][7]$, $x[3][7]$, $x[4][7]$, $x[6][7]$, $x[7][0]$, $x[7][1]$, $x[7][2]$, $x[7][3]$, $x[7][4]$, $x[7][6]$
7	$x[0][8]$, $x[1][8]$, $x[2][8]$, $x[3][8]$, $x[4][8]$, $x[5][8]$, $x[6][8]$, $x[8][0]$, $x[8][1]$, $x[8][2]$, $x[8][3]$, $x[8][4]$, $x[8][5]$, $x[8][6]$
8	$x[0][8]$, $x[1][8]$, $x[2][8]$, $x[3][8]$, $x[4][8]$, $x[5][8]$, $x[6][8]$, $x[8][0]$, $x[8][1]$, $x[8][2]$, $x[8][3]$, $x[8][4]$, $x[8][5]$, $x[8][6]$

TABLE III
FINAL RESULTS OF THE NINTH ITERATION PROCESS

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0	0.696	1.075	1.227	2.185	2.82	2.144	3.567	4.199
1	0.696	0	0.379	0.531	1.489	2.124	1.448	2.871	3.503
2	1.075	0.379	0	0.910	1.868	2.503	1.827	3.25	3.882
3	1.227	0.531	0.910	0	0.958	1.593	0.917	2.34	2.972
4	2.185	1.489	1.868	0.958	0	0.635	1.510	1.382	2.014
5	2.820	2.124	2.503	1.593	0.635	0	0.875	0.747	1.379
6	2.144	1.448	1.827	0.917	1.51	0.875	0	1.622	2.254
7	3.567	2.871	3.250	2.340	1.382	0.747	1.622	0	0.632
8	4.199	3.503	3.882	2.972	2.014	1.379	2.254	0.632	0

From the first iteration, a matrix data was obtained with changes in value in $x[0][2]$, $x[0][3]$, $x[2][3]$, $x[3][2]$, $x[3][0]$, and $x[2][0]$. Furthermore, the results of the first iteration calculation would be used for the second iteration with a value remained changing in the 1st-row position of the 0th column to the 8th column and the 0th row to the 8th row of the 1st column. The same way was performed up to the 8th iteration, as shown in Table II.

The eighth iteration generated matrix data with changes in values on data number $x[0][8]$, $x[1][8]$, $x[2][8]$, $x[3][8]$, $x[4][8]$, $x[5][8]$, $x[6][8]$, $x[8][0]$, $x[8][1]$, $x[8][2]$, $x[8][3]$, $x[8][4]$, $x[8][5]$, and $x[8][6]$. Furthermore, the calculation results of the eighth iteration were used for the ninth iteration with fixed values changing at the position of the 8th row of the 0th column to the 8th column and the 0th row to the 8th row of the 7th column.

The ninth iteration generated matrix data, as in Table III, with the change of values in the data so that all nodes would be connected to each other and result in distances between nodes. Table III shows the total distance from the starting point to the

TABLE IV
ROUTE TRAVERSED FROM THE STARTING POINT TO THE DESTINATION

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0	0	1	1	3	4	3	5	7
1	1	1	1	1	3	4	3	5	7
2	1	2	2	1	3	4	3	5	7
3	1	3	1	3	3	4	3	5	7
4	1	3	1	4	4	4	5	5	7
5	1	3	1	4	5	5	5	5	7
6	1	3	1	6	5	6	6	5	7
7	1	3	1	4	5	7	5	7	7
8	1	3	1	4	5	7	5	8	8

destination point. For example, the total distance to be traveled from point 0 to point 8 is 4.199 km.

The route traveled on the Floyd-Warshall algorithm from the starting point, node 0, to the destination point, node 8, is shown in Table IV. Next, the nodes directly connected to node 8 between node 0 (the searched starting point) and node 8 (the searched destination point) were observed. The node connected to node 8 was node 7. Node 7 was connected with node 5, node 5 was connected with node 4, and so on, so that the initial node was found, i.e., node 0. The results of the Floyd-Warshall algorithm calculation in Table IV indicate that the shortest distance from the respondent's point to the victim's location was T1 = 0-1-3-4-5-7-8, with a travel distance of 4.199 km.

III. RESULTS AND DISCUSSION

The display of accident notification applications on victims and respondents is shown in Fig. 7.

If the victim's application states that an accident occurred, the application will send a broadcast notification to the respondent. Notifications will pop up on the notification bar in the respondent's application. The data received by the respondents were the victim's name and location in the form of latitude and longitude coordinates. Simultaneously, information about the accident will appear on the respondent's application home page, as shown in Fig. 7. The displayed information is in the form of latitude and longitude coordinates of the victim and the distance of the respondent's location to the victim's location. On the map, the markers of the victim's location are displayed according to the latitude and longitude coordinates as well as the markers of the respondent's location. On the page, there are also several activities respondent can do, including looking at list photo, which allows respondents to see photos of events taken by other respondents; take photo that respondents can do to provide evidence of the incident; making an emergency call to number 112 to make a report related to the accident, and displaying the route from the respondent's location to the victim's location. The route displays the activity results of the Floyd-Warshall algorithm, which is the shortest route of several other route options.

Several tests were conducted using *smartphones* with the Android operating system version 4 to determine the performance of this traffic accident notification application. The performed accident notification application testing was divided into two parts as follows.

Fig. 7 Respondent's application displays the route to the victim's location.

TABLE V
RESULTS OF ROUTE SELECTION FROM THE FLOYD-WARSHALL ALGORITHM

No.	**Route**	**Distance (km)**	**Selected Routes**
1	T0 = 0-1-3-6-5-7-8	4.398	T1 = 0-1-3-4-5-7-8
2	T1 = 0-1-3-4-5-7-8	4.199	
3	T2 = 0-1-2-4-5-7-8	5.614	

1) Testing of Floyd-Warshall Algorithm for Determination of Shortest Distance: A comparison of the value of system's initial *Wij* matrix with the initial *Wij* matrix manually calculated was carried out. If the system's final *W** matrix results were the same as those of the manually calculated final *W** matrix, the route test results and mileage were correct.

2) Testing of Black Box and Internet Speed: Testing of black box on Android applications was done by looking at the display and function of the application without paying attention to the internal logic structure of the application. Test parameters were internet speed, and the time it took to send notifications from the victim's application to the respondent.

A. Floyd-Warshall Algorithm Testing Results

Floyd-Warshall algorithm testing was carried out in Surabaya by transforming the Surabaya city map, which was the object of research, into a map on Android by showing the respondents' location, the victim's location, and the location of police stations and hospitals. This Floyd-Warshall algorithm test used the starting and destination points to form a route network consisting of nine nodes with four road junctions. The initial x_{ij} and z_{ij} matrices were used to obtain the nodes and the shortest route. The initial w_{ij} matrix is a matrix to find the shortest distance between each pair of nodes, while the initial

TABLE VI
BLACK BOX TEST RESULTS

No.	Feature Name	Number of Testing	Successful	Error
1	Splash Screen	20	20	0
2	Login Page	20	20	0
3	Registration Page	20	20	0
4	Home and Notification Page	20	20	0
5	Route to Victim's Location	20	20	0
6	Take Photo	20	20	0
7	See Photos	20	20	0
8	Call 112	20	20	0
9	Logout Button	20	20	0

TABLE VII
RESULTS OF INTERNET PERFORMANCE TESTING IN SENDING NOTIFICATIONS

No.	Testing Time	Time (WST)	Number of Respondents	Average Time (ms)
1	Morning	08.00 – 09.00	10	17.97
2	Midday	12.00 – 13.00	10	9.86
3	Afternoon	16.00 – 17.00	10	76.84
4	Night	20.00 – 21.00	10	6.76
Average				27.86

z_{ij} is beneficial to find the shortest route between the nodes. The initial w_{ij} and z_{ij} matrices defined as $k = 0$ were then subjected to a calculation process with the Floyd-Warshall algorithm to find the shortest route between all points within the network. After the calculation was conducted until the last iteration, i.e., when $k = 9$, $i = 9$, and $j = 9$, the final x^* matrix was obtained, indicated by the final z^* matrix of the Floyd-Warshall algorithm calculation process. This result shows the shortest route between nodes, as seen in Table V. The result of the shortest route selection from the Floyd-Warshall algorithm was T1 = 0-1-3-4-5-7-8, with a distance of 4.199 km.

B. Floyd-Warshall Algorithm Testing Results

The black box testing was carried out on every feature in the application, such as register, login, home, receive notifications, list data gallery, route view button, emergency call, take photos, upload photos, and logout. The results of the black box test of twenty trials stated that 100% of the system worked well because it obtained twenty times success and 0 failure, as shown in Table VI.

System performance testing using an internet connection was carried out by sending notifications to ten respondents simultaneously. Data was taken at specific times: morning, midday, afternoon, and night. Then, samples were taken every 10 minutes for 60 minutes of the experiment. From the sample, the average was calculated so that the average internet connection obtained every morning, midday, afternoon, and night were 17.97 ms, 9.86 ms, 76.84 ms, and 6,76 ms, respectively. The sample was used to determine the averages: the average internet connection gained each morning was 17.97 ms, daytime 9.86 ms, afternoon time 76,84 ms, and night time 6,76 ms. The average internet connection at each time was then used to calculate the average overall internet connection time, generating a value of 27.86 ms. The results of these calculations are shown in Table VII.

IV. CONCLUSION

The following conclusions can be drawn based on the results of the tests that have been carried out. In general, an accident notification application has been successfully created using a smartphone that can classify accidents and send broadcast notification messages. Respondents can receive notifications on accidents occurring around the respondent's position and can travel to the accident location according to the route displayed by the respondent's application. The results of the black box testing indicate that 100% of the application's features ran well on the Android platform. The generated route did not deviate from the test scenario, which was 0-1-3-4-5-7-8, and resulted in the shortest distance of 4.199 kilometers. The average speed of notification delivery responses from victims to respondents was 27.86 ms.

CONFLICT OF INTEREST

The authors of the paper entitled "Application of Floyd-Warshall for the Shortest Route Search in a Traffic Accident Notification App" declare that there is no conflict of interest.

AUTHOR CONTRIBUTION

Methodology, formal analysis of writing, preparation of original drafts, reviewing and editing, Haniah Mahmudah; software, M Fajar Ibrahim; validation, Okkie Puspitorini; review, Nur Adi S and Ari Wijayanti.

REFERENCES

[1] E. Lesmana *et al.*, "The Application of Floyd-Warshall Algorithm in Solving Shortest Path Problem for Fire Evacuation System at High Rise Building (Case Study at eL Royale Hotel Bandung)," *9th Int. Conf. Ind. Eng., Oper. Manag.*, 2019, pp 2169–2178.

[2] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni, "Wolverine: Traffic and Road Condition Estimation Using Smartphone Sensors," 2012 *Fourth Int. Conf. Commun. Syst., Netw. (COMSNETS 2012)*, 2012, pp. 1–6.

[3] P. Mohan, V.N. Padmanabhan, and R. Ramjee, "TrafficSense: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones", *Proc. 6th ACM Conf. Embed. Netw. Sensor Syst.*, pp 323–336, 2008.

[4] S. Sharma and S. Sebastian, "IoT Based Car Accident Detection and Notification Algorithm for General Road Accidents," *Int. J. Elect., Comput. Eng. (IJECE)*, Vol. 9, No. 5, pp. 4020–4026, Oct. 2019.

[5] S. Farhan, S. Andryana, and N. Hayati, "Implementasi Bellman-Ford dan *Floyd-Warshall* dalam Menentukan Jalur Terpendek Menuju Universitas Nasional Berbasis Android", *J. Ilm. Penelit., Pembelaj. Inform. (JIPI)*, Vol. 5, No.2, pp. 123–132, Dec. 2020.

[6] F.W. Ningrum and T. Andrasto, "*Penerapan Algoritma Floyd-Warshall dalam Menentukan Rute Terpendek pada Pemodelan Jaringan Pariwisata di Kota Semarang," J. Tek. Elekt.*, Vol. 8, No. 1, pp. 21–24, Jan.-Jun. 2016.

[7] K. Manaf, *et al.*, "Designing Futsal Match Finder Application with Floyd-Warshall," *J. Phys.: Conf. Ser.*, Vol. 1280, No. 2, pp. 1–8, Nov. 2019.

[8] V.A. Nawagusti, "Penerapan Algoritma Floyd Warshall dalam Aplikasi Penentuan Rute Terpendek Mencari Lokasi BTS (Base Tower Station)

pada PT. GCI Palembang," *J. Nas. Teknol., Sist. Inf.*, Vol. 4, No. 2, pp. 81–88, Aug. 2018.

[9] B.I.A. Prasetyo and A. Maslan, "Analisis Perbandingan pada Algoritma Bellman Ford dan Dijkstra pada Google Map," *Khazanah Ilmu Berazam*, Vol. 3, No. 2, pp. 337–349, 2020.

[10] N.R. Chopde and M.K. Nichat, "Landmark Based Shortest Path Detection by Using A* and Haversine Formula," *Int. J. Innov. Res. Comput., Commun. Eng.*, Vol. 1, No. 2, pp. 298–302, Apr. 2013.

[11] Ramadiani, D. Bukhori, Azainil, and N. Dengen, "Floyd-Warshall Algorithm to Determine the Shortest Path Based on Android," *2018 IOP Conf. Ser.: Earth Environ. Sci.*, Vol. 144, pp. 1-8, May 2018.

[12] K. Gutenschwager, S. Völker, A. Radtke, and G. Zeller, "The Shortest Path: Comparison of Different Approaches and Implementations for the Automatic Routing of Vehicles," *Proc. 2012 Winter Simul. Conf. (WSC)*, 2012, pp. 3312-3323.

[13] M. Kairanbay and H.M. Jani,"A Review and Evaluations of Shortest Path Algorithms," *Int. J. Sci., Technol. Res.*, Vol. 2, No. 6, pp. 99-104, Jun. 2013

[14] U.E. Chukwuka, J.A. Oladunjoye, and S. Emmanuel, "A Study of Intelligent Route Guidance System Using Dijkstra's Heuristic Shortest Path Algorithm," *Int. J. Inf. Technol., Innov. Africa*, Vol. 13, No. 2, pp. 1-23, Apr. 2018.

[15] S. Sanan, L. Jain, and B. Kappor, "Shortest Path Algorithm," *Int. J. Appl., Innov. Eng., Manag. (IJAIEM)*, Vol. 2, No. 7, pp. 316-320, Jul. 2013.

[16] M. Głąbowski, B. Musznicki, P. Nowak, and P. Zwierzykowski, "Efficiency Evaluation of Shortest Path Algorithms," *AICT 2013: The Ninth Adv. Int. Conf. Telecommun.*, 2013, pp. 154-160.

[17] P. Charoenporn, "Comparison of Algorithms for Searching Shortest Path and Implementations for the Searching Routing System via Web Services," *Proc. 2018 the 8th Int. Workshop Comput. Sci., Eng. (WCSE 2018)*, 2018, pp. 516-520.

[18] R. Kampf, O. Stopka, L. Bartuška, and K. Zeman, "Circulation of Vehicles as an Important Parameter of Public Transport Efficiency," *Proc. 19th Int. Conf. Transp. Means*, 2015, pp. 143-146.

[19] W. Xing *et al.*, "An Improved Savings Method for Vehicle Routing Problem", *2016 2nd Int. Conf. Control Sci., Syst. Eng. (ICCSSE)*, 2016, pp. 1-4.

[20] Y.Z. Mehrjerdi, "A Multiple Objective Stochastic Approach to Vehicle Routing Problem," *Int. J. Adv. Manuf. Technol.*, Vol. 74, pp. 1149-1158, Sep. 2014.

[21] M. Stanojevic, B. Stanojevic, and M. Vujoševic, "Enhanced Savings Calculation and Its Applications for Solving Capacitated Vehicle Routing Problem," *Appl. Math., Comput.*, Vol. 219, No. 20, pp. 10302-10312, Jun. 2013.