International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V69I2P108

Volume 69 Issue 2, 53-60, February 2021

Original Article

Haversine Algorithm Design using the Google Maps API Method for Android-based Public Security Applications

Gerzha Hayat Prakarsha[1], Herlina Dessy Kristianty Lumbantobing[2], M. Rifqi Ramadhan[3], Ifan Prihandi[4]

[1,2,3,4] *Mercu vuana University, Jakarta, Indonesia.*

Received Date: 26 December 2020
Revised Date: 08 February 2021
Accepted Date: 011 February 2021

Abstract *- The rapid development of technology must of course be balanced with all the needs of society. Not only physically, but also social needs which include a sense of security. A sense of security will be obtained when someone can avoid an emergency that can disrupt the life of someone from a group of people or society. By building an Android-based public security application using the Panic Button and the Haversine algorithm, the public can connect with security forces in certain locations and other related parties. The purpose of the Public Security application is to help people get help quickly, accurately, and safely. This application also displays the closest agencies in the DKI Jakarta area. Haversine is an equation in navigation by giving a large circular distance (radius) between two points on the surface of a sphere (earth) based on longitude and latitude. This Public Security application is integrated with the Google Maps API to find out the location of the user, as well as other related parties and Google Firebase as the manager of all data in the application.*

Keywords *- Applications, Panic Button, Android, Haversine, Google Maps, Google Firebase*

I. INTRODUCTION

The rapid development of information technology has had a considerable influence in various fields, starting from the world of health, government, defense, and other fields and is an inseparable part of business activities and daily life [1]. By using several media such as laptops, computers, smartphones, or other media, people can access the internet to obtain the desired information. According to BPS (Central Statistics Agency) in Indonesia's Telecommunication Statistics data, it was noted that in Indonesia in 2017, the most widely used media to access the internet was smartphones with a percentage of 91.45% in 2017 or 241.9 million Indonesians accessing the internet using smartphone media [2].

Information circulating on the internet will be easily obtained, developed, and used as well as making the flow of information so fast that there is no longer any distance limiting the distribution of the information [3]. In order to maximize its use, technological developments must also be balanced with all the needs of society. Indonesia also continues to experience an increase in population every year with a growth rate of 1.33% in 2010-2018. This increase in population certainly affects the needs of the community which continues to grow [4].

Basic needs in society can be classified into biological or physical needs such as eating, drinking, and social needs such as self-actualization, social roles, and a sense of security. According to Abraham Maslow in the hierarchy of human needs (Maslow, 1943), security is at the second level under basic human needs such as clothing, food, and shelter. This shows that security is an important human need. Basically, humans cannot predict the danger that can threaten the safety or even the life of a person [5].

Criminality Statistics 2019, BPS (Central Statistics Agency) noted that in 2018, the percentage of the population who became victims of crime was 1.11 percent of the population of Indonesia. This percentage increased from the experience of society in the 2017 period of 1.08 percent with the following types of crimes [6].

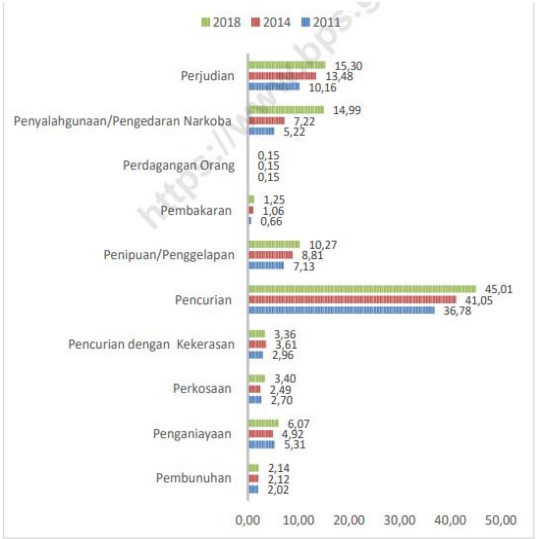

Fig. 1 Percentage of Sub-Districts Experiencing Crimes by Type of Crime During 2011, 2014 and 2018

It can be concluded that a better security system is needed by using the most media used by the community, namely smartphones, to be able to continue to reduce the incidence of crime. With that, the researcher built an Android-based application that can connect the community with the local security (security guard). This application is focused on housing that has a security party (security guard). The Public Security application is a public security application with a panic button, Google Maps as a travel map and directions, and real-time location. This application also applies the Haversine Algorithm to connect two closest user points within the same radius. Haversine is an equation in navigation by giving a large circle distance (radius) between two points on the surface of the sphere (earth) based on longitude and latitude [7].

By using the Public Security Application, it is hoped that the assistance process can be made fast, precise, and safe. The community can press the emergency button provided and the system will connect the community with local security forces.

A. *Research Problem*

How to build a public protection and security system in an Android-based application using the Panic Button? How is the application of the Haversine Algorithm in a built application?

B. *Object and Benefits*

The purpose of the implementation of this research is to build an Android-based application to improve the protection and security of the community from all kinds of emergency events that can harm or harm someone. This application connects users with local security, which later users can ask for help quickly and precisely.

C. *Limitation of Research*

- This application was developed in an Android-based environment using the application of the Haversine Algorithm.
- The application uses the Panic Button which can only be used a maximum of 1x (times) in 24 hours and requires an internet connection.
- Users can only request and receive assistance when the User is within a predetermined radius and in the same area.

II. STUDY LITERATURE

A. *Android Based Applications*

The definition of an application is a program that people use to do something on a computer system. Android is a Linux-based operating system designed for touch screen mobile devices such as smartphones and tablet computers (Safaat, 2015). Android was originally developed by Android, Inc., with financial support from Google, which later bought it in 2005 [8]. Then in November 2007 Google together with the Open Handset Alliance, namely the consortium of open mobile devices, released the Google Android SDK, after announcing it a week earlier and received a tremendous welcome. The SDK software can be used to develop applications on mobile devices, namely: Operating Systems, Middleware, and main applications for mobile devices [9]. Therefore, most of the code from android is released on the open-source Apache License, which means that anyone can download the complete source code of the android operating system [10].

B. *Haversine*

This formula was first discovered by Jamez Andrew in 1805 and was first used by Josef de Mendozay Ríos in 1801. The term haversine itself was coined in 1835 by Prof. James Inman. Josef de Mendoza y Ríos used haversine for the first time in his research on "Major Astronomical Nautical Problems", Proc. Royal Soc, 22 December 1796 [11]. One method for calculating the distance of latitude-longitude points on the earth's surface as a variable input is using the Haversine formula. Haversine as a formula in navigation calculates the distance of a circle between latitude and longitude with the assumption that the radius of R is 6367.45 km, and the location of 2 points on the spherical coordinates (latitude and longitude). The assumed Haversine formula ignores the structure of the earth's surface (valley depth and hillside height), which is quite accurate for most calculations due to the fact that the earth is slightly elliptical (ellipsoidal factor) [7]. Latitude is a line used to measure the distance between the north and south of the earth as opposed to the equator. Whereas longitude is a line used to measure the distance between the west and east of the earth from the prime meridian [12]. About the use of maps that already have two dimensions will have points depicted in whole numbers. In the calculation step, haversine is the first to convert the integer values of latitude and longitude numbers into radians, then these numbers are calculated in the haversine algorithm [13].

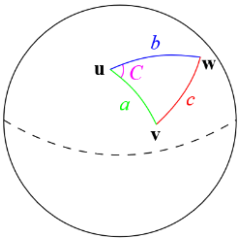

Fig. 2 Haversine Formula

$$\cos(c) = \cos(a - b) + \sin(a)\sin(b)\cos(C)$$

In spherical units, a "triangle" on the surface of a sphere is defined as the large circle connecting the three points u, v, and w on the sphere. If the lengths of the three sides are (from u to v), b (from u to w), and c (from v to w), and the opposite point of view c is C [11].

C. *Panic Button*

Panic Button is an emergency button located on an Android-based application, used by Application users in an emergency or urgent condition. This emergency button functions to send emergency information to fellow application users [14].

D. Google Maps API

Google Application Programming Interface (API) is a service provided to the public by Google Inc. released in 2005. Google API products allow users to create applications according to developer needs [15]. Google Maps API, Google Maps provides the user with details about the geographic status of his location, details about the surrounding buildings, the user's live location using GPS, the traffic status of specific street navigation, etc. For the development of new projects and commercial use of other entities, google offers subscriptions to the service map API [16]. In the Google Maps API, there are 4 types of map model options provided by Google, including [17]:

- ROADMAP, to display 2-dimensional maps.
- SATELLITE, to display satellite photos.
- TERRAIN, to show the physical relief of the earth appearing to the surface and show how high a location is.
- HYBRID, will display a satellite photo on it also illustrates what appears on ROADMAP (street and city name)

E. Google Firebase Database

Firebase is a web and mobile development platform. Firebase uses can be found in many applications. Firebase is traditionally used in applications that require fast exchange and refresh of data. Chasapis, Mitropoulos, and Douligeris (2019) discuss the advantages of using Firebase. Among the many, it is possible to quickly add more functionality to the application and to add cloud functionality without the need for a dedicated backend server [18]. Firebase Database is designed and developed by Google, where different types of content depending on the communication are available to design applications. The information is stored in the form of JavaScript Notation Objects (JSON) in this database. Firebase Real-time Database uses synchronization information which updates information every second and related instrument are updated with new data synchronously according to the database [19]. Until recently on the internet, many databases were available except the main problem of databases is time. Most of the databases update their real-time values with a duration of 15 seconds and so is not sufficient to provide a decent result for every real-time value. But despite this weakness, Google's database is better than other databases because here the real-time value updates within one second. Google Firebase database is a free platform where all kinds of applications (android, ios, web app) can be created, Firebase console platform provides many features, such as (push notification, cloud messaging, database). Without paying anyone can use their features. In the database, there are mainly two types of databases, namely cloud fire stores, and real-time databases. A real-time database is used to store real-time values and a cloud Firestore is used for cloud messaging [20].

III. ANALYSIS AND DESIGN SYSTEM

A. Research Flow Diagram

The Research Flowchart explains the stages of research that the authors use to make this research.

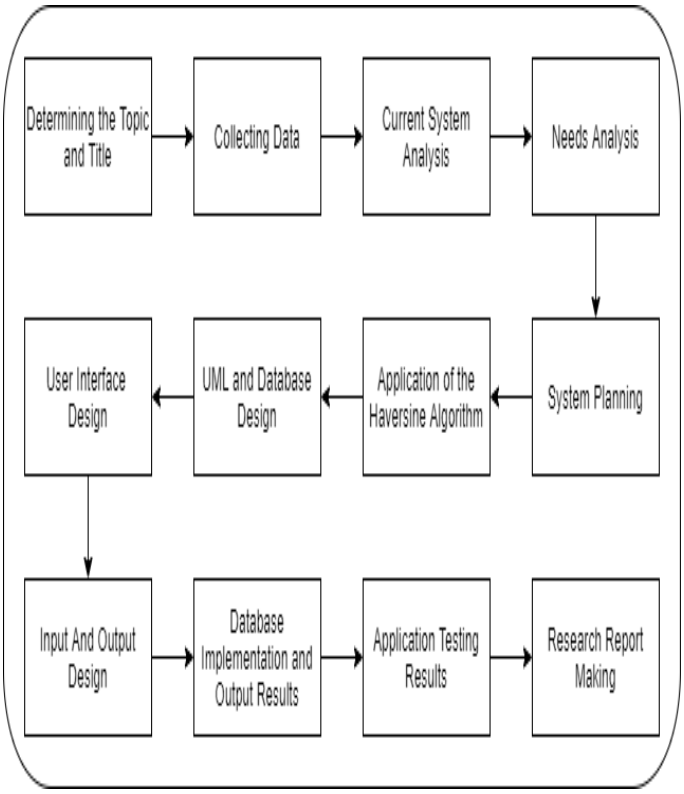

Fig. 3.1 Research Flow Diagram

B. System Analysis is Running

This research begins by visiting several housing estates located in Urban Village Petir, Tangerang City, to conduct observations and interviews on the process of the community security system that has been running in the housing complex, as well as conducting interviews with the Head of Hamlets (RW) and the Head of the Neighborhood (RT) or the housing management, security forces and the surrounding community to collect data and determine the background to the problem.

From the results of observations and interviews with related parties, the current stages of the business process during an emergency or people who need help, namely asking for help from the security forces (security guard) by contacting the security guard via telephone or chat or visiting the nearest security post. Furthermore, the security guard will come to the location in question to provide assistance or assistance to the party asking for help or residents of the housing. After that the security guard reports to the Head of Hamlets (RW) and the Head of the Neighborhood (RT) or the housing manager and the incident is recorded in the housing management report book.

C. *Proposal System*

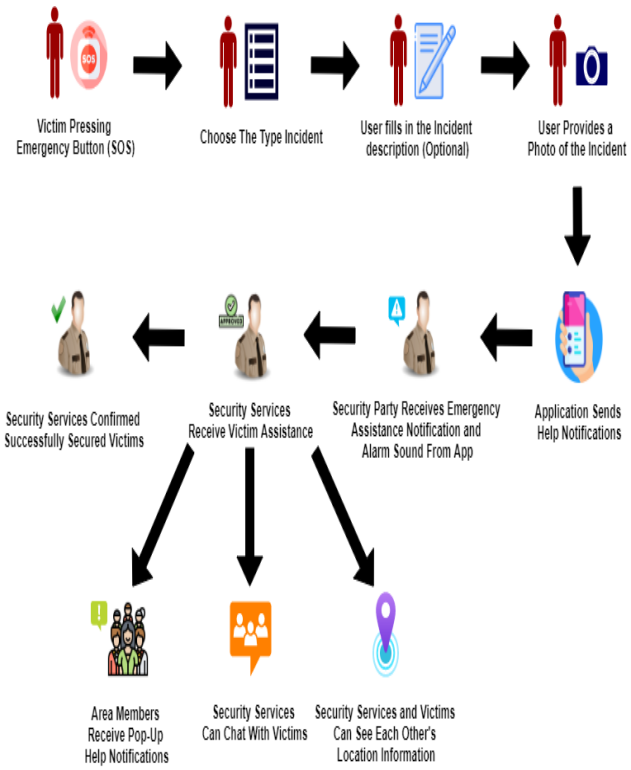

Fig. 3.2 Proposal System

D. *Supporting Hardware and Software*

a) Hardware

The hardware needed to support this application is a smartphone with the following specifications:

- Minimum Intel Atom Quad-Core Z3580 Speed 2.3Ghz
- Random Access Memory (RAM) with a minimum capacity of 2 GB
- Internal memory with a minimum capacity of 8 GB

b) Software

The software used to complete reports and system design includes:

Windows Operating System, Android Operating System, Android Studio, Firebase Realtime Database, Draw.Io, and Lucid.app, Figma and Photoshop, Firebase Notification, Github, and Node JS.

IV. APPLICATION OF THE HAVERSINE FORMULA

A. *Detect Point Area*

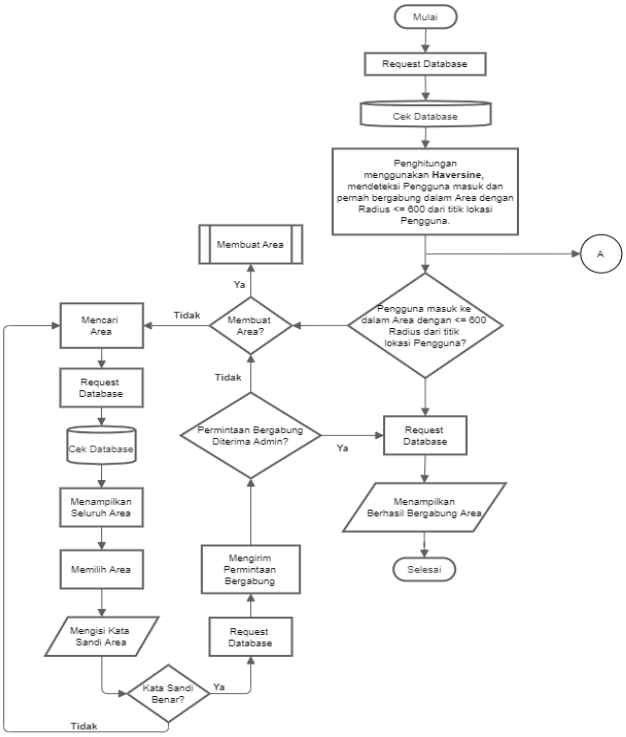

Fig. 3.3 Flow Chart Detect Point of User Location

The flow chart above describes a system that detects the user's location using the Haversine Algorithm calculation with the Area group around the User. The User will automatically join the Area group around him if the User has previously joined the Area group. If the User has not joined the Area group that is around him, the User will be directed to create a new Area group or can search and join an existing Area group.

B. *Creating Area*

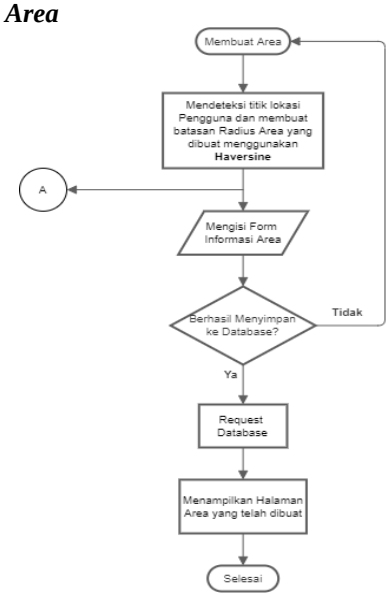

Fig. 3.4 Flow Chart Creating Area

Users can create a new Area group that will be determined radius distance with User point using Haversine Algorithm calculation.

C. Feature Prompts Help

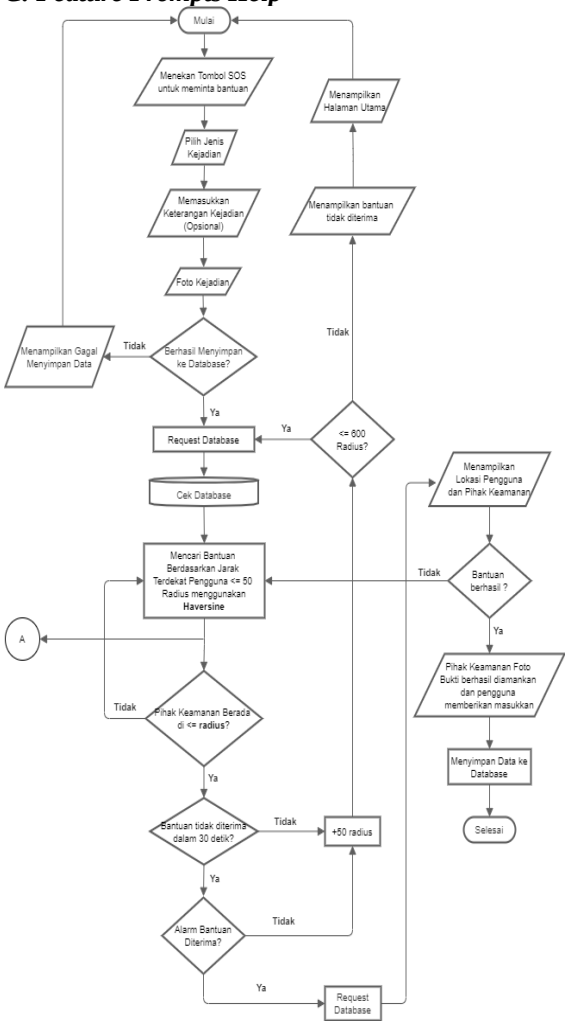

Fig. 3.5 Flow Chart Feature Prompts for Help

In the search for help, the User will press the SOS button and provide some information regarding emergency events. After that, the system will automatically wait for the location of the Security Party with a 600 radius in the Area group. If the secured party's location point is within a 50 radius of the user's location point, the system will seek assistance from the security party using the Haversine Algorithm calculation. If not, the system will continue to search using the computation of the Haversine Algorithm. Furthermore, if the Security Service is in a 600 radius the System Area group will issue a request for assistance and wait for assistance to be received. If help is rejected or not received within 30 seconds the system will add a radius of +50 for the Security Party search to the 600 radius limit. If you have not received assistance either, the system will display a help message not received

D. Haversine

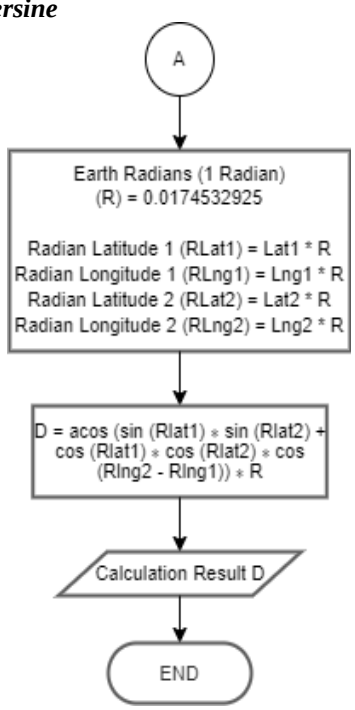

Fig. 3.6 Haversine's Algorithm

The flow diagram above is the Haversine Algorithm which is used in several processes in the Public Security Application.

E. Example of Calculating the Haversine Algorithm in Applications

1. The formula for finding the nearest area to the user's location point
   a) First Coordinate Point (Area Point: GLC Amerika Latin)

Latitude 1 = -6.1876
Longitude 1 = 106.7061

   b) Second Coordinate Point (Users Near GLC Amerika Latin)

Latitude 2 = -6.1884 /
Longitude 2 = 106.7059
Radian (R) = 0.0174532925 (1 Radian)
Radian Latitude1 (Rlatitude1) = -6.1876 * 0.0174532925 = -0.107993992673
Radian Longitude1 (Rlongitude1) = 106.7061 * 0.0174532925 = 1.86237277483425
Radian Latitude2 (Rlatitude2) = -6.1884 * 0.0174532925 = -0.108007955307
Radian Longitude2 (Rlongitude2) = 106.7059 * 0.0174532925 = 1.86236928417575

D = acos (sin(Rlatitude1) * sin(Rlatitude2) + cos(Rlongitude1) * cos(Rlatitude2) * cos(Rlongitude2 − Rlongitude1)) * R

D = acos (sin(−0.107993992673) * sin(−0.108007955307) + cos(1.86237277483425) * cos(1.86236928417575) * cos(1.86237277483425−1.86236928417575)) * 6371
D = Radius: 92.00 meters = 0.092 kilometers

Table 3.1 Example of Calculating the Haversine Algorithm Finding the Area Closest to the User's Location Point

Location of the Security Guard	Latitude	Longitude	Radian Latitude	Radian Longitude	Distance
GLC Cluster Amerika Latin	-6.1876	106.7061	-0.1079939926 73	1.8623 7277483425	Radius: 92.00 meters = 0.092 kilometers
GLC Cluster Asia	-6.1892	106.7005	-0.1080219179 41	1.8622 7503639625	Radius: 604.00 meters = 0.604 kilometers
GLC Cluster Australia	-6.1844	106.7037	-0.1079381421 37	1.8623 3088693225	Radius: 507.00 meters = 0.507 kilometers
Metland Puri	-6.1947	106.7091	-0.1081179110 4975	1.8624 2513471175	Radius: 785.00 meters = 0.785 kilometers
GLC Cluster East Asia	-6.1925	106.6995	-0.1080795138 0625	1.8622 5758310375	Radius: 842.00 meters = 0.842 kilometers

Table 3.2 Example of Calculating the Haversine Algorithm Finding the Security Officer Nearest to the User's Location Point

Location	Latitude	Longitude	Radian Latitude	Radian Longitude	Distance
GLC Cluster Amerika Latin	-6.1876	106.7061	-0.1079939926 73	1.8623 7277483425	Radius: 92.00 meters = 0.092 kilometers
GLC Cluster Asia	-6.1892	106.7005	-0.1080219179 41	1.8622 7503639625	Radius: 604.00 meters = 0.604 kilometers
GLC Cluster Australia	-6.1844	106.7037	-0.1079381421 37	1.8623 3088693225	Radius: 507.00 meters = 0.507 kilometers
Metland Puri	-6.1947	106.7091	-0.1081179110 4975	1.8624 2513471175	Radius: 785.00 meters = 0.785 kilometers
GLC Cluster East Asia	-6.1925	106.6995	-0.1080795138 0625	1.8622 5758310375	Radius: 842.00 meters = 0.842 kilometers

2. The Formula for Finding the Nearest Security Officer
   a) First Coordinate Point (A Security Guard's Point Near GLC Amerika Latin)

Latitude 1 = -6.1876
Longitude 1 = 106.7061

   b) Second Coordinate Point (Users Near GLC Amerika Latin)

Latitude 2 = -6.1884
Longitude 2 = 106.7059
Radian (R) = 0.0174532925 (1 Radian)
Radian Latitude1 (Rlatitude1) = -6.1876 * 0.0174532925 = -0.107993992673
Radian Longitude1 (Rlongitude1) = 106.7061 * 0.0174532925 = 1.86237277483425
Radian Latitude2 (Rlatitude2) = -6.1884 * 0.0174532925 = -0.108007955307
Radian Longitude2 (Rlongitude2) = 106.7059 * 0.0174532925 = 1.86236928417575
D = acos (sin(Rlatitude1) * sin(Rlatitude2) + cos(Rlongitude1) * cos(Rlatitude2) * cos(Rlongitude2 − Rlongitude1)) * R

D = acos (sin(−0.107993992673) * sin(−0.108007955307) + cos(1.86237277483425) * cos(1.86236928417575) * cos(1.86237277483425−1.86236928417575)) * 6371
D = Radius: 92.00 meters = 0.092 kilometers

Answers to the Haversine Formula Combined with Maximum Radius of Requesting Help
Distance between users and officers (PP distance) = radius: 92.00 meters or 0.092 kilometers
Radius for Requesting Assistance (radiusMB): 50.00 Meters = 0.05 kilometers
If radiusMB <= (Less than) distance PP then please request assistance
If not, then the Radius for Requesting Assistance (radiusMB) plus the Radius of 50.00 Meters

B. Implementation of Google Maps API in Public Security Applications

The following is an illustration of the use of the Google Maps API in the Public Security Application.

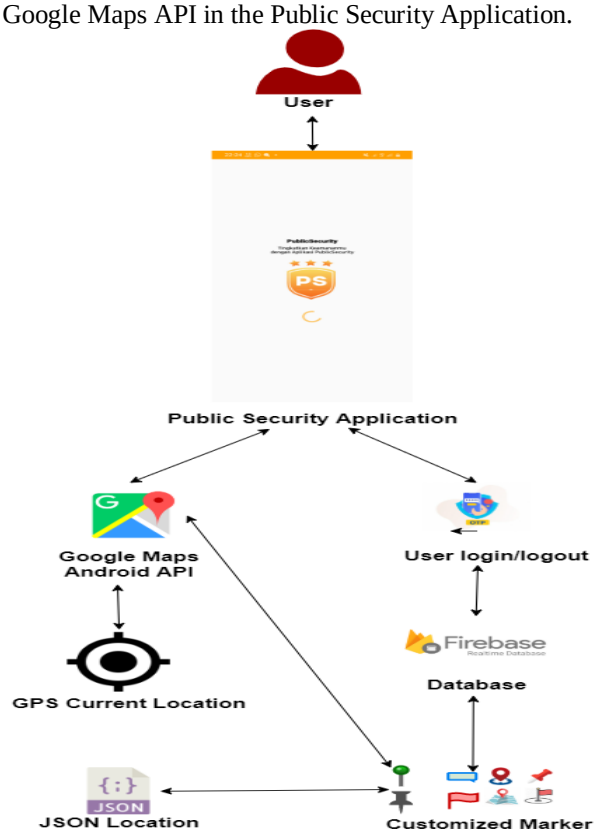

Fig 3.7 Implemented Google Maps API

V. IMPLEMENTATION AND TESTING

In the implementation stage and system testing is carried out after the analysis and design stages have been completed. This stage will explain the implementation and testing of the system that has been created, namely the Public Security Application.

The implementation stage has two scopes, namely the specification of system requirements which includes hardware and software, and the implementation of supporting application systems which include the coding process and the application of the user interface process according to the design.

The application of the Haversine Algorithm in the Public Security Application is used to detect the user's location point with the closest Area group, create Area groups, and request the help of the nearest Security.

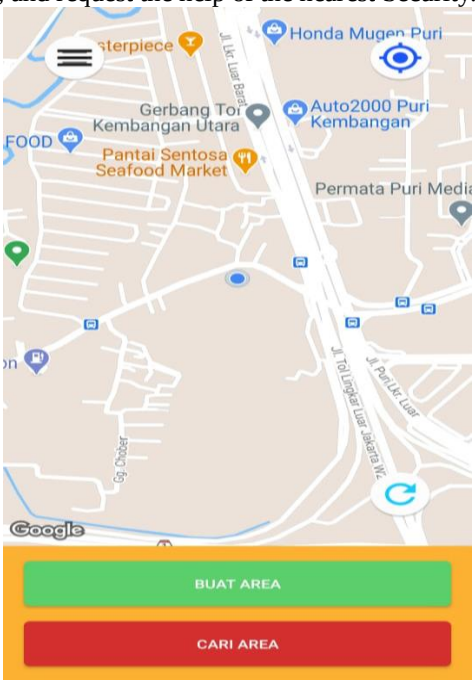

Fig 4.1 Initial Display Detection Location User Has Not Joined the Nearby Area group

When a User accesses the Area feature, the system will automatically detect the closest Area group to the user's location point. If the User is not already joined in the Area group closest to the user's current location point, the system will display the option to create a new Area group or search for the closest Area group and then join the Area group.

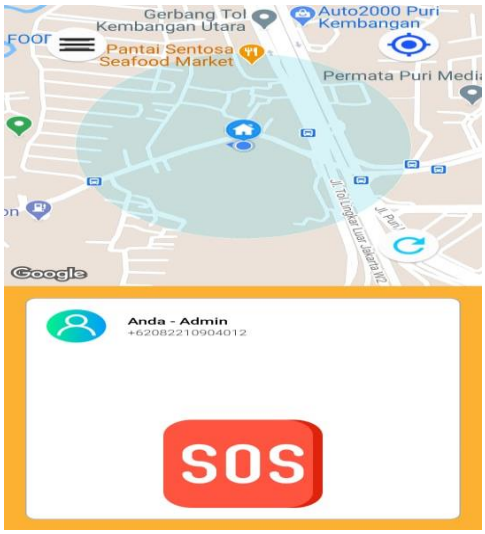

Fig 4.2 Initial Display Detection Location User Has Joined the Nearby Area group

Furthermore, the User will be directed to the initial view of the Area group page if the system succeeds in detecting the user's location point with the closest Area group and has previously joined the Area group.

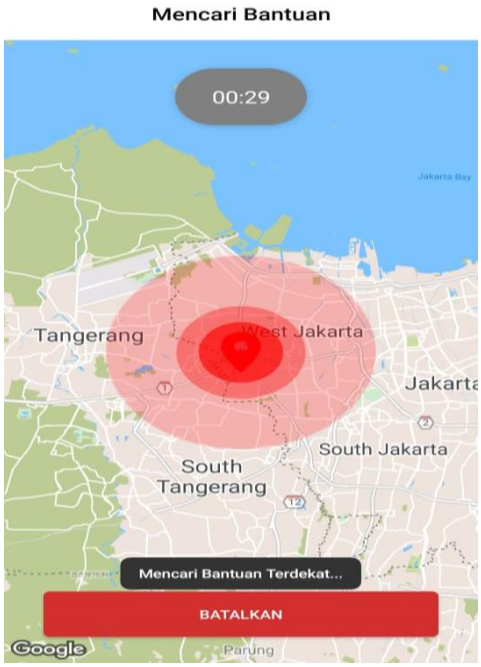

Fig 4.3 Display Nearest Security Detection Requesting Help

In requesting help, the User can select the SOS button and provide some information regarding emergency events. Furthermore, the system will automatically send a request to the nearest Security Party starting from a 50 radius from the user's location point.

VI. CONCLUSION

Based on the research that has been done, the following conclusions can be drawn:

- With the construction of a security system that helps connect Users with the Security Party, it can help every community who experiences an emergency and can reduce the number of crimes.
- The application of the Haversine Algorithm is very helpful in determining the radius to detect the closest Security Party or User.
- The results of this study are expected to be developed into the IOS version and the website.
- This application is also expected to be developed by linking requests for assistance to relevant agencies such as hospitals, police, fire departments, and so on. So that the handling of emergency events that occur in the community can be handled quickly and precisely.
- Continuation of the development of this application can later be suggested using the addition of A-Star Algorithm or DIJKSTRA Algorithm.

REFERENCES

[1] A. Ratnasari and I. Ranggadara, COBIT 5 for Improving Production Performance using DSS Domain, Int. J. Innov. Technol. Explor. Eng., 9(4)(2020) 678–681 doi: 10.35940/ijitee.c8619.029420.

[2] A. A. B. Ruíz, No Covariance structure analysis Title on health-related indicators in the elderly at home with a focus on subjective health 3(2)(2015) 54–67, [Online].Available: http://repositorio.unan.edu.ni/2986/1/5624.pdf.

[3] J. T. Santoso, M. C. Wibowo, B. Raharjo, and M. Mufadhol, Gammu and kalkun for information services and sales based on information technology, Int. J. Electr. Comput. Eng., 10(2)(2020) 2110-2116 doi: 10.11591/ijece.v10i2..

[4] BPS, Ht Tp S : // W W W . B . G, BPS, (Indonesian Stat., p. Jakarta: Badan Pusat Statistik, (2019).

[5] M. Prawira, H. T. Sukmana, V. Amrizal, and U. Rahardja, A Prototype of Android-Based Emergency Management Application, 2019 7th Int. Conf. Cyber IT Serv. Manag. CITSM (2019) doi: 10.1109/CITSM47753.2019.8965337.

[6] BPS, Statistik Kriminal Badan Pusat Statistik, , (2019) 1–218.

[7] E. Maria, E. Budiman, Haviluddin, and M. Taruk, Measure distance locating nearest public facilities using Haversine and Euclidean Methods, J. Phys. Conf. Ser., 1450(1)(2020) doi: 10.1088/1742-6596/1450/1/012080.

[8] N. C. Le, T. M. Nguyen, T. Truong, N. D. Nguyen, and T. Ngo, A Machine Learning Approach for Real-Time Android Malware Detection, Proc. - RIVF Int. Conf. Comput. Commun. Technol. RIVF (2020), doi: 10.1109/RIVF48685.2020.9140771.

[9] M. R. Amin and M. Atiquzzaman, Behavioral Malware Detection Approaches for Android, (2016).

[10] O. Mlouki, F. Khomh, and G. Antoniol, On the Detection of Licenses Violations in the Android Ecosystem, 382–392, (2016) doi: 10.1109/saner.2016.73.

[11] P. Dauni, M. D. Firdaus, R. Asfariani, M. I. N., Saputra, A. A. Hidayat, and W. B. Zulfikar, Implementation of Haversine formula for school location tracking, J. Phys. Conf. Ser., 1402(7)(2019) doi: 10.1088/1742-6596/1402/7/077028.

[12] M. Basyir, M. Nasir, S. Suryati, and W. Mellyssa, "Determination of Nearest Emergency Service Office using Haversine Formula Based on Android Platform, Emit. Int. J. Eng. Technol., 5(2)(2018) 270–278, doi: 10.24003/emitter.v5i2.220.

[13] C. N. Alam, K. Manaf, A. R. Atmadja, and D. K. Aurum, Implementation of haversine formula for counting event visitor in the radius based on Android application, Proc. 2016 4th Int. Conf. Cyber IT Serv. Manag. CITSM (2016) 2–7, doi: 10.1109/CITSM.2016.7577575.

[14] S. D. Damayanti, M. Suryanegara, and A. P. Button, Designing A LoRa-Based Panic Button for Bali Smart Island Project, 2019 7th Int. Conf. Smart Comput. Commun., (2019)1–5.

[15] N. G. Kalyanpad, C. K. Hanni, and K. V. K. Rao, Mode Choice Analysis using Web-based Revealed Preference Questionnaire, Stated Preference Experiment and Google Maps API, Transp. Res. Procedia, 48(2019) 3390–3400, (2020) doi: 10.1016/j.trpro.2020.08.115.

[16] P. Adhish, K. Abhijith, and S. R. M, Adaptive Traffic Light Control using Google Maps API at Multiple Road Intersections, Int. J. Eng. Adv. Technol., 9(2)(2019) 1802–1806, doi: 10.35940/ijeat.b2356.129219.

[17] A. M. Luthfi, N. Karna, and R. Mayasari,Google maps API implementation on IOT platform for tracking an object using GPS, Proc. - IEEE Asia Pacific Conf. Wirel. Mobile, APWiMob 126–131, (2019) doi: 10.1109/APWiMob48441.2019.8964139.

[18] P. Tykal, D. Brnovik, and J. Landa, Creating a food menu application for mendel university in Brno, Acta Univ. Agric. Silvic. Mendelianae Brun., 68(1)(2020) 275–280, doi: 10.11118/actaun202068010275.

[19] A. Kumar Sharma and L. Mohan Saini, IoT based Diagnosing Myocardial Infarction through Firebase Web Application, Proc. 3rd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA (2019) 190–195, doi: 10.1109/ICECA.2019.8822150.

[20] L. Goswami, Power Line Transmission through GOOGLE Firebase database, no. Icoei, , (2020) 415–420.

[21] M.Nandha Kishore, A.Sridhar, S.Divakara, Advanced Security Strategy in Smart E- Voting System, SSRG International Journal of Computer Science and Engineering 2(6) (2015) 13-19.