Finding Nearest Mosque Using Haversine Formula on Android Platform

Indah Setyorini[1]**, Desi Ramayanti**[2]
[1,2]Informatics Department, Faculty of Computer, Universitas Mercubuana, Jakarta, Indonesia
[1]Indahsetyorini97@gmail.com, [2] Desi.ramayanti@mercubuana.ac.id

Abstract- Prayer is the second pillar of Islam and it is emphasized after the two sentences of syahadat. It has been said that it's as perfect and as good as worship. Prayer is one of the obligations for Muslims and must be done both for Mukimin and Musafir. The virtue of prayer in the mosque is the same as the value of 20 degrees of reward compared to prayer elsewhere. With the traffic jam phenomenon that is increasing due to heavy traffic. For someone who is traveling home in the afternoon before maghrib, some people are always confused to do Maghrib prayer because of the lack of information about the nearest mosque. The problem can be solved by finding the shortest path is a problem to find the minimum route from the starting point to the destination point with the shortest route. This research was made to design an application that provides direction to the nearest mosque in the area of Larangan, Tangerang city. In the making of this system to find the shortest path can be solved using the haversine formula and using the java programming language, where java is used to build applications while the database used is the MySQL database and the use of Google Maps API in determining the route.

Keywords- Mosque, Haversine Formula, Google Maps

I. INTRODUCTION

Indonesia is one of a country that has a big population in the world and more than 87% of the population are Muslims. Shalat is the second pillar of Islam and it is a very important pillar after the shahada. Then, shalat is important to worship and prescribed as completion of worship. Furthermore, shalat is an obligation for women and men who have reached their full age and must be carried out under any circumstances, including travel[1].

People can take pray in everywhere, expect in restricted places such as toilets and graves. But, shalat has a better priority in the mosque. A mosque is a place that was established for people to take a pray, especially for praying together[2]. In almost all regions in Indonesia, there are many mosque buildings. Some are on the main road, street or in a residential. Some of the mosques can be accessed easily, and some are difficult, some can be passed by cars and motorbikes, and some can only be accessed by motorcycles.

In this research, the problem that be raised is how to provide easy access to the mosque in an area intended for those who are traveling. Praying on time is being a problem when we are on traveling and in an unknown area. So, when the time to pray has begun, we often difficult to find the mosque. This often results in a person being late in prayer, which should be done on time. Besides that, someone who is in a strange place will make it difficult to estimate the time to get to the location of the place of worship on time and access vehicles that can be used to get to the location.

The research focused on building an application that helps people to find the nearest mosque based on the location of people. This research also give provides the route and the information about environmental conditions leading to the mosque., whether it can be accessed by car or motorcycle. In this study, the application builds for mobile platforms based on the Android operating system.

Global Positioning System (GPS) is used to give a path and provide a recommendation to the nearest mosque. Global Positioning System (GPS) is a radio navigation system for positioning using satellites [3]–[7]. GPS can provide a position with three-dimensional speed and information on time and speed moving continuously throughout the world [3]. With the GPS smartphone users can find out the coordinates of the user in the form of latitude and longitude.

In a previous research Prasetyo and Hastuti build an application for giving recommendation the nearest church in Semarang, This research use the haversine method to find the nearest church based on user location[8]. n this research haversine formula was successfully applied to the application and concluded that haversine formula can produce optimal results.

In this study, the scope of the area used as a prototype is a mosque in Tangerang City. This application display a map that can find out the location of the nearest mosque from the user's position and also can provide guidance on the nearest path by using the Haversine formula. The usefulness of the Haversine formula is to calculate the distance between two points on earth based on the length of a straight line between two points without ignoring the curvature of the earth [9], [10].

II. METHOD

The stages in conducting this research are:

A. Data collection
The first stage is to collect data on the mosque in the Tangerang City ban area. The mosque data was obtained on the website of the Ministry of Religion.

B. Observation
Observation is the activity of direct observation of the research object, namely the location of the mosque in Tangerang City.

Received: January 04, 2019; Revised: August 18, 2019; Received: August 31, 2019; Published: August 31, 2019

C. Sofware development method

The software development method used in this study is to use the waterfall model. Waterfall model is a systematic and sequential information system development model.

The research method applied in this study is to use the waterfall model. Waterfall model is a systematic and sequential information system development model[11].

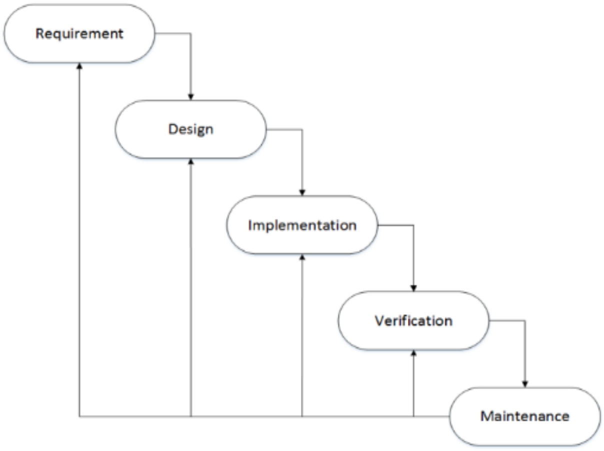


Figure 1. Waterfall Model

The stages of the waterfall research method are as follows:

1. Requirement Analysis

a. Problem Analysis

In this stage, the analysis process is based on daily life. Finding the nearest mosque is a problem that often occurs for someone who often moves outside the room usually will have difficult to search the location of the mosque and how close which roads can be taken to get to the mosque. This application will display a map that can find out the location of the mosque from the user's position and also can provide a guide to the nearest road that will be traveled using the haversine formula. So that users can easily access mosque information via a smartphone.

b. User Analysis

The user of this application is a Muslim, who is on his way and is in an area that is not yet understood the path to the mosque, and uses an Android device.

c. Functional Requirement Analysis

This application has functions that will overcome the problems in this study, which are as follows:

- Finding the nearest location is a function that can search the nearest mosque from a user location, including a route to the mosque. So, the user is expected to no longer difficult to find the nearest mosque while on the road.
- Prayer Time Notification is a functional feature can be useful for a user to get a reminder prayer time so that people do not forget to pray.
- The last functional requirement is the application provides the information on access to the intended mosque whether it can be by car or just a motorcycle, this can help users in deciding the transportation mode to the mosque.

2. Design

In this study, the Unified Modeling Language (UML) is used to design the system. UML is a system development technique that uses graphical language as a tool for documenting and specifying systems [12].

In figure 2 shows a usecase diagram. There are several functions that are done by an actor, such as:

1. Finding the nearest mosque by displaying the nearest mosque in a map and marking the path from the user's position.
2. See the list of nearest mosques from the user's position.
3. See prayer times and find out the current prayer times.

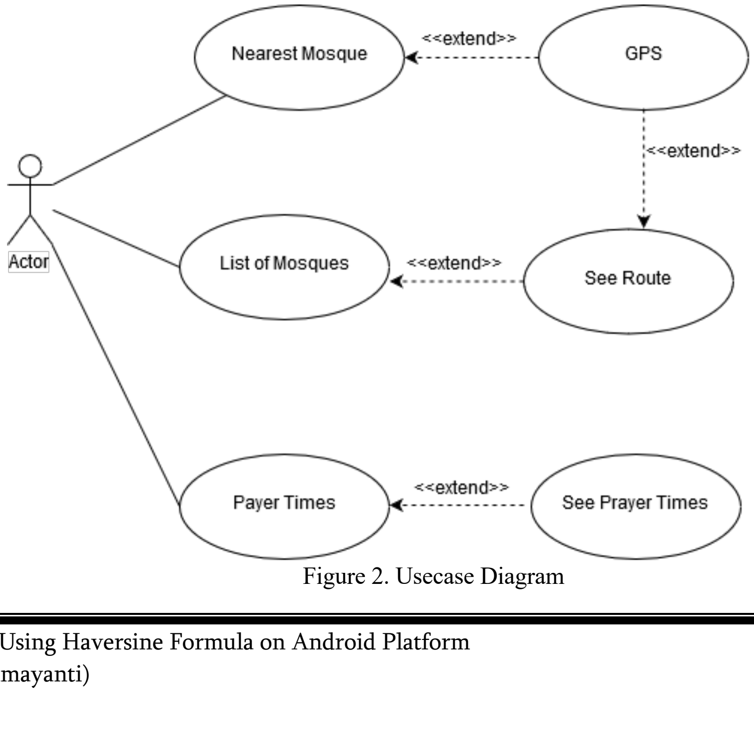
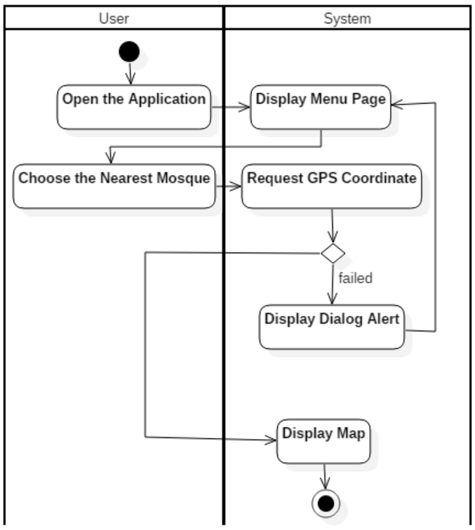


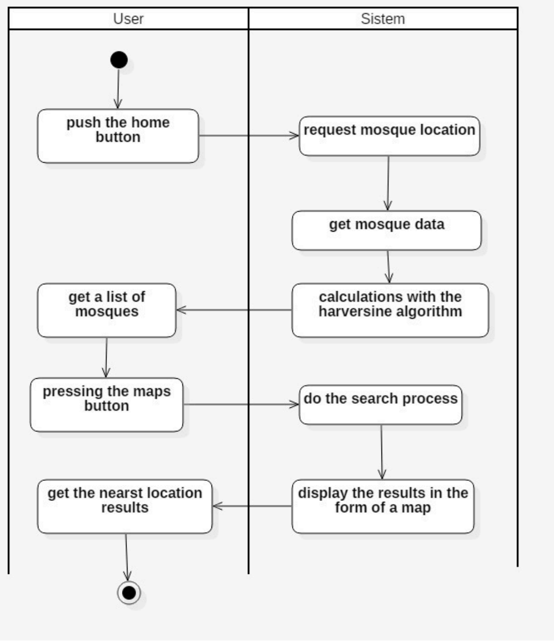
Figure 2. Usecase Diagram

The details in every function that has described in the use case shown in the activity diagram. In figure 3 shows the first workflow when a user opens the application first. When a user opens the application, a user can access the applicaton menus. Next, a user can choose the nearest mosque by clicking the menu button. Then, the system will request the coordinates of the user's position with the Global Positioning System (GPS). If a user can read the position from GPS, the application will display maps. But, if a user cannot access the GPS, the application will show the alert dialog. This activity diagram shows in figure 3.



Gambar 3. Activity Diagram for Finding Nearest Mosque

Figure 4 describes the activity to display a list of mosques. The activity diagram starts from opening the application and displaying the home application menus. The user chooses the mosque list. The system will retrieve json data on the web server. Then the system will display a listing of location of the mosque and calculate the nearest distance using the haversine algorithm. Then, the system displays the result including the details and location of the mosque. Next, the system displays the route with the nearest destination to the mosque.



Gambar 4. Activity Diagram for Displaying List of Mosque

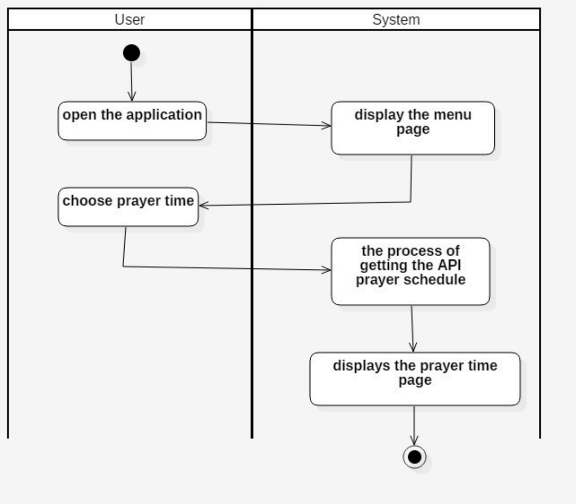

User
System
open the application
display the menu page
choose prayer time
the process of getting the API prayer schedule
displays the prayer time page

Figure 5. Activity Diagram for Displaying Prayer Time

Figure 5 shows the activity for displaying prayer times. The activity diagram starts by opening the application. Then, the system will display the application menus. Next, a user selects the prayer time and the system gets the prayer time from API. So that, user can see the current prayer time.

3 Implementation

Implementation is the stage of implementing a system that has been approved and includes programs that have been created. The system includes the stages of input, process, and output. This is a simple system concept because all systems can have multiple inputs and outputs [8]. In this study, the implementation of the nearest mosque search system was applied using Android Studio and using web services to store mosque latitude and longtitude data. The shortest path search is completed using the haversine formula and using the Google Maps API to determine the route.

4. Verification

In this stage, the testing process is carried out by using the blackbox method. In this blackbox testing phase, the application is tested using USB debugging for android smartphones so that it can immediately find out whether the functions in the application are running well or not and also know in which parts of the error. So, the application can be improved to minimize errors and bugs.

III. RESULT AND DISCUSSION

Haversine Formula is a formula used in navigation that can provide a large circle between two points on the ball / earth from longitude and latitude.[13] By knowing the latitude and longitude of two points on the surface of the earth the distance between the two points can be known.[14]. All equations are used based on the shape of the rounded earth by eliminating the factor that the earth is slightly elliptical. This is a special case of a general formula in the spherical trigonometry of haversine law relating to the sides and angles of spherical triangles.[15][16]

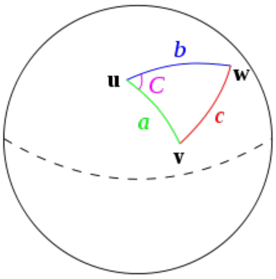


Figure 6. Spherical triangle by the law of haversine formula

The use of the haversine formula is accurate enough to find the nearest location by using the haversine formula which is implemented in MySQL which will be called and then read on the mobile application. The following is the form of the formula of “Haversine Formula”:

$$\Delta lat2 - lat1 \qquad (1)$$

$$\Delta long = long2 - long1\alpha = sin^2\left(\frac{\Delta lat}{2}\right) + \cos(Lat1) \times \cos(Lat2) \times sin^2\left(\frac{\Delta long}{2}\right) \qquad (2)$$

$$c = 2 \times tan2(\sqrt{a, \sqrt{1-a}})) \qquad (3)$$

$$d = R \times c \qquad (4)$$

where :

- R is a radius of the earth 6371(km)
- Δlat is magnitude of the change in latitude
- $\Delta long$ is magnitude of the change in longitude
- c is the calculation of the intersection of the axes
- d is distance (km)

Example the calculation of Haversine Algorithm:

The coordinates of the closest mosque are to utilize the Google Maps API. Simulation of the haversine formula algorithm in calculating the distance between two points is as follows:

1.First coordinate point
(Universitas Mercubuana)

Latitude 1	$= -6.2091756 * \frac{\pi}{180}$
	= -0.1083156188 Radian
Longitude 1	$= 106.7384487 * \frac{\pi}{180}$
	= 1.861992938433333 Radian

2. Second coordinate point
(Masjid Jami Khairul Huda)

Latitude 2	$= -6.2154495 * \frac{\pi}{180}$
	= -0.1084250635 Radian
Longitude 2	$= 106.7351667 * \frac{\pi}{180}$
	= 1.861935685766667 Radian

3. Δlat = -0.1083156188 – (-0.1084250635)
4. $\Delta long$ = 1.8619929384333 – 1.861935685766667
= 0.000057252666633

5. $\alpha = sin^2\left(\frac{\Delta lat}{2}\right) + \cos(Lat1) \times \cos(Lat2) \times sin^2\left(\frac{\Delta long}{2}\right)$
= sin^2 (0.0001094447/2) + cos (-0.1083156188) * cos (-0.1084250635) * sin^2 (0.000057252666633/2)
= 0.000001541849724

6. c = 2 * a sin ($\sqrt{a}$)
= 2 * a sin ($\sqrt{0.000001541849724}$
= 0.0001094447

7. d = R*c
= 6371(Km) * 0.0001094447
= 0.6972721837 Km

The result of the example above is 0.69 Km which is calculated from the coordinates of Universitas Mercubuana to the location of Masjid Jami Khairul Huda.

A. Interface

Implementation of the interface that explains the functions and usability of the program that has been made along with the appearance of the design.

The following is the menu display of the application:

a. Menu Home

This menu contains the main view of the application. On this home, the menu displays the closest mosque, the list of the closest mosques and equipped with a prayer schedule as a reminder when prayer time has arrived. When the user presses one of a button on this page, the system will direct to another page according to the button the user presses. The following is the appearance of the menu page interface.

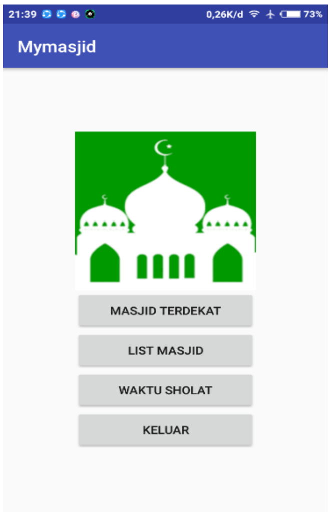


Figure 7. Menu Home

b. Nearest Mosque

On the nearest mosque menu, there is a map display after the user presses the nearest mosque button on the main menu page. This page displays maps and followed by the marker position of the closest mosque from the user's position. When one of the markers is pressed it will display the name of the mosque and the direction button at the bottom. Making this page requires an API to be able to display Google Maps.

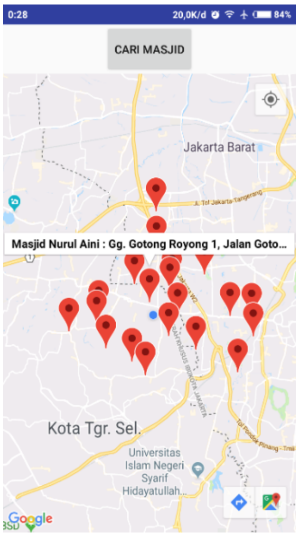
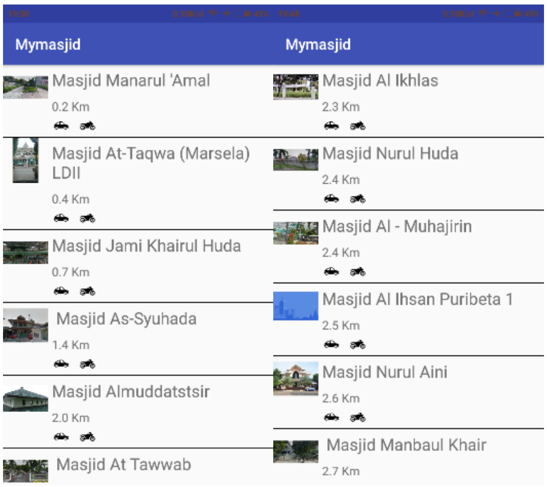

0:28
20,0K/d 84%
CARI MASJID
Jakarta Barat
Masjid Nurul Aini : Gg. Gotong Royong 1, Jalan Goto...
Kota Tgr. Sel.
Universitas Islam Negeri Syarif Hidayatullah
Google

Figure 8. Nearest Mosque

c. List of Mosque

In this menu display a list of the location of each mosque with the nearest distance from the user's location that is using the Haversine method to calculate the nearest distance



Mymasjid

Masjid Manarul 'Amal
0.2 Km

Masjid At-Taqwa (Marsela) LDII
0.4 Km

Masjid Jami Khairul Huda
0.7 Km

Masjid As-Syuhada
1.4 Km

Masjid Almuddatstsir
2.0 Km

Masjid At Tawwab

Mymasjid

Masjid Al Ikhlas
2.3 Km

Masjid Nurul Huda
2.4 Km

Masjid Al - Muhajirin
2.4 Km

Masjid Al Ihsan Puribeta 1
2.5 Km

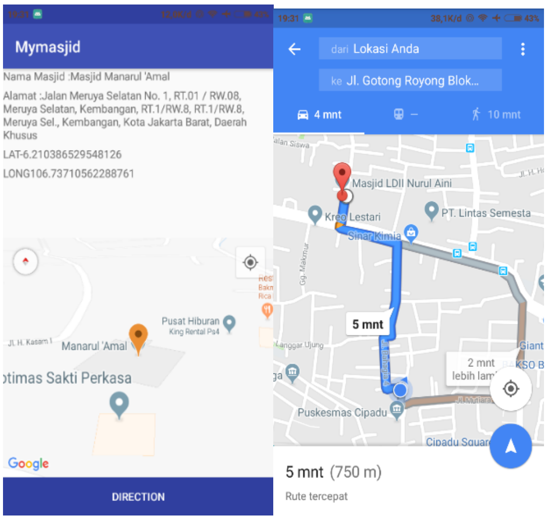
Masjid Nurul Aini
2.6 Km

Masjid Manbaul Khair
2.7 Km

Figure 9. List of Nearest Mosque

d. Maps

This menu displays a map of the mosque that the user has chosen from the list of nearest mosques in which contains the address of the intended mosque, latitude and longitude data as well as directions to the mosque.



Mymasjid

Nama Masjid :Masjid Manarul 'Amal

Alamat :Jalan Meruya Selatan No. 1, RT.01 / RW.08, Meruya Selatan, Kembangan, RT.1/RW.8, RT.1/RW.8, Meruya Sel., Kembangan, Kota Jakarta Barat, Daerah Khusus

LAT-6.210386529548126

LONG106.73710562288761

Figure 10. Location instructions

e. Form data web admin

The web admin data form contains mosque data in the form of mosque name, latitude, and longitude as well as information. In this form, the admin can edit, delete and add data.

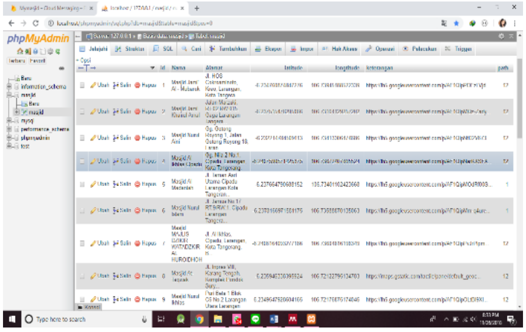


Figure 11. Form data mosque

IV. CONCLUSION

Based on the discussion of the research that has been made, the following conclusions can be drawn, as follows:

1. Has succeeded in making the application to search the nearest mosque in the Larangan, Tangerang City based on Android.
2. Has successfully applied the Haversine Algorithm in the application to search the nearest mosque in Larangan, Tangerang City.
3. This mosque search application generates the closest mileage in order to save time, energy, and costs.
4. This application requires an internet connection and GPS.
5. Database in the nearest mosque application using MySQL database.

Some of the following suggestions and input are expected to provide improvements in further research that can be added to other features, such as other objects, the addition of Google Maps features, and the use of other more effective methods can also be scrutinized to get better results.

V. REFERENCES

[1] B. A. B. Ii and A. S. Maktubah, “ا ا ا راو ة ا اوةاو ة ا ا او هو اوة ا ا او ة ا ا او !ن &ا نا ة ا او ,ا و ة ا ا او ة ا او” pp. 5–23, 1976.

[2] S. Kurniawan, “Masjid dalam lintasan sejarah umat islam,” *J. Islam. Stud.*, vol. 4, no. 2, pp. 169–184, 2014.

[3] A. I. Niculescu, B. Wadhwa, and E. Quek, “Smart City Technologies: Design and Evaluation of An Intelligent Driving Assistant for Smart Parking,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, no. 6, pp. 1096–1102, 2016.

[4] S. G. Fernandez, R. Palanisamy, and K. Vijayakumar, “GPS &amp; GSM Based Accident Detection And Auto Intimation,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 11, no. 1, p. 356, Jul. 2018.

[5] R. H. Putra, F. T. Kusuma, T. N. Damayanti, and D. N. Ramadan, “IoT: smart garbage monitoring using android and real time database,” *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 17, no. 3, p. 1483, Jun. 2019.

[6] K. Curran, “The Locator Framework for Detecting Movement Indoors,” *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 16, no. 1, pp. 390–401, Feb. 2018.

[7] M. A. B. Sidik *et al.*, “Arduino-Uno Based Mobile Data Logger with GPS Feature,” *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 13, no. 1, p. 250, Mar. 2015.

[8] D. Prasetyo and K. Hastuti, “Penerapan Haversine Formula Pada Aplikasi Pencarian Lokasi Dan Informasi Gereja Kristen Di Semarang Berbasis Mobile,” *Skripsi Tek. Inform. Univ. Dian Nuswantoro*, 2015.

[9] Farid and Y. Yunus, “Analisa Algoritma Haversine Formula Untuk Pencarian,” *Ilk. J. Ilm.*, vol. 9, pp. 353–355, 2017.

[10] N. M. Dhanya, “Anti-theft vehicle tracking system using GPS and location prediction,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 6, pp. 2584–2589, 2018.

[11] G. W. Sasmito, “Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal,” *J. Pengemb. IT*, vol. 2, no. 1, pp. 6–12, 2017.

[12] Sri Mulyani, *Metode Analisis dan Perancangan Sistem*. Bandung, 2016.

[13] A. Fauzi, F. Pernando, and M. Raharjo, “Penerapan Metode Haversine Formula Pada Aplikasi Pencarian Lokasi Tempat Tambal Ban Kendaraan Bermotor Berbasis Mobile Android,” *J. Tek. Komput. AMIK BSI*, vol. IV, no. 2, 2018.

[14] H. Nazih and J.-M. Bard, “Microalgae in Human Health,” *Microalgae Heal. Dis. Prev.*, pp. 211–226, 2018.

[15] Yulianto, Ramadiani, and A. H. Kridalaksana, “Penerapan Formula Haversine pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal,” *J. Ilm. Ilmu Komput.*, vol. 13, no. 1, pp. 14–21, 2018.

[16] S. I. Purnawan, F. Marisa, and I. D. Wijaya, “Aplikasi Pencarian Pariwisata Dan Tempat Oleh- Oleh Terdekat Menggunakan Metode Haversine Berbasis Android,” vol. 3, no. 2, pp. 9–16, 2018.