DESIGN AND BUILD OF SEARCHING SYSTEM FOR THE NEAREST FISH SHOP ON AN ORNAMENTAL FISH MARKET WEBSITE USING THE HAVERSINE ALGORITHM

WAWAN GUNAWAN[1], EMIL R. KABURUAN[2], BIKTRA RUDIANTO[3], ANGGI PUSPITASARI[4], BUDI SUDRAJAT[5], DIRGAHAYU ERRI[6], ARIEF NUR HIDAYAH[7]

[1,2,7]Informatics Engineering, Mercu Buana University, Department of Computer Science, Jakarta, Indonesia

[3]Informatics, Nusa Mandiri University, Department of Information Technology, Jakarta, Indonesia

[4,6]Information System, Bina Sarana Informatika University, Department of Informatics & Engineering, Jakarta, Indonesia

[5]Computer Technology, Bina Sarana Informatika University, Department of Informatics & Engineering, Jakarta, Indonesia

E-mail: [1]wawan.gunawan@mercubuana.ac.id, [2]emil.kaburuan@mercubuana.ac.id, [3]biktra.brd@nusamandiri.ac.id, [4]anggi.apr@bsi.ac.id, [5]budi.bst@bsi.ac.id, [6]dirgahayu.dge@bsi.ac.id, [7]41518010106@student.mercubuana.ac.id

ABSTRACT

To address the problem of high shipping costs and long delivery times due to the distance between stores and rare stocks of fish, we have developed a system on Aqua Store ID that allows buyers to find the nearest fish shop based on their address. The system utilizes data from the Aqua Store ID internal database, which includes store data and fish product data. Additionally, we use the haversine algorithm to calculate the distance between the user's coordinates and the store coordinates. The data used in this study is obtained from the Aqua Store ID internal database. It includes information about the stores and their respective fish products. By utilizing the haversine algorithm, we can calculate the distance between the user's location, obtained from the address they input, and the coordinates of the stores in the Aqua Store ID database. To ensure the reliability and functionality of the system, thorough testing was conducted using the black box method. The black box method involves testing the system's functionality without examining its internal code. Predefined steps were followed during the testing process, and the results indicate that the system runs well and successfully provides search results for the nearest fish shop based on the user's address. With the implementation of this system, we expect that buyers on Aqua Store ID can easily locate the nearest fish shop based on their address. This will help solve problems related to shipping costs and delivery times, especially if the buyer is able to find a fish shop that is closer in proximity. By reducing the distance between the store and the buyer, we provide a more efficient and cost-effective buying and selling experience for ornamental fish on Aqua Store ID.

Keywords: *Haversine Algorithm, Shortest Distance Measurement, Buying And Selling Ornamental Fish, Waterfall Method, Black Box Method)*

1. INTRODUCTION

The activity of buying and selling ornamental fish currently uses technology such as the marketplace. One marketplace for buying and selling ornamental fish that is still developing is Aqua Store ID. Aqua Store ID is an online good buying and selling platform or it can be said as a marketplace focused on buying and selling ornamental fish, the platform offers ornamental fish products from various types of fish, food, aquarium pumps, and other supporting products related to ornamental fish. Various ornamental fish shops have registered on the platform to market their products. As of March 2022, there were at least 32 ornamental fish shops with a total of 69 transactions in March.

There are 18 complaints are complaining about the long distance of the store. The complaints are still growing because fish also cannot stay alive for long time. There will be much easier for the buyers if they can find the nearest store to be more efficient and effective. Long store distances cause high shipping costs and long delivery times because the farther the location between the buyer and the seller, the higher the shipping costs [1], while the delivery time affects the condition of the fish which will cause a decrease in oxygen levels during the fish delivery process decoration is done[2]. This is because when the user searches for the type of fish or product he needs the system does not recommend products that appear based on the user's domicile so the products displayed are only based on search keywords with existing products, even if the distance between the store and the user's domicile is far away will still be displayed in the top search order as long as Products that are traded are by the search keywords.

This problem can be solved by designing and building a nearby fish shop search system that implements the haversine algorithm to find the nearest fish shop based on the address entered by the buyer. Haversine is an algorithm used to find the distance between two points based on latitude and longitude. The advantages of this algorithm when compared to other algorithms are, the haversine algorithm has a high level of accuracy, also a lower error percentage, and a simpler way of working [3]–[5]. The following is the formula for the haversine algorithm.

$$\Delta lat = lat2 - lat1$$
$$\Delta long = long2 - long1$$
$$a = sin^2\left(\frac{\Delta lat}{2}\right) + cos(lat1) \times cos(lat2)\ \times sin^2\left(\frac{\Delta long}{2}\right)$$
$$c = 2\ \times atan2\left(\sqrt{a}, \sqrt{1-a}\right)$$
$$d = R\ \times c$$

Information:

R = The radius of the earth is 6371 km

Δ lat = Value of change in latitude

Δ long = Value of change in longitude

c = Results of the calculation of the intersection of the axes

d = Distance in km units[6], [7]

In a previous study [8]implementing the haversine algorithm using latitude and longitude data of users and referral hospitals to get a comparison value with Google Maps of 3 meters.

In the previous study a comparison of methods was carried out to calculate the shortest distance between Euclidean, Haversine, and Manhattan, the results of the study explained that Haversine gets the highest accuracy value of 98.66%, Euclidean of 98.51% and Manhattan of 75.98%. From this study, it was concluded that the Haversine algorithm produces a high accuracy value compared to Euclidean and Manhattan[8].

Whereas in this study, the data needed to implement this algorithm is the latitude and longitude of the shop and the buyer. Store latitude and longitude data are obtained from Aqua Store's internal database, while buyer's latitude and longitude data are obtained when buyers want to find the nearest fish shop by entering the buyer's address. In this study, the Here Map library will be used to display maps on the website. Here Map is an API service provided by Here to display maps and get latitude and longitude data via the address sent.

2. LITERATURE REVIEWS

Based on the problem stated, haversine algorithm used in previous research including: finding the nearest hospital for Covid-19 [9]; looking for souvenir store [10]; bird’s contest location [11]; safe meeting point of natural disaster[12]; identified school zone[13]; smart parking system[14]; patient monitoring systems[15]; fire emergency responses [16]; navigation systems[17]; public transportation routes [18]; and findings pharmacy stores [19]. The Haversine method is shown to produce high accuracy values, indicating its effectiveness for distance calculations.

There are two commonly used algorithms; Djikstra Algorithm and Haversine Algorithms. The two are usually used to find the shortest distance. However, in this research we use haversine algorithms because it fits with our needs.

The research questions that would be answer are:

- How to build an ecommerce for decorative fish with haversine algorithm to find nearest store?
- How to find the shortest route to the decorative fish using the haversine algorithm?

3. RESEACH METHODS

This study uses a type of quantitative research using the waterfall system development method. The waterfall is a method for developing systems that are carried out in a structured and sequential manner[20]. The following are several stages of the waterfall method that will be applied to this study.

2.1 Requirement Analysis

The stage where the researcher analyzes what needs will be used to create the system. In this study, several needs are needed, namely. Dataset of product data and store data registered in the Aqua Store ID database. API Key Here Map is used to implement the Here Map library. Git which stores the code from the frontend and backend Aqua Store ID. And access the Aqua Store ID server to process the code upload to the Aqua Store ID website.

2.2 System Design

The stage of making a system design will be used as an illustration for implementing the system to be made. At this stage, the researcher made several designs such as use case diagrams, activity diagrams, sequence diagrams, class diagrams, flowcharts, and the display design for the nearest store search page.

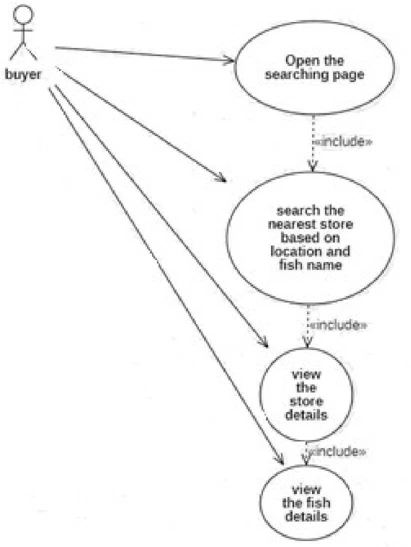

Figure 1: Use Case Diagram

Based on Figure 1, is a use-case diagram of the closest fish shop search system. In the picture, there is 1 actor, namely the buyer. Buyers are people who will buy fish on the Aqua Store ID website. In the diagram above, buyers can open the nearest store search page, can search for the nearest store based on the location and fish they want to search for, can view the store, and can see the fish that appear based on the fish they are looking for.

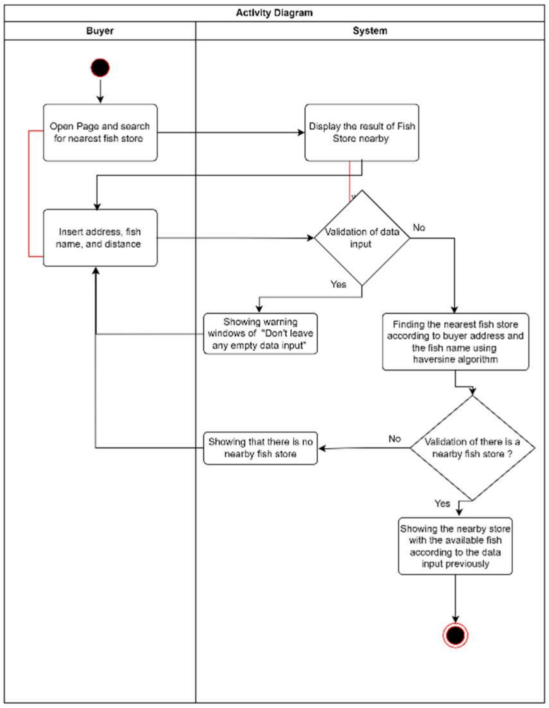
Activity Diagram
Buyer
System
Open Page and search for nearest fish store
Display the result of Fish Store nearby
Insert address, fish name, and distance
Validation of data input
No
Yes
Showing warning windows of "Don't leave any empty data input"
Finding the nearest fish store according to buyer address and the fish name using haversine algorithm
Showing that there is no nearby fish store
No
Validation of there is a nearby fish store ?
Yes
Showing the nearby store with the available fish according to the data input previously

Figure 2: Activity Diagram

Figure 2 is an activity diagram made for the nearest fish shop search system. In this diagram, the buyer will open the nearest fish shop search page, then the system will display a store search page based on the design that has been made. Then the buyer enters the address, fish name, and distance limit, after that the system will validate whether there are empty inputs or not, if there are blanks it will return a message that the data cannot be empty. If not, the system will search for the nearest fish shop using the haversine algorithm based on the address and name of the fish entered by the buyer. After that the system validates whether the store data is empty or not, if it is empty the system will display a message that there is no nearest shop around the address filled in, if it is not empty then the system will display the nearest store along with the fish in that shop based on the address and name of the fish sought by buyers.

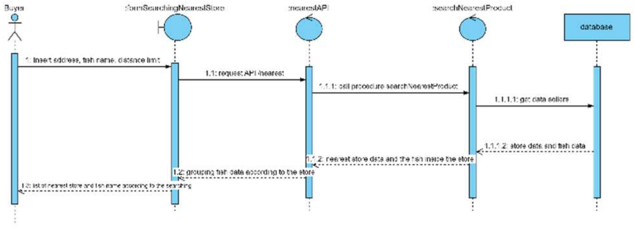
Buyer
:formSearchingNearestStore
:nearestAPI
:searchNearestProduct
database
1: Insert address, fish name, distance limit
1.1: request API /nearest
1.1.1: call procedure searchNearestProduct
1.1.1.1: get data sellers
1.1.1.2: store data and fish data
1.1.2: nearest store data and the fish inside the store
1.2: grouping fish data according to the store
1.3: list of nearest store and fish name according to the searching

Figure 3: Sequence Diagram

Figure 3 is a sequence diagram of the system to be created. In this diagram, there is 1 actor, namely the buyer who will enter address data, fish names, and distance limits through the nearest store search form. The entered data will be sent to the API with the /nearest endpoint. The API will call the searchNearestProduct procedure, which contains a query to retrieve sellers and product data from the Aqua Store ID database. The parameters sent to the procedure are the buyer's latitude data, the buyer's longitude, the fish name, and the distance limit. This procedure will call the getDistance function which functions to calculate the distance between the store and the buyer by calculating the haversine algorithm. After the distance is obtained, the procedure will sort with an ascending query. After that, the procedure will return the nearest store data and products in that store based on the address and name of the fish that the buyer is looking for.

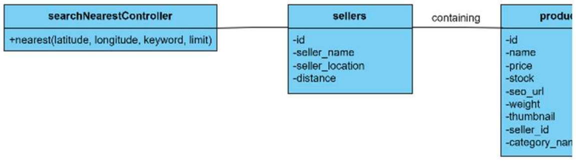

Figure 4: Class Diagram

Figure 4 is a class diagram of the system to be created. In the diagram, there is a searchNearestController that has the nearest method that functions to retrieve store and product data based on the location points entered in the form of latitude, longitude, keyword, and limit. This method will return a value in the form of a shop that has fish based on the address and name of the fish that the buyer is looking for.

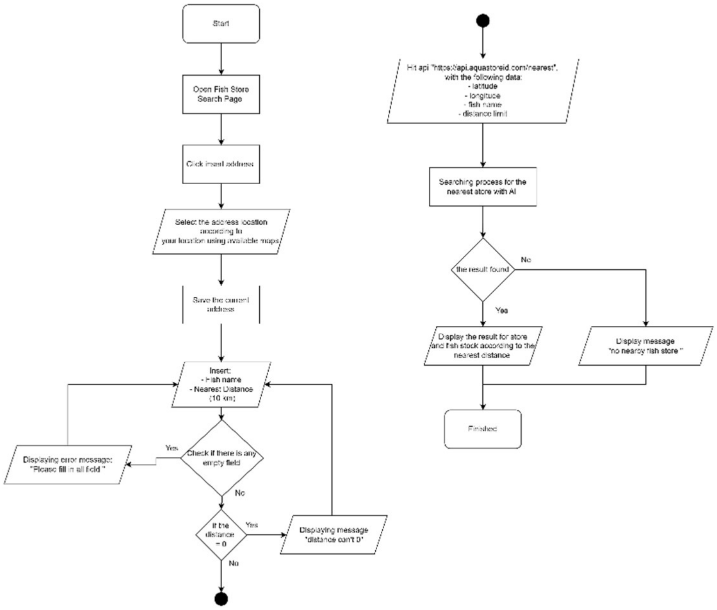
Start
Open Fish Store Search Page
Click insert address
Select the address location according to your location using available maps
Save the current address
Insert:
- Fish name
- Nearest Distance (10 km)
Check if there is any empty field
Yes
Displaying error message: "Please fill in all field"
No
If the distance = 0
Yes
Displaying message "distance can't 0"
No
Hit api "https://api.aquastoreid.com/nearest", with the following data:
- latitude
- longitude
- fish name
- distance limit
Searching process for the nearest store with AI
the result found
No
Yes
Display the result for store and fish stock according to the nearest distance
Display message "no nearby fish store"
Finished

Figure 5: Flowchart

Figure 5 is a flowchart of the nearest fish shop search system that will be made. In the picture above the system, the flow will start when the buyer opens the search page for the nearest store. After that, the buyer enters the address by clicking the input address button, if the button is clicked a popup will appear in the form of a map. In this popup, buyers will search for addresses by searching from the search field or clicking on location points on the map. To bring up the map use the map service from Here Map. Here Map is an API service provided by Here to display maps and get latitude and longitude data via the address sent. The reason researchers use Here Map is that it can be used free of charge and the documentation for using this library is easy to understand and quite complete. After that, the buyer will enter the name of the fish he wants to find and the distance limit he wants to determine. After the buyer enters these 3 data there is a process of checking the columns and distance limits so that the buyer is not wrong in filling in the entered column. If there is no empty data, a request will be made to the API which will be made to the backend code with the endpoint "https://api.aquastoreid.com/nearest" by sending latitude, longitude, fish name, and distance limit data. The API will search for the nearest fish shop using the haversine algorithm based on the fish you are looking for and according to a predetermined distance limit.

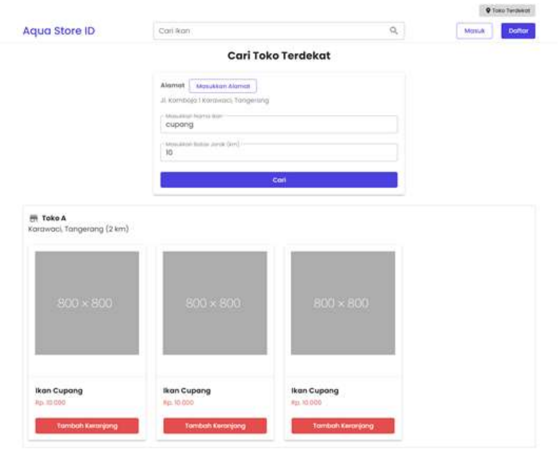

Figure 6. Page Design

Figure 6 is the page design of the nearest fish shop search feature that will be made. When a buyer visits the Aqua Store ID website for the first time, the initial appearance will not be like the one above, the buyer must click the "Nearest Store" button located at the top right of the website, then the buyer will be directed to a page like a picture above. In Figure 2.3 there is a search field where you have to fill in the address, fish name, and distance limit with an initial value of 10 km. Buyers must fill in these 3 columns to find the nearest store based on the name of the fish they want to find. If the buyer has clicked the search button, the shop will appear along with the fish in the shop. And will also display district, city, and distance information from the store.

2.3 Implementation

The stage where the researcher implements the system that has previously been designed. This stage is in the form of a code generation process. The code generated is in the form of code for the front end and back end of Aqua Store ID. A frontend is code written for the front view of the website, while the backend is code written for the back of the website whose purpose is to manage data.

2.4 Testing

The stage where the system that has been implemented will be tested to find deficiencies in the system. At this stage, testing will be carried out using the black box method where this test focuses on the functionality side of the website. Black box is a testing method that focuses on the functionality of a system, this test is carried out to improve the system without any errors occurring[21].

2.5 Operation & Maintenance

The stage where the system that has been tested will be installed on the Aqua Store ID website and maintenance will be carried out to detect errors that were not found at the testing stage.

3. RESULT AND DISCUSSION

The following are the results of system testing using the black box method that was carried out at the testing stage. The results of the test can be seen in Table 1.

Table 1: Black Box Testing Results.

Description	Expected results	Conclusion
Open the search page for the nearest store	Displays a request for permission to access the location and displays the nearest shop page according to the design that has been made	Success
Clicking the button when any or all of the fields are empty	Displays an error notification "*Harap isi semua kolom*!"	Success
Clicking the enter address button	Displays a popup in the form of maps to find addresses	Success
Choose an address via the map	The address pin will change according to the point that has been clicked	Success
Choose an address through the address search field	The address pin will change according to the address point you are looking for	Success
Clicking on the save address button	The address popup will close and the address will be saved	Success
Clicking the search button when all the fields are filled in	Displays a list of the closest shops and the fish in the shop according to the fish you are looking for. If the number of stores is more than 5, pagination will appear	Success
Clicking the pagination button	Store listings will change	Success
Clicking the search button with fish names that are not registered in the database	Displays an error notification "*Tidak ada toko yang berjarak 10 km di sekitar alamat mu*"	Success

Based on Table 1, there are 3 columns consisting of descriptions, expected results, and conclusions. From the table, it can be seen that all experiments carried out using the black box method on the system that has been made have been successful and in line with expectations.

4. CONCLUSION

Application testing and algorithm testing that has been carried out on the nearest fish shop search system on the Aqua Store ID website. So it can be concluded several things, namely: The system that has been built to find the nearest fish shop on Aqua Store ID using the haversine algorithm can run well and without any problems. The design of a search system for the nearest fish shop was successfully carried out by applying the waterfall software development method and implementing the haversine algorithm. From the results of testing the system can be used to find the nearest fish shop based on the address point entered by the buyer.

REFERENCES:

[1] R. Holwati, T. Setyowati, and T. D. Reskiputri, "Pengaruh Viral Marketing, Online Marketing, Dan Store Location Terhadap Keputusan Pembelian (Studi kasus Pada UMKM Arsa Holtikultura Desa Karangsono)," 2021.

[2] S. Supriyanto and L. Dharmawantho, "EFEKTIVITAS SISTEM PENGANGKUTAN IKAN MENGGUNAKAN SISTEM BASAH," *Buletin Teknik Litkayasa Akuakultur*, vol. 19, no. 2, p. 105, Nov. 2021, doi: 10.15578/blta.19.2.2021.105-108.

[3] B. Priambodo, N. Ani, and Y. Jumaryad, "Predict Next User Location to Improve Accuracy of Mobile Advertising," *J Phys Conf Ser*, vol. 1175, p. 012099, Mar. 2019, doi: 10.1088/1742-6596/1175/1/012099.

[4] S. A. Prakoso and A. Wahab, "Sistem Informasi Pencarian Rute dan Informasi Transjakarta Menggunakan Metode Haversine Formula 'Berbasis Andorid,'" 2019. [Online]. Available: https://jurnal.ikhafi.or.id/index.php/jukomika/240

[5] S. I. Purnawan, F. Marisa, and I. Dharma Wijaya, "Aplikasi Pencarian Pariwisata Dan Tempat Oleh-Oleh Terdekat Menggunakan Metode Haversine Berbasis Android," *JIMP-Jurnal Informatika Merdeka Pasuruan*, vol. 3, no. 2, 2018, [Online]. Available: https://maps.google.com

[6] R. T. Jurnal, "APLIKASI PENCARIAN USTADZ UNTUK WILAYAH DKI JAKARTA MENGGUNAKAN ALGORITMA HAVERSINE FORMULA BERBASIS ANDROID," *PETIR*, vol. 9, no. 2, pp. 102–111, Jan. 2019, doi: 10.33322/petir.v9i2.174.

[7] Chandra Husada, Kristoko Dwi Hartomo, and Hanna Prillysca Chernovita, "Implementasi Haversine Formula untuk Pembuatan SIG Jarak Terdekat ke RS Rujukan COVID-19," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 5, pp. 874–883, Oct. 2020, doi: 10.29207/resti.v4i5.2255.

[8] Y. Miftahuddin, S. Umaroh, and F. R. Karim, "PERBANDINGAN METODE PERHITUNGAN JARAK EUCLIDEAN, HAVERSINE, DAN MANHATTAN DALAM PENENTUAN POSISI KARYAWAN," *Jurnal Tekno Insentif*, vol. 14, no. 2, pp. 69–77, Aug. 2020, doi: 10.36787/jti.v14i2.270.

[9] Chandra Husada, Kristoko Dwi Hartomo, and Hanna Prillysca Chernovita, "Implementasi Haversine Formula untuk Pembuatan SIG Jarak Terdekat ke RS Rujukan COVID-19," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 5, pp. 874–883, Oct. 2020, doi: 10.29207/resti.v4i5.2255.

[10] S. I. Purnawan, F. Marisa, and I. Dharma Wijaya, "Aplikasi Pencarian Pariwisata Dan Tempat Oleh-Oleh Terdekat Menggunakan Metode Haversine Berbasis Android," *JIMP-Jurnal Informatika Merdeka Pasuruan*, vol. 3, no. 2, 2018, [Online]. Available: https://maps.google.com

[11] D. D. Prihantoro and M. I. Wahyuddin, "Implementasi Algoritma Haversine Formula dan Location Based Service Pada Aplikasi Pencarian Lokasi Bird Contest Berbasis Android," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 1, p. 663, Jan. 2022, doi: 10.30865/mib.v6i1.3546.

[12] A. N. Abadi Nugroho, "Penerapan Metode Haversine Formula Untuk Penentuan Titik Kumpul pada Aplikasi Tanggap Bencana,"

METIK JURNAL, vol. 4, no. 2, pp. 69–75, Dec. 2020, doi: 10.47002/metik.v4i2.190.

[13] D. Ikasari, W. Ikasari, and R. Andika, “Implementation of Haversine Formula to Determine the Shortest Path Using Web Based Application for a Case Study of High School Zoning in Depok,” *American Journal of Software Engineering and Applications*, vol. 10, no. 2, p. 19, 2021, doi: 10.11648/j.ajsea.20211002.11.

[14] R. R. Safitri, A. Pratiarso, and A. Zainudin, “Mobile-based Smart Parking Reservation System with Rate Display Occupancy Using Heuristic Algorithm and Haversine Formula,” in *2020 International Electronics Symposium (IES)*, IEEE, Sep. 2020, pp. 332–339. doi: 10.1109/IES50839.2020.9231654.

[15] E. R. Pratama, F. Renaldi, F. R. Umbara, and E. C. Djamal, “Geofencing Technology in Monitoring of Geriatric Patients Suffering from Dementia and Alzheimer,” in *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, IEEE, Sep. 2020, pp. 106–111. doi: 10.1109/IC2IE50715.2020.9274637.

[16] M. Sulaiman, H. Liu, M. Binalhaj, W. W. Liou, and O. Abudayyeh, “GIS-Based Automatic Flight Planning of Camera-Equipped UAVs for Fire Emergency Response,” in *2020 IEEE International Conference on Electro Information Technology (EIT)*, IEEE, Jul. 2020, pp. 139–144. doi: 10.1109/EIT48999.2020.9208329.

[17] N. Chan Soe and T. Lai Lai Thein, “Haversine Formula and RPA Algorithm for Navigation System,” *International Journal of Data Science and Analysis*, vol. 6, no. 1, p. 32, 2020, doi: 10.11648/j.ijdsa.20200601.14.

[18] K. Poonsilp*, “Multi-Modal Route Recommender System for Bangkok Public Transportation,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, pp. 1380–1384, Mar. 2020, doi: 10.35940/ijrte.F7732.038620.

[19] B. I. Akhigbe, O. J. Ososanya, and E. A. Olajubu, “A Mobile Based Pharmacy Store Location-aware System,” *Journal of information and organizational sciences*, vol. 46, no. 1, pp. 213–236, Jun. 2022, doi: 10.31341/jios.46.1.12.

[20] M. A. Rizki, V. Yasin, and A. S. Rini, “PERANCANGAN SISTEM PENGENDALIAN KEHADIRAN DAN MELACAK LOKASI BERBASIS WEB DI KANTOR NOTARIS P.SUANDI HALIM DENGAN METODE WATERFALL,” *Jurnal Widya*, vol. 2, no. 1, pp. 43–59, 2021, [Online]. Available: https://jurnal.amikwidyaloka.ac.id/index.php/awl

[21] P. Saman and C. Indah Ratnasari, “Pengujian Black Box Pada Aplikasi Pembelajaran Bahasa Mandarin Berbasis Android Black Box Testing on an Android-Based Mandarin Learning Application,” *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, vol. 4, no. 1, pp. 10–22, 2022.