Haversine Formula To Find The Nearest PetShop

Rakhmat Purnomo1*, Tri Dharma Putra2, Hadi Kusmara3, Wowon Priatna4, Fajar Mukharom5
1,2,3,4,5Computer Science, Universitas Bhayangkara Jakarta Raya, DKI Jakarta
e-mail: *rakhmat.purnomo@dsn.ubharajaya.ac.id, tri.dharma.putra@dsn.ubharajaya.ac.id, 
3hadi.kusmara@dsn.ubharajaya.ac.id, 4wowon.priatna@dsn.ubharajaya.ac.id, 
5fajar.mukharom@mhs.ubharaja.ac.id

Abstract
The population of cats and dogs is increasing, whether they are kept or not. The public’s awareness to raise these animals needs to be supported by a business place for keeping and buying and selling pet food or a PetShop. People who keep animals often have difficulty (1) finding the nearest PetShop, (2) access to the location, (3) knowing what services are provided, even (4) knowing whether there is a willingness to consult animal care and make a consultation appointment. This study aims to bring together PetShop and the community in the care and maintenance of animals, especially cats and dogs in Bekasi City. The system development life cycle method used is WaterFall. The WaterFall method consists of 5 stages, namely: Communication, Planning, Modeling, Construction, and Deployment. On stage Contraction, the Haversine Formula is inserted to find the nearest PetShop and with the help of the Google Maps API, a route to the PetShop is determined from the application user. Unified Modeling language (UML) diagrams are used during the analysis and system design processes such as Use Case Diagrams, Activity Diagrams, Sequence Diagrams, Class Diagrams, Data Models, and User Interfaces. The result of this research is the ZuPet Web application which was built according to the Waterfall system development methodology, and the Haversine Formula has been inserted to optimize the search for the shortest route. The functionality of the application has been tested using the BlackBox Testing method. The application can be accessed at the address https://zupet.my.id.

Keywords -- Haversine Formula, Web ZuPet, PetShop, Waterfall, UML.

1. INTRODUCTION

The population of dogs and cats is increasing every year. This is because there are still many wild animals and also a lot people is increasing in keeping these animals. It was recorded that in the Capital City of DKI Jakarta by the DKI Jakarta Food, Maritime and Agriculture Security Service (KPKP), when vaccinating in 2020 there were around 42,000 rabies-transmitting animals, which were dominated by cats and dogs [1]. In line with these data, Bekasi City has a large population of both domesticated and wild animals. Many residents keep animals, especially cats and dogs. This has also prompted some people to start a PetShop business to support their needs for food, medicine, vitamins, and animal health consultation. In addition, several PetShops in Bekasi have also offered various services for animal hygiene, health care, and animal care.

Based on data from Google Maps and Google Business Site, there are 162 Pet Shops spread over 12 sub-districts, namely Bantar Gebang sub-districts (2), West Bekasi (23), South Bekasi (28), East Bekasi (25), North Bekasi (19), Jatiasih (16), Jatisampurna (2), Medan Satria...
The number of PetShop locations is also based on the needs of people who have lots of pets, especially cats and dogs. However, for the community there are difficulties in finding the nearest PetShop location when needed.

The method for finding the location of the nearest object that is often used by researchers is using the Haversine Formula. The Haversine Formula will calculate the distance between two points by utilizing the latitude and longitude data of the starting point and end point of the location [2]. Research conducted by Indah Setyorini [3], uses the Haversine Formula on an Android-based application for the nearest mosque in Tangerang City so that people can save time, energy and costs when looking for mosques. In the health sector, research conducted by Rahmi Hidayati [4] uses the Haversine Formula to find the nearest health facility in the city of Pontianak. Other researchers are also with the same topic to find the closest distance, using the Haversine Formula used to find locations for android-based bird contests [5], looking for tourist attractions for tourist recommendations [6], looking for schools according to the closest zoning [7], looking for futsal fields [8], looking for vehicle tire patches [9], looking for ustaz for religious lectures [10], and looking for places for tourists [11]. Many studies have used the Haversine Formula to determine the closest distance.

Based on the research that has been done using the Haversine Formula in various fields, the researcher wants to contribute to develop the knowledge of the Haversine Formula to find the nearest Pet Shop in Bekasi City.

2. RESEARCH METHOD

The research method used follows the Software Development (SD) method, namely WaterFall. Figure 1. is the Waterfall Model which is a classical model that is systematic or sequential in building Software Development [12]. This model is included in the general model of software engineering and was first introduced by Winston Royce around 1970 so it is often considered ancient, but is the most widely used model in software development.

There 5 stages in WaterFall Model, namely:

1. Communication (Project Initiation & Requirements Gathering)

Communication with potential users is very important to understand and achieve the objectives that achieved before starting any technical work. The result of this communication is in the form of project initialization, such as analyzing the problems that occur, collecting the necessary data and helping to define the characteristics and functions of the software. Additional data collection can also be taken from books, articles in journals, and the internet.

At this stage, PetShop data and data from people who have cats and dogs are collected. PetShop data is taken from Google Maps and Google Business Site which has been
registered by PetShop in Bekasi City. After that, the sampling technique was used. The researcher uses incidental sampling, which is a sampling technique based on coincidence, that is, anyone who coincidentally meets the author can be used as a sample, if it is deemed that the person who happened to be met is suitable and in accordance with the criteria as a data source. In this study, the sample calculation according to the Slovin formula [13], was used, Please take a look on formula number (1), namely:

\[
Sample \, Size \, (n) = \frac{N}{1 + Ne^2}
\]  

Where:
- \( n \) = total sample population
- \( N \) = size of population that is known
- \( e \) = margin of error from the size of population

1) PetShop Sample Population

In this research, the author sets the margin of error, which is as big as 14%. So that population sample of PetShop is known, namely:

\[
n = \frac{162}{1 + 162(0,14)^2} = \frac{162}{3,2} = 50,625 \text{ rounded up to be } 51
\]

From the total population of PetShop exists in Bekasi City, that is 162 PetShop, then population sample that can be taken from this research, by margin of error 14%, using Slovin Formula with total data is 51 PetShop.

2) Respond Sample Population

In this research, the author sets the margin of error, which is as big as 7%. Then we get the size of population sample, namely:

\[
n = \frac{2,543,676}{1 + 2,543,676(0,07)^2} = \frac{2,543,676}{12.465} = 204,06 \text{ rounded up to be } 205
\]

From the population of inhabitants in Bekasi City, that is, 2,543,676 people, then the population sample that was taken in this research with margin of error 7%, by using Slovin Formula is 205 responds.

Based on these data, the researcher made a UseCase Diagram to make it easier to see the system as a whole. Figure 2. is a UseCase Diagram. UseCase diagrams are brief information about what actors can do in the application [14], where there are 2 actors namely admin and user (user). Admin can manage users, manage PetShop and the services provided, and manage articles. Users can search for the nearest PetShop, view articles and tips, and make appointments with the PetShop.
2. **Planning (Estimating, Scheduling, Tracking)**

The next stage is the planning stage which includes estimating the technical tasks to be carried out, the risks that can occur, the resource requirements in making the system, the product to be produced, work scheduling plans, and tracking in the process of working on the system. At this stage, the system specifications for users are determined, namely:

1) **Software Needed**
   - Operating System: Android/iOS
   - Minimal Version: Android version 5.0/iOS version 10.0
   - Supporting Application: Google Chrome, Safari or other Web Browser

2) **Hardware Needed**
   - Memory/RAM: 2 GB
   - Disk Capacity: 8 GB
   - Compatibility: GPS (Global Positioning System)

System specification for cloud’s based server which is known as VPS (Virtual Private Server), namely:

1) **Software Needed**
   - Operating System: Linux Ubuntu version 18.04
   - Web Server: Nginx version 1.14.0
   - Database: MySQL version 5.7.34
   - Front End Packet: React JS
   - Back End Packet: Express JS (Node JS)

2) **Hardware Needed**
   - Processor: Intel Core Processor (Broadwell, IBRS)
   - Memory/RAM: 1 GB
   - Disk Capacity: 20 GB
3. **Modeling (Analysis & Design)**

This stage is a system architecture design and modeling phase that focuses on designing data structures, software architectures, interfaces, and program algorithms. This phase aims as a big picture of what will be done.

At this stage, after the planning process is carried out, it will produce an Activity Diagram. There are 11 Activity Diagrams created, as an example, Figure 3. is the Nearest PetShop Activity Diagram. The user opens the application, then the system will display the main page. Users can see a list of PetShops along with their distance. Users can choose one of the PetShops so that complete information about the nearest PetShop can appear.

![Activity Diagram of The Nearest PetShop](image)

Figure 3. Activity Diagram of The Nearest PetShop
Figure 4. Sequence Diagram of the Nearest PetShop

Figure 4. is the Closest PetShop Sequence Diagram. Sequence Diagram describes the behavior of objects in use case diagrams by explaining the life time of objects with messages sent and received [14]. The picture shows the relationship between the Actor and the Interface, Controller, and Data.

Figure 5. Class Diagram
Figure 5. shows Class Diagram which describe the system structure as Class-Class which will be made to built system [14]. Class has attribute and operation. Figure 6. is a Data Model that describes data entities that are interconnected [14]. The data model is created to organize the existing data elements and connect them so that they can become relationships between several data tables. The data model becomes one of the references in implementing the next database, to determine the relationship between tables.

4. Construction (Code & Test)

This construction stage is the process of translating the design form into machine-readable code or form/language. When the coding Activity is complete, the system and code will be tested as a validation stage. The goal is to find the probability of this things happening.

At this stage, the application of the haversine formula is explained. Figure 7. is the application of the Haversine Formula on the ZuPet Web Application. The Haversine Formula is applied to a web application created to find the distance between 2 locations, namely the user's location and the location of the PetShop (destination). The algorithm is processed by the database by taking the latitude and longitude values from the user's location. The data obtained is then entered into a SQL query to get the value of the distance between 2 locations.

To be able to calculate the distance, the user must allow access to the location of the device used, otherwise the Haversine Formula calculation process cannot be carried out.
Figure 7. Flowchart for Implementation of Haversine Formula on Web Zupet Application

Figure 8. System Development Block
5. Deployment (Delivery, Support, Feedback)

The Deployment stage is the stage of implementing the system that has been made to the user or customer, regular software maintenance, software improvement, software evaluation and software development. So that the system used can continue to run and develop according to its function.

Figure 8. is a System Development Block. This picture shows the relationship of the Haversine Formula on the ZuPet Web Application. The input block serves to get the info from the user and the required preferences. The Process Block contains the Haversine Formula which is integrated with the ZuPet Web Application and PetShop data in the Bekasi City. The Process Block is a list of the closest PetShop recommendations and routes to the location.

3. RESULTS AND ANALYSIS

3.1. Haversine Formula

Haversine was discovered in 1835 by Prof. James Inman [15]. Rios first used Haversine in his research on the Main Problems of Nautical Astronomy [16]. Haversine is used to determine the distance between stars [8]. Furthermore, the Haversine Formula is used to calculate the distance between two points on the earth's surface using latitude and longitude data as input. The Haversine Formula is used based on the shape of the earth which is spherical like a sphere, and is a special form of spherical trigonometry that deals with the sides and angles of triangles in the plane of the sphere.

The Haversine Formula will be used in calculating the distance between 2 location points based on GPS (Global Positioning System). In this case, it is the GPS point between the user's location and the GPS point of the destination location, namely PetShop. The Haversine Formula is an important equation in navigation that gives the distance of a great circle between two points on the surface of a sphere (earth) based on longitude and latitude.

\[
d = 2r \arcsin\left(\sin^2\left(\frac{\text{lat}_2 - \text{lat}_1}{2}\right) + \cos(\text{lat}_1) \times \cos(\text{lat}_2) \times \sin^2\left(\frac{\text{long}_2 - \text{long}_1}{2}\right)\right)
\]

(2)

Based on the theory that the shape of the earth is not perfectly round, the Haversine Formula according to R. Chopde and Nichat (2013) can use the following equation [2]:
Information:
- r = earth’s radius is 6,371 (km)
- lat1 = latitude of point 1 or the position of user (radian)
- long1 = longitude of point 1 or the position of user (radian)
- lat2 = latitude point 2 or the destination location (radian)
- long2 = longitude point 2 or the destination location (radian)
- d = distance between 2 points (km)

From formula (2), to determine the closest distance to the user’s location, the user must have several comparison destinations. The location with the closest distance can be determined based on several existing destination locations, then it will be selected based on the shortest distance that the user can travel. For the calculation of the Haversine Formula, it will be carried out on a MySQL query (database) on the system, then the results will be sent in JSON format so that the calculation process and data transfer takes place quickly.

The latitude and longitude data of the destination location is in the form of PetShop location data in Bekasi City. The data is obtained from Google Maps based on data and location information that has been registered by each PetShop and has been verified by Google. The data is stored in the database to be processed by the Haversine Formula when the user accesses and sends location information in the form of latitude and longitude from the device used.

3.2. Implementation of Haversine Formula on Application

The process of obtaining the distance value between the user and the PetShop, the system communicates between the front end and back end (database) to exchange data. The data taken from the user’s location will be processed by the database in the form of a MySQL query until a new value is obtained which is called distance. The formula line of the Haversine Formula in the mysql query program that is made is as follows:

```
((acos(sin((:latitude_user)*pi()/180)) * sin(`latitude_PetShop`*pi()/180)) +
cos((:latitude_user)*pi()/180)) * cos(`latitude_PetShop`*pi()/180)) * cos(((:longitude_user) -
`longitude_PetShop` ) * pi()/180))))
```

In order to know the distance between the two points, the latitude and longitude values of the two locations are needed. These needs can be met by the database, by writing a complete mysql query as follows:

```
SELECT *,
((acos(sin((:latitude_user)*pi()/180)) * sin(`latitude_PetShop`*pi()/180)) +
cos((:latitude_user)*pi()/180)) * cos(`latitude_PetShop`*pi()/180)) * cos(((:longitude_user) -
`longitude_PetShop` )) * pi()/180))) * 180/pi()) * 60 * 1.1515 * 1.609344) as distance FROM
`PetShop` ORDER BY distance ASC;
```

Where :
- latitude_user : obtained from the user's device
- longitude_user : obtained from the user’s device
- latitude_PetShop : obtained from PetShop table database
- longitude_PetShop : obtained from PetShop table database
- 60 : degree to arc minute conversion value
- 1,1515 : degree conversion from nautical miles to miles
- 1,609344 : degree conversion from miles to kilometers

3.3. User Interface

The application implementation is described with the generated Interface. This interface describes the functions and uses of the applications that have been created during the analysis and design process. Here is the application interface.
1) Registration Interface

Figure 10. is the Interface Register. Users are required to register to take advantage of the services on the application. The data to be filled in is full name, email, password, and mobile number. After successfully registering, the user can perform the next step, namely logging in.

2) Login Interface

Figure 11. is the Login Interface. After successfully registering, users can login using the email and password data created during registration.

3) Home Interface

Figure 12. is a Home Interface that contains important information that users need when using the system. Users are required to allow device location access to the system in order to obtain information on the nearest PetShop that is around the user's location.

4) Appointment List Interface

Figure 13. is the Appointment List Interface a user created with PetShop. A list of all appointments will appear, both future appointments and the history of each appointment.

5) Nearby PetShop Search Interface

Figure 14. is the Nearest PetShop Search Interface. There is a PetShop search textBox. The list of PetShops on the page will be sorted using the haversine algorithm and sorted by the smallest distance value or can be called the closest distance.
6) Interface Detail PetShop

Figure 15. is the PetShop Detail Interface. On this page, users can also make an appointment with the PetShop they want to visit by pressing the Make Appointment button.
7) Navigation Interface to the PetShop

Figure 16. is the Navigation Interface to the PetShop. To visit the PetShop, if the user does not know the details of the PetShop location, they can use the navigation feature. The feature is linked to Google Maps and can direct users to go to the PetShop from their location. The following is a display of the navigation feature that can be directed directly to the Google Maps application.
3.4 Testing The Application

Testing the ZuPet Web application in which its functions is to find the nearest location is carried out using the blackbox testing method [18]. The implementation of the test is carried out to test the function of the system that is made whether it can run according to the expected output or results. Table 1 is the test results using BlackBox Testing.

Table 1. Testing The Application of Web ZuPet and Formula Haversine

<table>
<thead>
<tr>
<th>Testing Class</th>
<th>Testing Scenario</th>
<th>Expected Result</th>
<th>Testing Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Searching The PetShop</td>
<td>Opening list of page of the nearest PetShop</td>
<td>Display the list of PetShops in the order of the closest distance from the user</td>
<td>Succeed</td>
</tr>
<tr>
<td></td>
<td>Inputing the name of PetShop on</td>
<td>Display of list PetShop according</td>
<td>Succeed</td>
</tr>
</tbody>
</table>
### Testing Class

<table>
<thead>
<tr>
<th>Testing Scenario</th>
<th>Expected Result</th>
<th>Testing Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>searching field</td>
<td>to the name which search by user</td>
<td></td>
</tr>
<tr>
<td>Doing searching filter based on category that is wanted</td>
<td>Display of PetShop list according to category chose by user</td>
<td>Succeed</td>
</tr>
<tr>
<td>Chosing one of the PetShop from PetShop list</td>
<td>Display information details of the chosen PetShop</td>
<td>Succeed</td>
</tr>
<tr>
<td>Showing PetShop service and feature making appointment</td>
<td>Display list of PetShop services and can make appointment with PetShop</td>
<td>Succeed</td>
</tr>
<tr>
<td>Doing navigation to PetShop location by navigation button</td>
<td>Display is redirected to a google maps page and navigation from the user's location to the PetShop</td>
<td>Succeed</td>
</tr>
</tbody>
</table>

### 3.5 System Publication

Web ZuPet Application are stored on web address, https://zupet.my.id [19]. Public can see research results on that address.

### 4. CONCLUSION

Based on the results of research that has been completed, the conclusions are as follows: The ZuPet Web Application was successfully created using the WaterFall system development method. This application uses the Haversine Formula to determine the closest distance between the user and the location of the PetShop. This application is very useful for users to see the location of the nearest PetShop along with the services provided and can also make an
appointment to visitation. This application requires internet connection and GPS access. This application uses a MySQL database.

Suggestions for further research are the addition of other functions that can make this application more complete, such as involving veterinarians for examination, or the application of other algorithms so that research results can be more optimal.

REFERENCES


