Implementation of Haversine formula for school location tracking

P Dauni*, M D Firdaus, R Asfariani, M I N Saputra, A A Hidayat and W B Zulfikar

Department of Informatics, UIN Sunan Gunung Djati Bandung, Bandung, Indonesia

*popon.dauni@if.uinsgd.ac.id

Abstract. The School Location Information Application is an application developed to produce various information and it can be seen and distributed in order to support school tracking and their locations. School Location Information Application is done by collecting data analysis, design, and implementation stages. This application is made by Android base for clients using Java programming language and the server base on web to process data using HTML and PHP accompanied by SQL as the database manager. However, The Haversine Formula method is used to determine distance between user’s location and school location that they want. The result of this application development is expected to be able to solve some problems related to the school tracking and to support the users to easily find the closest school with available information.

1. Introduction
The advanced and sophisticated technology these days, makes many people get huge benefits, one of them is to get the location and navigation of schools easily only by using a geographic school system application which is made by a Mobile or Smartphone system. We know that this kind of technology mobile or Smartphone become a media to get any kind of information and to communicate with other people. And nowadays, it’s advanced to be a school tracking geographically. This system will provide the information to get the geographic location and to find a way to the school.

The enactment of Law No. 22 of 1999 concerning Regional Autonomy which was revised with Law No. 32 of 2004 concerning Regional Governments causes each region to do education expansion area including in both quality and quantity improvements [1].

The new regulation as the implementation of the Education and Culture Ministry Law Number 17 of 2017 concerning PPDB, that is for the new student admission has implemented a zoning system which is determined by the distance between the location of the student's house and the closest school.

With the GPS (Global Positioning System), it will help us to find the location and route, so we can follow the guide without looking at the map or memorizing certain road’s name. This system can help us to guide the direction of the user's position to certain destination in faster and easier way. Whereas, the tool to find the distance between user’s location and the destination, we can use a method that we call Haversine Formula [2,3].
2. Methodology
The methodology used in this study consists of data collection and system development methods. The data collection includes field and literature studies.

2.1. Data collection

2.1.1. Field study. It consists of observation and interviews. Observation is a data collection technique which is done by research and direct review of the problems taken. Whereas, interview is a technique of collecting data by holding question and answer directly. It has to do with the topic taken.

2.1.2. Literature review. For this method, we take several books as a theoretical foundation material to obtain a statement that can support the preparation of the research.

2.2. Method of development system
Rational Unified Process (RUP) is a software engineering method developed by collecting various best practices found in the software development industry. The main characteristic of this method is to use “use case driven” and iterative approaches to the software development cycle. RUP uses the object oriented concept, with activities that focus on developing models using the Unified Model Language (UML) [4,5].

![Rational unified process architecture.](image)

**Figure 1.** Rational unified process architecture.

Data analysis techniques in making this software use the RUP (Rational Unified Process) development method, which includes several phases including inception, elaboration, construction and transition.

2.3. Data processing and data analysis methods

2.3.1. Data processing. Data processing is done through four stages which include editing, coding/scoring, entry, and tabulating. Editing is the process of checking the amount of complete data so that if there is a discrepancy, it can be completed immediately by the researcher. Tabulating (tabulation) is entering data into certain tables. Data Entry (entering data) is a process of entering data obtained into the database.

2.3.2. Data analysis. In the process of data analysis, the researcher reviews all available data at the Education Department and several sources. They are several stages that have to be done. It starts with collecting data and information needed about school information, identifying existing problems, analysing required system, and learning about some components related to the school location.
information application that will be designed. The last is to create the application design by considering existing needs [5].

2.4. Haversine formula

This formula was first discovered by James Andrew in 1805, and it was first used by Josef de Mendoza y Rios in 1801. The term of haversine itself was created in 1835 by Prof. James Inman.

Josef de Mendoza y Ríos used haversine for the first time in his research on "Main Problems of Astronomical Nautical", Proc. Royal Soc, December 22, 1796. Haversine was used to find distances between stars at the first place.

Haversine Formula has its own law that is all equations are used based on the shape of a spherical earth by eliminating the factor that the earth is slightly elliptical (ellipsoidal factor). This is a special case of a general formula in spherical trigonometry which is related to the sides and angles of a spherical triangle. The calculation of the distance from one point to another on the earth surface is affected by a certain degree of curvature [6-9].

![Figure 2. Haversine's law.](image)

In a ball unit, a "triangle" on the surface of a sphere is defined as large circles connecting three points: u, v, and w on the ball. If the length of the three sides is (from u to v), b (from u to w), and c (from v to w), and the angle of opposite view c is C, so that the law of haversines consists of three, as follow:

\[
\text{haversine}(c) = \text{haversine}(a - b) \\
\sin(a)\sin(b)\text{haversine}(C)
\]  

The Haversine formula serves to find the distance between the user’s location and the destination location by performing calculations with this following:

\[
\Delta\text{long} = (\text{long}2 + \text{long}1) \cos\left(\frac{\text{lat}1 + \text{lat}2}{2}\right) = \cos a \cos b - \sin a \sin b \\
\Delta\text{lat} = (\text{lat}2 - \text{lat}1) \\
a = \sin 2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}1)\cos(\text{lat}2)\sin 2\left(\frac{\Delta\text{long}}{2}\right) \\
d = \sqrt{(a)R}
\]  

Information:
R = the radius of the earth is 6371 (km)
(1 degree = 0.0174532925 radians)
\(\Delta\text{lat}\) = amount of change in latitude (km)
\(\Delta\text{long}\) = magnitude of change in longitude (km)
d = distance (km)
3. Result and discussion

3.1. Use case application diagram

![Use case diagram](image)

Figure 3. Use case diagram.

Based on Figure 3, the system has two actors namely the user and admin. Each actor has their respective roles. Users can search for school information and news, while admin is to process data from systems such as adding, changing, and deleting data [11].

3.2. Architecture system

![System architecture](image)

Figure 4. System architecture.

Based on Figure 4, it is explained that this android application is an application that requires a combination with other systems such as a web service. The location of the user's position is known by using GPS which can capture longitude coordinates and latitude. Applications on the front end are applications that are used by users on Mobile devices that use the Android platform.

3.3. Android

Android is a Linux-based operating system for cellular phones such as Smartphone and tablet computers. Android also provides an open platform for developers to create their own applications that will be used for a variety of motion devices. Initially, Google Inc. bought Android Inc., a newcomer who makes software for cell phones. Later in the development of Android, the Open Handset Alliance was formed, a consortium of 34 hardware, software and telecommunications companies, including Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, and NVidia [12,13].

3.4. Implementation of Haversine formula

Next is the Pseudo code from Haversine formula:

Deklarasi (Kamus):

\[
RBumi = 6371.0
\]

longitude, latitude, haversine, jarak = int

Deskripsi:

Read RBumi
Read longitude
Read latitude
Radian longitude = degree to radian (longitude2 – longitude 1)
Radian latitude = degree to radian (latitude2 – latitude 1)
Haversine = 2 * \( \sin(\text{radian latitude}/2) \) *
\( \sin(\text{radian latitude}/2) \) + \( \cos(\text{degree to radian(latitude1)})*
\( \cos(\text{degree to radian(latitude1)})*\)
\( \sin(\text{degree longitude}/2)* \sin(\text{degree longitude}/2) \)
Jarak = \( R_{\text{Bumi}} \) * Haversine

Next is the implementation of Haversine Formula in the form of a PHP code to calculate the distance from the user's location to the destination school. The system needs to know longitude coordinate points, latitude points of user location and school location.

```php
function haversine($latitude1, $longitude1, $latitude2, $longitude2)
{
    $R = 6371.0;
    $dLon = deg2rad($longitude2 - $longitude1);
    $dLat = deg2rad($latitude2 - $latitude1);
    $sa = \sin($dLat/2) * \sin($dLat/2) +
        \cos(deg2rad($latitude1))*
        \cos(deg2rad($latitude1))*
        \sin($dLon/2) * \sin($dLon/2);
    $c = 2 * \sin(sqrt($sa));
    $d = $R * \$c;
    return round($d, 2);
}
```

3.5. Testing
Table 1 shows the results of calculations using Haversine Formula from ten samples which calculated the distance between the user's location and several schools.

| No | User location | School name | Application |
|----|---------------|-------------|-------------|
| 1  | -7.009528, 107.547778 | -6.9202, 107.6211 | 11.41 Km |
| 2  | -7.009528, 107.547778 | -6.8828, 107.6132 | 14.84 Km |
| 3  | -7.009528, 107.547778 | -6.9054, 107.6184 | 12.73 Km |
| 4  | -7.009528, 107.547778 | -6.9074, 107.5782 | 10.62 Km |
| 5  | -7.009528, 107.547778 | -6.9029, 107.6355 | 13.88 Km |
| 6  | -7.009528, 107.547778 | -6.9265, 107.6568 | 13.44 Km |
| 7  | -7.009528, 107.547778 | -6.8742, 107.6189 | 15.95 Km |
| 8  | -7.009528, 107.547778 | -6.9513, 107.592 | 6.88 Km |
| 9  | -7.009528, 107.547778 | -6.8972, 107.6158 | 13.44 Km |
| 10 | -7.009528, 107.547778 | -6.9149, 107.6339 | 12.66 Km |
4. Conclusion
Based on the discussion of research in this application development, it can be concluded that the School Location Information Application can provide some information and the location of the school that is closest to the users. This application can also calculate the distance from the user's location to the destination using the Formula Haversine Method. So that, the users who use this application will get the convenience to find the closest school of their choice.

Acknowledgment
We thank all parties, especially to the authors for their assistance in the form of techniques and methodologies, and also for them who give us good advises that make this paper greatly improved. We also want to show our gratitude to UIN Sunan Gunung Djati Bandung research institute, Department of Informatics at the faculty of science and technology and Department of Information System Universitas Kebangsaan, for their big assistance to make this research completed properly.

References
[1] Indonesia P R 2014 Undang-Undang Republik Indonesia Nomor 23 Tahun 2014 Tentang Pemerintahan Daerah 561–565
[2] Alam C N, Manaf K, Atmadja A R and Aurum D K 2016 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
[3] Utomo R G, Maylawati D S and Alam C N 2018 Implementasi Algoritma Cheapest Insertion Heuristic (CIH) dalam Penyelesaian Travelling Salesman Problem (TSP) J Online Inform. 3 1 61
[4] Javed M, Ahmad B, Jan M A, Abid M A and Shah M A 2012 RUP Certification via CRM Certification Process : Development of Software with Zero Defect Rate 48 1–10
[5] Anwar A 2014 A Review of RUP (Rational Unified Process) Ashraf Anwar Int J Softw Eng. 5 8
[6] Basyir M, Nasir M S and Mellyssa W 2018 Determination of Nearest Emergency Service Office using Haversine Formula Based on Android Platform Emit Int J Eng Technol. 5 2 270
[7] Khairina D M, Ramadhinata F W and Hatta H R 2017 Pencarian Lokasi Jalur Nugraha Ekakurir (JNE) Terdekat Menggunakan Haversine Formula (Studi Kasus Kota Samarinda) Pros SENIATI 3 1–5
[8] Ganesh L and Kumar D V 2015 Indoor Wireless Localization using Haversine Formula Int Adv Res J Sci Eng Technol. 2 7 59–63
[9] Hartanto S, Furqan M, Putera A, Siahaan U and Fitriani W 2017 Haversine Method in Looking for the Nearest Masjid Int J Recent Trends Eng Res. 3 8 187–195
[10] Anisyia A and Swara G Y 2017 Implementation of Haversine Formula and Best First Search Method in Searching of Tsunami Evacuation Route IOP Conf Ser Earth Environ Sci. 97 1
[11] Ibrahim N, Ibrahim R, Saringat M Z, Mansor D and Herawan T 2011 Consistency rules between UML use case and activity diagrams using logical approach Int J Softw Eng Its Appl. 5 3 119–134
[12] Holla S and Katti M M 2012 Android Based Mobile Application Int J Comput Trends Technol. 3 3 486–490
[13] Team GDT 2016 Android Developer Fundamentals Course