Implementation of Hill Climbing Algorithm on Tourist Attraction Android Based Application

1th Ramadhani, Shumaya Resty
Information of Technology
Politeknik Caltex Riau
Pekanbaru, Indonesia
shumaya@pcr.ac.id

2th Kreshna, Jan Alif
Information of Technology
Politeknik Caltex Riau
Pekanbaru, Indonesia
janalif@pcr.ac.id

3rd Rianda, Septian
Information of Techology
Politeknik Caltex Riau
Pekanbaru, Indonesia
septianrianda@alumni.pcr.ac.id

***Abstract*—West Sumatra is one province that has been a popular destination among tourists. West Sumatra is a strategic area that holds natural beauty that is no less great, and this is evenly distributed throughout the region. Data shows that the number of tourist visits continues to increase every year before the pandemic hits Indonesia. Travelers often use google maps to find routes to destinations. However, with little knowledge about the area, people only visit familiar tourist locations. Meanwhile, West Sumatra offers plenty of beautiful places. With those problems in mind, a mobile application is created to help those visitors searching the nearest tourist spots around them in real-time. Not only that, but the system also gives information about the list of places worth visiting in West Sumatra. The system implemented a hill-climbing algorithm to find the shortest path. Based on the evaluation, the application is well accepted for people who want to visit West Sumatra. According to survey conducted on this research, overall 94.6% of users agree that this app is useful. The app is handy and fast enough to display the desired result to the users.**

Keywords—hill climbing, android, LBS, West Sumatra, cyclomatic complexity

I. INTRODUCTION

Travelling is one of the activities that most people would love to do. Indonesia is a country that offers many wonderful places and views for domestic and international tourists to explore. In 2019 itself, there was around 1.3777.067 foreign tourist that visits Indonesia through international airport [1]. Although the number experienced a slight decline compared to 2018, which is around 2.03%, it is still appreciated. West Sumatra is one of the tourist destinations that are popular in Sumatra.

According to the Ministry of Tourism, the average number of international tourists visiting West Sumatra is around five thousand people each month [2]. Most of them may spend one to three days minimum in this province to enjoy the scenery [3]. This number shows that West Sumatra is attractive as it brings beautiful scenery, delicious food, and a friendly community for people worldwide.

Lack of tourism information is one of the reasons why the visitor number in West Sumatra remains low. The hidden gem spots are unreachable and remain unknown, while that could have been a potential attraction to many tourists. Aside from that, many travelers wasted their time finding the route to their destination and ended up missing the spots on the way. They visit only the most popular places without looking at other locations, especially in West Sumatra. People use Google Maps to find locations, yet it displays only specific information and route. In contrast, the information of the spots nearby is needed as well to travelers.

Certainly, by this fact, the need for a system that describes tourist spots is necessary. The system's goal is to find the nearest tourist attractions from the user position as well as the route to the location. Additionally, the system could provide a piece of clear information about the tourist objects. Thus, users can decide which place they would like to visit. Furthermore, the system could be handy and simple; thus, users feel comfortable using it while traveling. By having this system, travelers could maximize their trip to West Sumatra.

This application is designed using the Simple Hill Climbing method and using the Global Positioning System (GPS) function on the smartphone to detect user location. The Simple Hill Climbing method is relatively simple compared to other search optimization methods. However, the execution speed of this method could be a point to be considered, since the algorithm is implemented in mobile phones.

II. LITERATURE REVIEWS

A. Previous Research Studies

The Hill climbing search algorithm is commonly used to find the shortest path to the destination. There have been numerous studies related to this algorithm. One of the studies uses this method to calculate the distance between attractions. Not only that, but researchers also used the method to optimize the route to the 5 specific popular locations in every province of Indonesia. The study shows that the accuracy of determination from the research is 93.3% [4].

Another implementation that uses this method is to determine the shortest path for kerosene distribution. The system is an android based application for the agents and customers in Ambon city. Customers can see where the

nearest agents are around them. Meanwhile, the agents get the optimal route to distribute the products around the city [5].

Interestingly, some studies also show that hill-climbing could be implemented in Games. The research used simple hill-climbing in crossword puzzles to get a wording pattern in empty boxes. Hence, the more wording combination created to fill the boxes, the solution could be optimal. This algorithm is implemented automatically to create computer games [6].

B. Hill Climbing Search Algorithm

One of mathematical optimization method is called hill climbing for local search [7]. Hill climbing is one of the heuristics techniques to find rapid solutions compared to classic methods. Using the heuristic technique, hill climbing could reduce the amount of time needed on iteration processes to solve problems. However, it may not produce the best solution.

In heuristic, hill climbing consists of two types, simple hill-climbing, and steepest-ascent hill climbing. The algorithm that is used in this study is the simple hill climbing and implemented to find the nearest tourist location in West Sumatra. This method searches for better solutions in its surrounding by evaluating the current state. When the goal state finds that the heuristic value is less than the initial, the iteration process stops. The process may keep going when there is no better result, and there are operators left. Here is the pseudocode of this algorithm.

```
INITIALIZE initial state, current state, goal state
IF initial state != goal state
      WHILE neighbors != 0
        current state = neighbors
          IF current state < initial state
            THEN initial state = current state
          END
      END WHILE
END

IF initial state = goal state
RETURN "Success"
END
```

Hill climbing algorithm is straightforward to understand. Moreover, the execution process is fast and easily adapt to the case study [8]. Therefore, the consideration of implementing this algorithm in a mobile application is acceptable to get a fast solution.

C. Android

Android is an operating system that generally runs on mobile devices. Android was originally developed by a company called Android Inc. which Google later purchased in 2005. Devices that embed the Android operating system were first launched in 2007, around 13 years ago [9]. This Android-based operating system is currently prevalent and has become a popular choice for users of smartphones, tablet PCs, and Smart TVs. The development of Android opens up huge market opportunities for mobile app developers to compete in developing various types of mobile applications that run on this operating system.

D. Location Based Services

Location-Based Services (LBS) is the concept of an application that uses a combination of geographical location and services [10]. This service could be accessed in almost all mobile devices through network in order to get spatial coordinates.

Even though this service is popular and provides benefits to applications, collecting user locations as information is quite sensitive. Systems should not store the private data arbitrarily or without a permit. Hence the usage of these services should be restricted or need proper security improvement [11].

This implementation uses LBS from mobile devices to get the user's location by taking advantage of GPS. Application may ask user permission to enable GPS. When declined, then users cannot use the searching feature. Nevertheless, this application will not save user location.

III. RESEARCH METHODS

A. Implementation of Hill Climbing

This application compares the distance obtained from the user's location to the nearest tourist location from the user. This application will ask for tourist data that has been inputted into firebase. After the data is obtained, the data is stored into an Array List which later will be used as a comparison of the distance between tours hence the process will be faster rather than requesting from database.

After the data has been taken, the next stage is finding the nearest tourist attraction using the Simple Hill Climbing algorithm. At this stage, the user's location is the initial node, and the data is collected in the destination node. Then a comparison of the initial node with the existing tourist data nodes is carried out. After getting the results using the Simple Hill-Climbing algorithm, the route will be created using Google Direction to find the shortest route. The Simple Hill-Climbing algorithm in this application is written using the Java programming language. By this way, the algorithm could work efficiently before displaying the result as markers on screen.

B. Data Collection Method

The tourism spot data is obtained from the official web of West Sumatra is https://sumbar. travel/ and https://dispar.sumbarprov.go.id/. Hence the data could be more valid as it is registered and publicly known by people. There are 37 tourist attractions in West Sumatra spread from one district to another. Some of the most popular objects are jam gadang, lembah anai, lembah arau, air manis beach and more. The data obtained will be used to reference the list of tourist locations in the android application.

IV. IMPLEMENTATION

This chapter explains how the system is implemented. It contains the system architecture, flowchart, requirements, and the result of a mobile-based application.

A. Architecture System

Fig. 1 is the architecture of the mobile application. The app gets the user's location using GPS and saves it into Firebase. The next step is implementing the simple hill-climbing

algorithm to calculate and find the shortest path between objects. Lastly, the output is mapped using google maps on android.

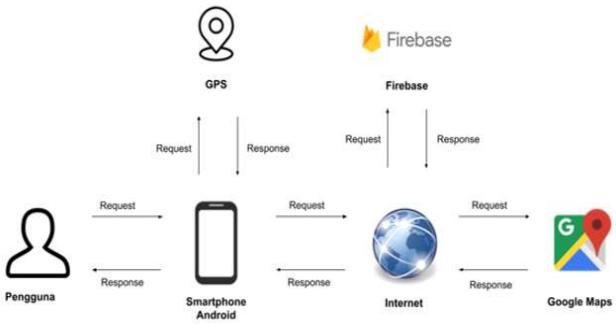

Fig. 1. System Architecture

This app needs an internet connection to access objects' locations through google API and Firebase. There is no local storage used in this mobile app. The application used google API to retrieve additional information about the objects.

B. *Flowchart System*

Fig. 2 below shows the flow of the android based mobile application. Once the app is on, it asks the user to activate the GPS. This process is needed to get the current user position. Application retrieves all attractions in West Sumatra and information and display it on the map on the application.

The system will implement the algorithm after all location data is obtained from Firebase and API. The algorithm calculates the distance between one point to another and finds the shortest path. The location range is set to five kilometers to prevent such a far location from arising on the app. When the result shows no data found, it displays a notification and asks the user to set another starting position.

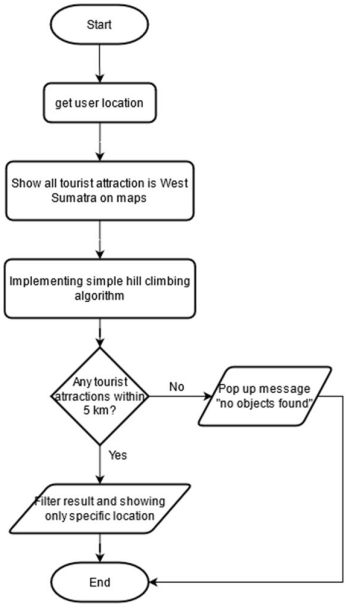

Fig. 2. Flowchart of the system

C. *Usecase*

Usecase diagram as it shows in Fig. 3 is made according to the user's requirements. The stakeholder of this mobile app is the tourists or visitors who want to visit West Sumatra.

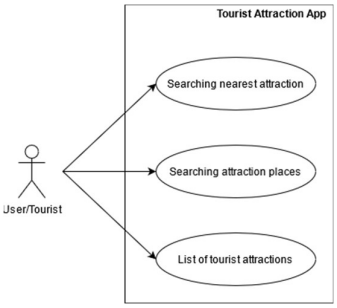

Fig. 3. Usecase application

The focus of the mobile app is to automatically find the nearest place within the designated range to the user position. Besides that, users can also find specific places by inputting the location into the system. As an additional feature for new visitors, the app also provides a list of tourist attractions in West Sumatra.

D. *Interface*

Pictures below display the interface of android based application. Fig. 4 is the page to find the tourism spot nearby.

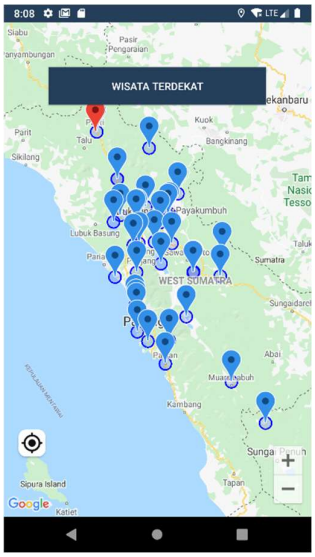

Fig. 4. Finding nearest location page

GPS detects user position automatically once it is activated. When the user presses the button, the app displays the result within the area, as shown below. The red marker is the user location, and the blue is the spot (Fig. 5).

Fig. 5. The nearest objects within range

This page is used to search tourist locations that have been stored in Firebase as a dropdown list. User selects location and map shows the way to specific input by giving route. Application displays the coordinates as it shown in Fig. 6.

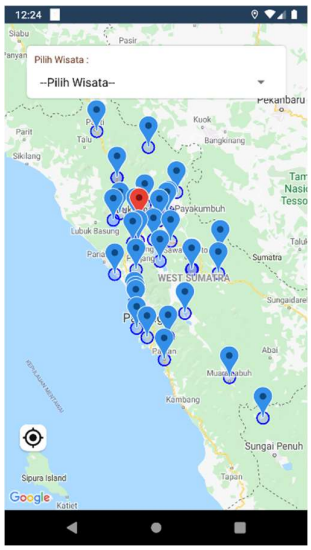

Fig. 6. Searching location page

This page in Fig. 7 displays all tourism spots as a list. Users can click on an item and get all information and description that the user might need.

Fig. 7. List of tourist attractions

If the user clicks on each item, the app displays details and information about the objects (Fig. 8). The user could read the story behind the tourist spot and take a look at the gallery it provides.

Fig. 8. Details page

V. EVALUATION

The researcher conducted several evaluations for this implementation, such as counting time execution, checking the algorithm's complexity, and giving questionnaires to several users.

A. Computational Time

Testing on-time execution is mandatory since the application could load numerous object locations in a short amount of time. The evaluation of this testing is to ensure that a simple hill-climbing algorithm works in fair time to the phone user. This testing used all data locations stored in Firebase. However, the tourism spot in West Sumatra that the writer found is limited, which is less than 30 locations.

Therefore, to test loading time for the app, dummy data is added into storage from a thousand up to a million points. Here are the details of the average loading time:

TABLE I. TABLE 1. LOADING TIME

No. of Iteration	*Total Data Coordinates*	*Avg. Loading Time (s)*
1.	1.000	0.003
2.	10.000	0.004
3.	100.000	0.015
4.	1.000.000	0.111

The data above shows that it takes a longer time to execute expandable data from storage. It needs time to retrieve all data from the cloud and then process it using a hill-climbing algorithm before inserting the result as a marker to the map. Nevertheless, the result of this evaluation is still tolerable, especially for mobile apps. The average time remains under 0.5 second for around a million-location data, hence could be considered hence it has less impact on user experience.

B. Cyclomatic Complex (CC)

This Cyclomatic Complexity test aims to assess whether the code of the Simple Hill-Climbing algorithm has been written well. The metrics could help identify the part of the code that can potentially lead to error and possibly lead to ineffectiveness of algorithm implementation [12].

The test using embedded Plugins from Android Studio, namely Metrics Reloaded [13]. This plugin calculates the value of Cyclomatic Complexity for all projects in Android. The v(G) is the cyclometric complexity, ev(G) is the essential cyclometric complexity, and iv(G) is the Module Design Complexity Metric.

TABLE II. TABLE 2. RESULT OF METRICS RELOADED ON SIMPLE HILL CLIMBING

method	*ev(G)*	*iv(G)*	*v(G)*
LokasiTerdekat.simpleHillClimbing()	1	4	4

Based on the results above, v(G) = 4 is obtained for the Simple Hill Climbing algorithm. The result shows that the code implementation has low complexity [14]. However, this metrics (CC) has disadvantages that cannot measure the satisfaction of code understandability [15].

C. Questionnaire Evaluation

The researcher surveyed this mobile app to gather knowledge about user response. This survey was conducted on 30 respondents by using online media, Google Forms. The age of respondents is above 17 years old and loves to travel, especially to West Sumatra. More than half of respondents (73.3%) agree that this mobile application is helpful, specifically for tourists who never visit West Sumatra before.

Application features are also expected to meet user requirements. The survey shows that 50% of respondents agree that the features provided were sufficient, and the rest expect more additional focus besides tourism spots. For example, adding culinary and shopping locations.

92.6% of respondents recognize that the app is easy to understand. They were able to operate and navigate through the app without any problems. Moreover, the information provided in the mobile app is clear and straightforward. Therefore, the users do not need assistance when using the application.

VI. CONCLUSION AND FUTURE WORK

According to the implementation and evaluation above, a simple hill algorithm is a one of the proper optimization algorithms for finding the nearest tourist locations. The system is a mobile-based application; hence requires the enhancement of the searching speed before displaying the result to the user. According to the evaluation on computation time, the algorithm only needs less than 0.2 seconds to get the result from a huge amount of data and put the marker on the screen. Therefore, the user may get the information needed immediately.

The algorithm is also implemented with a good coding style showed by evaluating the outcome of the complexity metrics. However, there is still uncertainty on the code understandability since the evaluation did not measure the satisfaction of the code readability.

This algorithm could produce better results for future reference to find and recommend locations when combined with artificial intelligence methods. Therefore, the travelers could have recommendations of which place to visit. Additionally, the application could also cover the tourist locations and accommodation places, culinary, and added real-time traffic.

REFERENCES

[1] K. P. dan E. Kreatif, "Data Kunjungan Wisatawan Mancanegara Bulanan Tahun 2019," 2019. https://kemenparekraf.go.id/post/data-kunjungan-wisatawan-mancanegara-bulanan-tahun-2019 (accessed Jul. 12, 2021).

[2] S. Provinsi, "Perkembangan Pariwisata dan Transportasi Sumatera Barat Agustus 2019," 2019. https://sumbarprov.go.id/home/news/17484-perkembangan-pariwisata-dan-transportasi-sumatera-barat-agustus-2019 (accessed Jul. 12, 2021).

[3] P. Ekonomi and S. Barat, "Potensi Daya Tarik Obyek Pariwisata Dalam Pembangunan Ekonomi Sumatera Barat," *Economica*, vol. 1, no. 1, pp. 1–15, 2012, doi: 10.22202/economica.2012.v1.i1.101.

[4] D. Fitriati and N. M. Nessrayasa, "Implementasi Algoritma Hill Climbing Pada Penentuan Jarak Terpendek Kota Wisata Di Indonesia," *J. Ris. Inform.*, vol. 1, no. 3, pp. 127–132, Jul. 2019, doi: 10.34288/JRI.V1I3.35.

[5] R. G. Elly, "Penerapan Metode Simple Hill Climbing Untuk Penentuan Jalur Terpendek Penyaluran Minyak Tanah Pada Agen Berbasis Android - Akakom Repository," *Digital Library STMIKAKAKOM*, 2017. https://eprints.akakom.ac.id/3741/ (accessed Jul. 12, 2021).

[6] P. M. Hasugian, "Implementasi Algoritma Simple Hill Climbing Pada Aplikasi Game Tts (Teka – Teki Silang)," *J. Mantik Penusa*, vol. 19, no. 1, Jun. 2016, Accessed: Jul. 12, 2021. [Online]. Available: http://www.e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/232.

[7] A. Schaerf and A. Meisels, "Solving Employee Timetabling Problems by Generalized Local Search," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1792 LNAI, pp. 380–389, 1999, doi: 10.1007/3-540-46238-4_33.

[8] M. Shehab, A. T. Khader, and M. Laouchedi, "A Hybrid Method Based on Cuckoo Search Algorithm for Global Optimization Problems," *J. Inf. Commun. Technol.*, vol. 17, no. 3, Jul. 2018, doi: 10.32890/JICT2018.17.3.4.

[9] "What is Android | Android." https://www.android.com/what-is-android/ (accessed Jun. 10, 2020).

[10] A. V. Jochen Schiller, "Location-Based Services - Google Buku," in *Location-Based Services*, San Francisco: Morgan Kaufmann

Publishers, 2004.

[11] N. Fei, Y. Zhuang, J. Gu, J. Cao, and L. Yang, "Privacy-preserving relative location based services for mobile users," *China Commun.*, vol. 12, no. 5, pp. 152–161, May 2015, doi: 10.1109/CC.2015.7112037.

[12] L. Dr. Rosenberg, T. Hammer, and J. Shaw, "Software metrics and reliability," *Proc. 9th Int. Symp. Softw. Reliab. Eng.*, pp. 1–8, 1998.

[13] B. Wang, "An Android Studio Plugin for Calculating and Measuring Code Complexity Metrics in Android Applications," Towson University, Maryland, 2016.

[14] "PMD - Rule Set: Code Size Rules." https://pmd.sourceforge.io/pmd-4.3.0/rules/codesize.html (accessed Jul. 14, 2021).

[15] G. A. Campbell, "Cognitive Complexity," Switzerland, 2016. doi: 10.21075/kacsn.2009.11.3.237.