

Mosque Search with KD-Tree Nearest Neighbor Case Studies and Field District of Johor

*Dedi Setiawan¹

Department Computer Engineering,
STMIK Triguna Dharma Medan,
Jln. A.H. Nasution No 73 F,

*Iskandar Zulkarnain²

Department Computer Engineering,
STMIK Triguna Dharma Medan,
Jln. A.H. Nasution No 73 F,

*Hendryan Winata³

Department Computer Engineering,
STMIK Triguna Dharma Medan,
Jln. A.H. Nasution No 73 F,

*Ismawardi⁴

Department Computer Engineering,
STMIK Triguna Dharma Medan,
Jln. A.H. Nasution No 73 F,

Abstract-Mosque to be one right choice of worship. Mosque closest to these areas are often chosen first before choosing the other mosques. Therefore made a decision support system that can perform a calculation to find the nearest mosque. This system menggunakan metode KD-Tree that peranya in the system is to create a tree that will be used as groove comparisons and calculations, and the Nearest Neighbor method for calculating the shortest distance between the user and the mosque. With this system is expected to help the user or so travelers can easily get to the nearest mosque and as expected.

Keywords: Mosque, KD-Tree, Nearest Neighbor

1. PRELIMINARY

1.1. Background

Nearest Neighbor method is a technique of classification based on the proximity of the object, by comparing the distance of each objek. Approach. used on Nearest Neighbor itself a classification approach that searches all training data are relatively similar to the test data. This classification technique called lazy learning because this technique does not build the classification model in advance, such as a decision tree (decision tree) based classification rule (rule-based), and so on. K-Dimensional Methods or KD-Tree Tree constitute method for file group. Known for k-dimensional tree-dtree k is a binary tree node her a k-dimensional point. K can be worth 2 or more. K-way d-tree segment data is the same as the usual binary tree. A region is divided into two, then each of the two regions was subdivided into two regions, thereby on up can not be subdivided.

Mosque is a house of worship of Islam or Muslims. Mosque means place of prostration, and another title for mosques in Indonesia is a mosque, broken or surau. These terms are not intended for use mosques for Friday prayers, and generally small. Besides being used as a place of worship, the mosque was also a center of Muslim

community life. The activities of festivities, discussions, religious studies, lectures and study the Koran is often carried out in mosques. Even in the history of Islam, the mosque helped play a part in social activities to the military. For the general public or tourists who had just arrived and have not been peddling foot in a city order to get what they want? Yes, it can need a system that can assist in determining the nearest mosque. In this study will use a combination of methods KD-Tree and Nearest Neighbor in the nearest mosque quest completion.

1.2. Formulation of the problem

Based on the background above, the formulation of the problem gained is how to design and build a system to determine the nearest mosque by applying KD-Tree and Nearest Neighbor who can assist in the search by the user or tourists.

1.3. Research purposes

The purpose of this study is:

- 1) Applying the KD-Tree method and Nearest Neighbor in decision support systems.
- 2) Determine the nearest mosque using the KD-Tree and Nearest Neighbor.

1.4. Benefits of research

The benefits of this research is to know and understand how to locate the nearest mosque by applying KD-Tree and Nearest Neighbor system.

1.5 Scope of problem

Limitation of problems in this research that the data used in the determination of the nearest mosque by using the coordinates and information about the mosque is based on data obtained from direct conservation where the mosque is located.

2. THE RESEARCH METHODOLOGY

2.1. Problem Identification and Analysis

Problem identification is done by direct observation with the general public (tourists).

2.2. Systems Development Method

This study was developed by applying the waterfall model. This model proposes an approach to software development that is systematic and sequential.

1) File Collection

File collection was conducted to describe the nearest mosque retrieval system, used several methods of file collection among others:

a) Method Interview (Interview)

Is a direct method of data collection by interviewing the parties relating mosque of the file of the mosque.

b) Method Library (Library Research)

Library method is done by studying and collecting some literature from the Internet or media-related books in the study process.

c) Methods of Observation

A method of collecting data directly from the field to determine the coordinate points of the mosque, if the coordinates of the mosque have not been obtained in the interview method.

2) Analysis

Analysis of the stage to specify the KD-Tree method and Nearest Neighbor will be used. This includes the process of initial formation of the tree is done by the KD-Tree to calculations made by Nearest Neighbor resulting distance the nearest mosque.

3) Testing

Testing is a stage to determine whether the programs are made in conformity with the specifications of the design phase.

3. RESULTS AND DISCUSSION

3.1. System Overview

The system made a decision support system that uses the KD-Tree method and Nearest Neighbor mobile based android that can help users or tourists in search of the nearest mosque. The system is built using the Java programming language and uses MongoDB and SQLite databases. The system will then generate information based on the distance between the nearest mosque application users point coordinates with the coordinates of points mosque.

3.2. KD-Tree Method and Nearest Neighbor

K-d-tree segment data is the same as the usual binary tree. A region is divided into two, then each of the two regions was subdivided into two regions, demikian on up can not be subdivided.

KD-Tree reconstruction process using the recursive method parameters on each iteration are arrays of coordinates mosques and depth. Depth value is used to determine the value axis. The initial value for the arrays are all the coordinates of the mosque and the depth value is 0 for each iteration process is as follows:

1) Calculating the value axis with the formula: $axis = depth \text{ mod } 2$

2) Perform sorting at the mosque coordinate array based on the value axis. If the axis = 0 then the sorting is done based on longitude, and if axis = 1 based latitude.

3) Finding the coordinates of the median, by the way, (i) $index_median = amount_coordinate \text{ div } 2$ and (ii) $coordinate_median = coordinates [index_median]$.

4) Determine the node: $node = coordinates [index_median]$

5) Determine the left node and right node using a new iteration. The next iteration using this parameter as follows: (i) an array of coordinates = sub array coordinate = $coordinates [0] [index_median - 1]$ as well as the depth = $depth + 1$, and (ii) the array coordinate = sub array coordinate = $coordinates [index_median + 1] [amount_coordinate]$

Coordinates mosque used in the calculation of the ratio manual No 10 which serve as an example:

1. (-3.32381, 114.60398)
2. (-3.32399, 114.5914)
3. (-3.33472, 114.62178)
4. (-3.32377, 114.58736)
5. (-3.33675, 114.61646)
6. (-3.32646, 114.59583)
7. (-3.33977, 114.61875)
8. (-3.3239, 114.60139)
9. (-3.32866, 114.60192)
10. (-3.32769, 114.5897)

Iteration 1

File = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Iteration 1.1

File = 4, 10, 2, 6

n = 4

depth = 1

axis = 1 mod 2 = 1

sort by latitude

sorted file : 4, 2, 6, 10

median = 4 div 2 = 2

Iteration 1.2

File = 9, 1, 5, 7, 3

n = 5

depth = 1

axis = 1 mod 2 = 1

sort by latitude

sorted file : 1, 9, 5, 7, 3

median = 5 div 2 = 2

Iteration 1.1.1

File = 4

n = 1

depth = 2

because n = 1 then node = file[4]

Iterasi 1.1.2

File = 6, 10

n = 2

depth = 2

axis = 2 mod 2 = 0

sort by longitude

median = 2 div 2 = 1

node = file[10]

Iteration 1.2.1

File = 1

n = 1

n = 10

depth = 0

axis = depth mod 2 = 0 mod 2 = 0

sort by longitude

sorted file : 4, 10, 2, 6, 8, 9, 1, 5, 7, 3

median = n div 2 = 10 div 2 = 5

node = file [s] = (-3.3239, 114.60139)

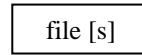


Figure 1. Tree Iteration 1

node = file[2]

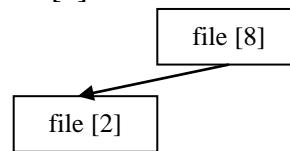


Figure 2. Tree Iteration 1.1

node = file[9]

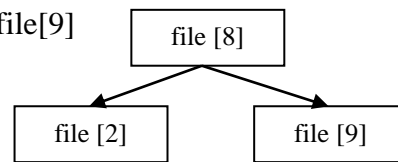


Figure 3. Tree Iteration 1.2

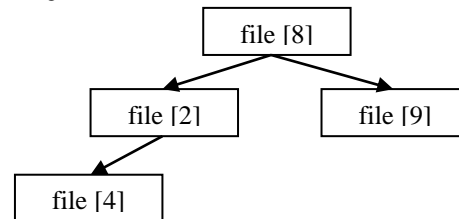


Figure 4. Tree Iteration 1.1.1

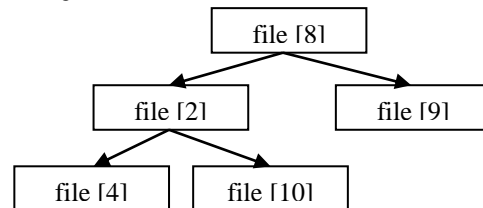


Figure 5. Tree Iteration 1.1.2

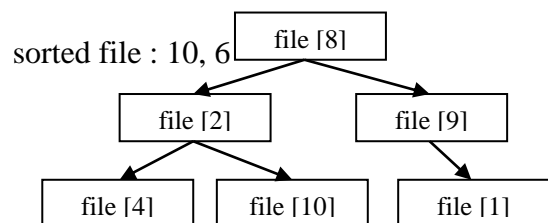


Figure 6. Tree Iteration 1.2.1

depth = 2
 because $n = 1$, then node = file[1]

Iteration 1.2.2

File = 5, 7, 3
 $n = 3$
 depth = 2
 $axis = 2 \bmod 2 = 0$
 sort by longitude
 sorted file : 5, 7, 3
 $median = 3 \div 2 = 1$
 node = file[5]

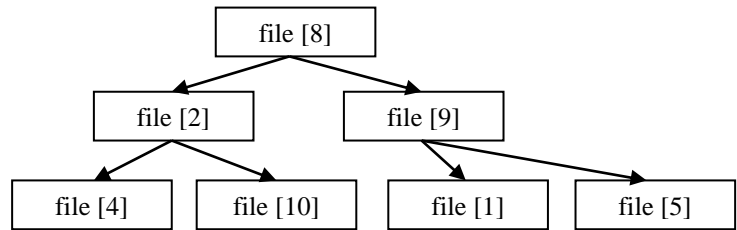


Figure 7. Tree Iteration 1.2.2

Iteration 1.1.2.1

File = 6
 $n = 1$
 depth = 3
 because $n = 1$, then node : file [6]

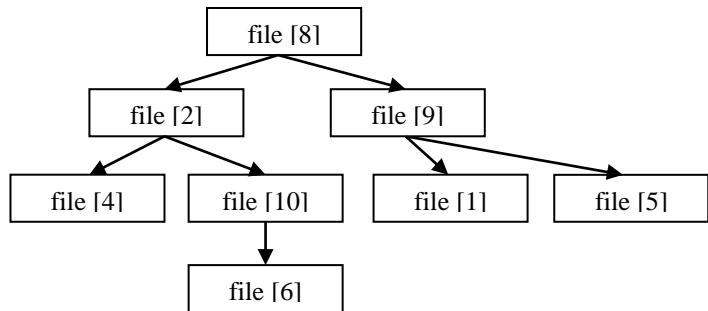


Figure 8. Tree Iteration 1.1.2.1

Iteration 1.2.2.1

File = 7, 3
 $n = 2$
 depth = 3
 $axis = 3 \bmod 2 = 1$
 sort by latitude

sorted file : 7, 3
 $median = 2 \div 2 = 1$
 node = file[7]

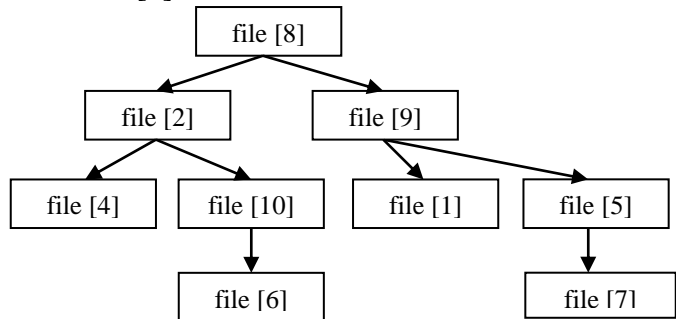


Figure 9. Tree Iteration 1.2.2.1

Iteration 1.2.2.1.1

File = 3
 $n = 1$
 depth = 4
 because $n = 1$, then
 node = file[3]

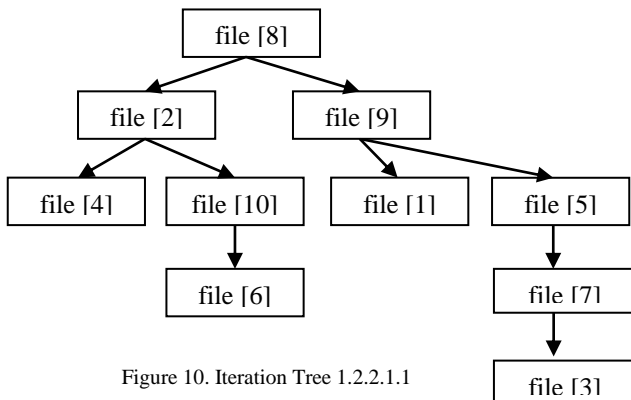


Figure 10. Iteration Tree 1.2.2.1.1

Nearest Neighbor is a classification technique based on proximity of the object. Proximity herein defined by the size of the distance, for example Euclidean. Euclidean distance between two points, eg Titik1 = (x1, y1) and point2 = (x2, y2) are: $Dist (point 1, point 2) = \sqrt{(x1-x2)^2 + (y1-y2)^2}$

The working principle Nearest Neighbor entered on the application is seeking the shortest distance between the file to be searched is the nearest mosque to the application user

Table 1. Calculation Manual Nearest Neighbor

X1	-3.54439
Y1	114.84220

No	Name Mosque	X2	Y2	Distance
1	Mosque A	-3.32769	114.58970	0.332740149
2	Mosque B	-3.32058	114.58225	0.343024509
3	Mosque C	-3.32377	114.58736	0.33707194
4	Mosque D	-3.32417	114.59019	0.334674124
5	Mosque E	-3.32399	114.59140	0.333882736
6	Mosque F	-3.32640	114.59051	0.33296902
7	Mosque G	-3.32724	114.59662	0.327817827
8	Mosque H	-3.32859	114.59687	0.326737375
9	Mosque I	-3.32866	114.60192	0.322916048
10	Mosque J	-3.32646	114.59583	0.328926334
11	Mosque K	-3.32390	114.60139	0.326506016
12	Mosque L	-3.32381	114.60398	0.32466165
13	Mosque M	-3.32982	114.61583	0.311904665
14	Mosque N	-3.33675	114.61646	0.30671444
15	Mosque O	-3.33977	114.61875	0.302985256
16	Mosque P	-3.35119	114.62845	0.288125104
17	Mosque Q	-3.35421	114.62962	0.285235779
18	Mosque R	-3.34720	114.62178	0.295752703
19	Mosque S	-3.30733	114.58822	0.347425146
20	Mosque T	-3.29292	114.58958	0.356447796
21	Mosque U	-3.44395	114.83307	0.100856097
22	Mosque V	-3.44299	114.82483	0.10287897
23	Mosque W	-3.44280	114.82181	0.103617984
24	Mosque X	-3.55302	114.84571	0.1009314638
distance Smallest				0.009314638

The above table is a table of the data point coordinates and their mosques user coordinates. Of those coordinates can be calculated and the smallest will result intended as the nearest point. Suppose mosque position (latitude (x2), longitude (y2)) and the user's position (latitude (x1) and longitude (y1)), then the data one by one calculated using the Nearest Neighbor. After all the calculations are obtained smallest result (0.009314638) mosque Z, the result intended to be the closest point from the user.

Similarly, the results displayed on the search application nearest mosque. Applications run on the same user position at coordinates (latitude (-3.54439) and longitude (114.84220) is shown with a blue dot on the map google. Once the process is done then obtained the nearest mosque is a mosque Z as shown in Figure 11 below.

From the above results can be taken that the Nearest Neighbor on the application is running as it should, and goes well in the calculation of the coordinates of the nearest search.

In this application KD-Tree has a role in helping to speed up the calculations done by Nearest Neighbor is to make a tree (tree) of all the coordinate points of the existing, real tree (tree) it is a path that connects between the point -point that will be used as a comparison for Nearest

Neighbor groove. Nearest Neighbor do not need to compare all of the coordinates of the existing, but only compares accordance with the path that has the KD-Tree created. The lack perluan compare all the coordinate data Nearest Neighbor who made more efficient with the help of the KD-Tree, the comparison does not use it.

4. CONCLUSION

The conclusion of this study are:

1. Being able to implement the method of the KD-Tree and Nearest Neighbor in this system.
2. Method KD-Tree and Nearest Neighbor in this system has been able to determine the nearest mosque from the user point to the nearest mosque to the right point.

5. BIBLIOGRAPHY

- [1] Munir, R. 2010. "Discrete Mathematics Fourth Edition" .Bandung: Publisher Information
- [2] Han, Jiawei, and Micheline Kamber, "Data Mining: Concepts and Techniques", Elsevier Inc., United States of America, 2006.
- [3] Rudi, Rudolf Hermanto, "Utilization Tree for Indexing Spatial Data Base", Department of Information Technology, Bandung, 2012.