*Journal of Electrical Engineering and Computer Sciences* ISSN: 2528-0260
*Vol. 5, Issue 2, December 2020* p.845-864

# GEOGRAPHIC INFORMATION SYSTEM MAPPING OF HOUSING LOCATIONS USING WEB-BASED BREADTH FIRST SEARCH ALGORITHM

## (Case Study: Sidoarjo Regency)

[1] ARIF ARIZAL, [2]M. MAHAPUTRA HIDAYAT, [3]DWI BUDI MARWANTO

[1]Informatics Engineering Study Program, Faculty of Engineering

Bhayangkara University – Surabaya

e-mail: [1]arifarizal@ubhara.ac.id, [2]mahaputra@ubhara.ac.id, [3]dwibudim@gmail.com

ABSTRACT

*Sidoarjo Regency as one of the cities with an increasingly dense population and the needs of people who want to find information about housing quickly, makes a Geographical Information System indispensable, especially in terms of finding housing locations. Therefore, the authors create a web-based housing geographic information system in Sidoarjo Regency using Breadth First Search to design the system interface and logic, MySQL for database system design and Bing Api for mapping. Information presented in the form of housing names, addresses, photos of housing, pictures of house types, house plans, availability and prices of each type of house. With the GIS (Geographical Information System), it is hoped that the public can more easily and quickly get housing information in Sidoarjo Regency. The results of this study show the details of the housing location from the user's location according to the radius to be selected and know the approximate distance of travel time to the residential location.*

**Keywords:** *Web based, Bing API, Housing, GIS*

## I. INTRODUCTION

### 1.1 Background Research

Sidoarjo Regency is a regency located in East Java Province. The geographical location of Sidoarjo Regency is very close to and directly adjacent to the metropolitan city as well as the capital of East Java Province, namely the city of Surabaya. As a buffer city of the city of Surabaya, Sidoarjo Regency makes a very good and strategic investment land in the East Java Region.

The large number of property developers in Sidoarjo Regency makes housing buyers have to select the many criteria they want. The process of buying a house is very important for prospective buyers who make the wrong decision. To overcome helping prospective buyers choose the desired criteria, an information system that is capable of integrating and processing non-spatial and spatial data is needed, especially the mapping of housing with public facilities, education buildings and health buildings. This is what causes prospective buyers to have limited information about buying a home. A media that can deliver information related to the mapping of housing development areas is needed, so that problems such as housing searches are dynamic. The use of Geographic Information System (GIS) technology is very helpful in mapping / determining housing points in Sidoarjo Regency.

## II. BASIC THEORY

### 2.1. Definition of Geographical Information Systems

Geographic Information System or better known as Geographical Information System, is an integrated information system specifically used to manage various data which has spatial information in spatial form, where this Geographic Information System technology can be used for scientific investigations, resource management. development planning, cartography, and even data are also used to plan routes. In practical terms, we can say that a Geographical Information System is a computerized system that has the ability to build, manage, analyze, store, and

display a Geographical Information System in the form of a mapping where users who build data and operate it are also part of the system..

**2.2. Breadth-First Search (BFS) Method**

The Breadth-First Search (BFS) algorithm or also known as the wide search algorithm is an algorithm that performs a wide search that visits a node pre-order, namely visiting a node then visiting all the nodes that are adjacent to that node first. Furthermore, the nodes that have not been visited and are adjacent to the nodes that have been visited, and so on. If the graph is a rooted tree, then all vertices at level d are visited before the vertices at level d + 1.

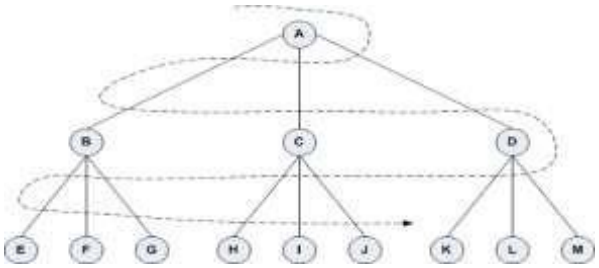



Figure 2.1. *Example of Graph in Dijkstra's Algorithm*

In the BFS algorithm, visited child nodes are stored in a queue. This queue is used to refer to neighboring nodes which will be visited later in the queuing order. To clarify how the BFS algorithm works and the queues it uses, here are the steps for the BFS algorithm:

Put the end (root) node into the queue.

b. Take a node from the start of the queue, then check if it is a solution.

c. If the node is a solution, the search is completed and results are returned.

d. If node is not a solution, enter all neighboring nodes (child nodes) into the queue.

e. If the queue is empty and every node has been checked, the search is complete and returns no results.

f. Repeat the search from step two.

An example is shown below:

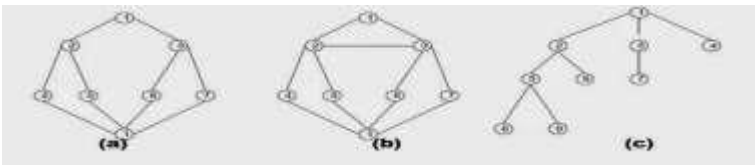


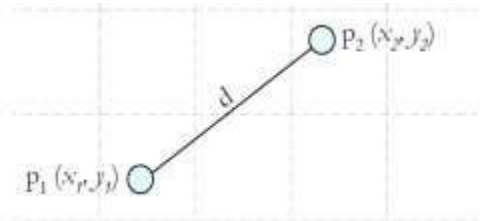

Figure 2.2 *Example of BFS*

To do the searching process on all nodes that are at the same level or hierarchy first before continuing the searching process on the nodes at the next level. The sequence of the BFS searching process is shown in the example of Figure 3.3. Then for the sequence of solutions Figure (a) BFS (1): 1, 2, 3, 4, 5, 6, 7, 1. Figure (b) BFS (1): 1, 2, 3, 4, 5, 6, 7, 1 Picture (c) BFS (1): 1, 2, 3, 4, 5, 6, 7, 8, 9.

**2.3. Euclidean Distance Method**

Euclidean Distance is the calculation of the distance from 2 points in Euclidean Space. Euclidean Space was introduced by Euclid, a mathematician from Greece around 300 B.C.E. to study the relationship between angle and distance. This euclidean is related to Pythagoras' Theorem and is usually applied to 1, 2 and 3 dimensions. But also simple when applied to a higher dimension. Euclidean Distance is a heuristic function obtained based on direct distance

free from obstacles such as to get the value of the length of the diagonal line in the triangle. But before getting the result the two points must be represented in 2-dimensional coordinates (x, y). Two points p1 = (x1, y1) and p2 = (x2, y2) become the following equation.




Euclidean Distance (d) = ($\sqrt{(\boldsymbol{x2} - \boldsymbol{x1})2 + (\boldsymbol{y2} - \boldsymbol{y1})2}$)

Euclidean formula

So that from the formula above we can implement it.

$$\text{Distance} = \sqrt{(\boldsymbol{Lat2} - \boldsymbol{Lat1})^2 + (\boldsymbol{Long2} - \boldsymbol{Long1})^2}$$

## III. ANALYSIS AND SYSTEM DESIGN

### 3.1. Flowchart System

A system flowchart is a graphical depiction of the steps and sequence of procedures of a program. Flowcharts help analyzers and programmers to solve problems into smaller segments and help in analyzing other alternatives in program operation. The following is an overview of the system flowchart that will be made**.**

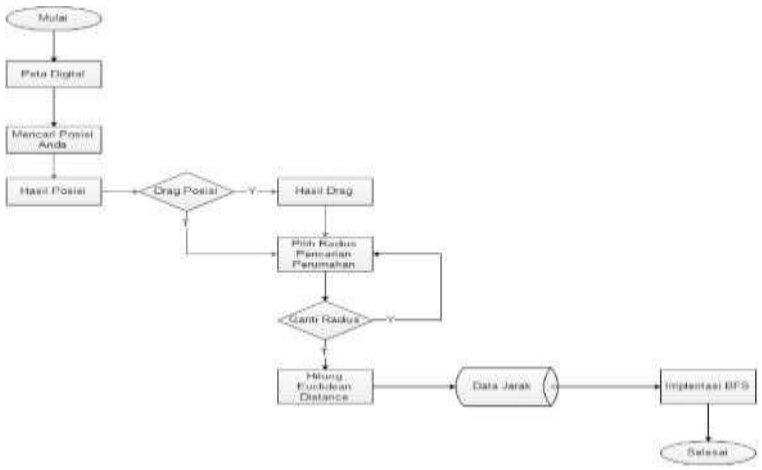



Figure 3.1. *System flowchart*

### 3.2 Entity Relationship Diagram (ERD)

Entity Relationship Diagram or ERD is a diagram that describes the arrangement of tables and their attributes and determines the relationships between tables. ERD also explains the relationship between attributes and tables, where attributes have a function to describe the characteristics of the table.

The following is an overview of the ERD of the system to be created:


Figure 3.2.ERD

### 3.3. Data Flow Diagrams (DFD)

Data flow diagram abbreviated as DFD or data flow diagram is a diagram that describes the flow of data in a system. In this system, there are several DFD levels which are described below.

#### 3.3.1. Context Diagram (CD)

Context Diagram (CD) or context diagram is the highest level of DFD. This diagram illustrates the data flow on a global system. According to Afyenni, the context diagram should only describe one process, not more, and not describe the data store (2014). This context diagram also describes the external entity with the system in general. The following is an illustration of the system context diagram that will be made:


Figure 3.3. *Context Diagram / Context Diagram*

#### 3.3.2. Data Flow Diagram Level 0 (DFD 0)

Data Flow Diagram level 0 or DFD 0 is a DFD that describes the processes that are in the context diagram. The following is an overview of the DFD 0 of the system to be created.

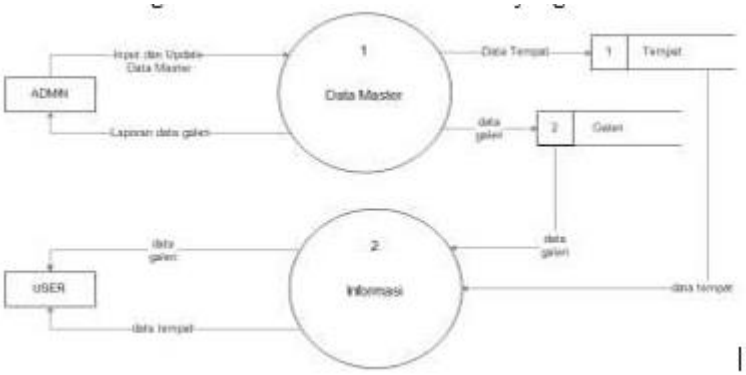



Figure 3.4 DFD 0

The image above is an image of level 0 data flow diagram which has 2 processes which are explained as follows:

a. Master data
   The master data process is the process of entering data carried out by the admin. The data entered is housing data stored in a table where and gallery data is stored in the gallery table.
b. BFS Mapping

The BFS mapping process is a mapping process with BFS calculations this process is carried out by the user and the mapping data is taken from the place table.

### 3.3.4. Data Flow Diagram Level 1 (DFD 1)

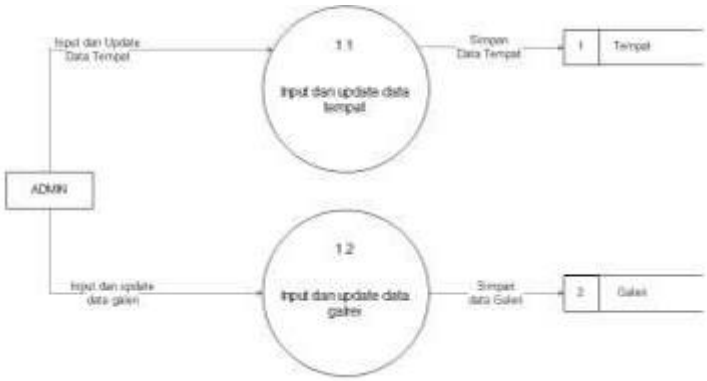



Figure 3.5. *Data Flow Diagram Level 1*

The image above is a data flow diagram level 1 which has 2 processes, which are explained as follows:

a. Input and update place data
   The process of input and update of place data is a process carried out by the admin to add and change housing data and the table used to store is the place table.
b. Input and update Gallery data
   The process of entering and updating Gallery data is the process carried out by the admin to add and change Gallery data and the table used to store is the Gallery table.

### 3.3.4. Data Flow Diagram Level 2 (DFD 2)

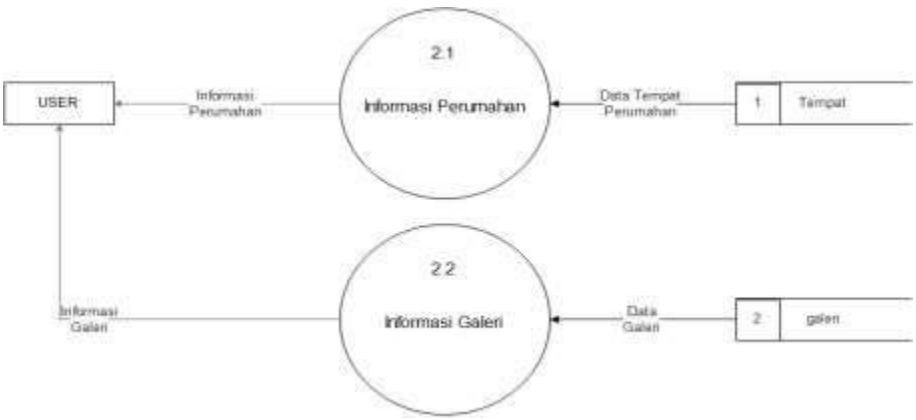



Figure 3.6. *Data Flow Diagram Level 2*

The image above is an image of level 2 data flow diagram which has 2 processes which are explained as follows:

a. Housing Information
   Housing information process is a process to view housing information.
b. Gallery Information
   Gallery information process is a process to view housing gallery information.

## 4. TESTING AND RESULTS

### 4.1. Testing the Breadth First Search Algorithm

Application performance testing is testing how the application performs in carrying out the methods applied in this application. In this test, the results of the system application path output will be compared with the manual

calculation of the Breadth First Search method. In this test, a 5 radius will be taken from the user's location point, and then it will be matched with the application results and manual calculations.

**1.** Testing Radius 1 Km.

Figure 4.1 *Testing Results Radius 1 Km*

From the 1 km radius test results in Figure 4.1, 5 housing data are obtained by calculating the distance based on a 1 km radius. The color of the lines in the 1 Km radius test is distinguished by the radius of the level. 1 km radius in green.

| Vertex M | | | | | |
|---|---|---|---|---|---|
| Vertex Get out of the Queue | Vertex Get into the Queue | | | | |
| 0(L0) | 0(L0) | | | | |
| 47 (L1) | 47 (L1) | 56 (L1) | 57 (L1) | 75 (L1) | 79 (L1) |
| 56 (L1) | 56 (L1) | 57 (L1) | 75 (L1) | 79 (L1) | |
| 57 (L1) | 57 (L1) | 75 (L1) | 79 (L1) | | |
| 75 (L1) | 75 (L1) | 79 (L1) | | | |
| 79 (L1) | 79 (L1) | | | | |

Figure 4.2 *Vertex M Radius 1 Km*

In Figure 4.2 the results of the Vertex M radius of 1 km are the results of map drawing with the Breadth First Search method using a technique where the first step is the expanded root node, after that then all successors of the root node are also expanded. This continues to be done repeatedly until the leaf (the node at the lowest level no longer has a successor). To calculate the rarity between housing nodes using the Euclidean Distance formula, the following is a manual calculation and in table 4.1 is the search result for a radius of 1 km.

$$Node\ 0-47 = (\sqrt{(7.4652298 - 7.463635164)^2 + (112.7011641 - 112.6958789)^2})\ x\ 111.319$$

$Node$ 0-56 = 0.614

$$Node\ 0-56 = (\sqrt{(7.4632597 - 7.463635164)^2 + (112.6926238 - 112.6958789)^2})\ x\ 111.319$$

$Node$ 0-56 = 0.364

$$Node\ 0-57 = (\sqrt{(7.4638844 - 7.463635164)^2 + (112.6936773 - 112.6958789)^2})\ x\ 111.319$$

$Node$ 0-57 = 0.246

$$Node\ 0-75 = (\sqrt{(7.4667163 - 7.463635164)^2 + (112.6935328 - 112.6958789)^2})\ x\ 111.319$$

$Node$ 0-75 = 0.431

$$Node\ 0-79 = (\sqrt{(7.462136 - 7.463635164)^2 + (112.6905656 - 112.6958789)^2})\ x\ 111.319$$

$Node$ 0-79 = 0.614

Table 4.1 *Calculation Results of 1 Km Radius*

| ***NODE*** | | **Koordinat *Node*(1) (x,y)** | | **Koordinat *Node*(2) (x,y)** | | **KM Distance** |
|---|---|---|---|---|---|---|
| **0** | 47 | -7.463635164 | 112.6958789 | -7.4652298 | 112.7011641 | 0.614 |
| **0** | 56 | -7.463635164 | 112.6958789 | -7.4632597 | 112.6926238 | 0.364 |
| **0** | 57 | -7.463635164 | 112.6958789 | -7.4638844 | 112.6936773 | 0.246 |
| **0** | 75 | -7.463635164 | 112.6958789 | -7.4667163 | 112.6935328 | 0.431 |
| **0** | 79 | -7.463635164 | 112.6958789 | -7.462136 | 112.6905656 | 0.614 |

**2.** Testing Radius 2 Km

Figure 4.3 *Testing Results Radius 2 Km*

From the results of the 2 km radius test in Figure 4.3, 6 housing data is obtained by calculating the distance based on a 2 km radius. The color of the lines in the 2 Km radius test is distinguished based on the radius level. 1 km radius in green, 2 km radius in red.

| Vertex M | | | | |
|---|---|---|---|---|
| Vertex Get out of the Queue | Vertex Get into the Queue | | | |
| 0(L0) | 0(L0) | | | |
| 49 (L1) | 49 (L1) | 50 (L1) | 68 (L1) | |
| 50 (L1) | 50 (L1) | 68 (L1) | 54 (L2) | |
| 68 (L1) | 68 (L1) | 54 (L2) | 61 (L2) | 70 (L2) |
| 54 (L2) | 54 (L2) | 61 (L2) | 65 (L2) | 70 (L2) |
| 61 (L2) | 61 (L2) | 65 (L2) | 70 (L2) | |
| 65 (L2) | 65 (L2) | 70 (L2) | | |
| 70 (L2) | 70 (L2) | | | |

Figure 4.4 *Vertex M Radius 2 Km*

In Figure 4.4 the results of the Vertex M radius of 2 km are the results of map drawing with the Breadth First Search method using a technique where the first step is the expansion of the root node, after which all successors of the root node are also expanded. This continues to be done repeatedly until the leaf (the node at the lowest level no longer has a successor). To calculate the rarity between residential nodes using the Euclidean Distance formula, in table 4.2 below are the results of the search for a radius of 2 km.

Table 4.2 *Calculation Results of a 2 Km Radius*

| *NODE* | | Koordinat *Node*(1) (x,y) | | Koordinat *Node*(2) (x,y) | | KM Distance |
|---|---|---|---|---|---|---|
| **0** | 49 | -7.488177753 | 112.7102587 | -7.4872951 | 112.7124319 | 0.261 |
| **0** | 50 | -7.488177753 | 112.7102587 | -7.4884217 | 112.7173893 | 0.794 |
| **0** | 68 | -7.488177753 | 112.7102587 | -7.4942286 | 112.7145774 | 0.827 |
| **49** | 54 | -7.4872951 | 112.7124319 | -7.4948051 | 112.7166897 | 0.961 |
| **50** | 54 | -7.4884217 | 112.7173893 | -7.4948051 | 112.7166897 | 0.714 |
| **68** | 54 | -7.4942286 | 112.7145774 | -7.4948051 | 112.7166897 | 0.243 |
| **50** | 61 | -7.4884217 | 112.7173893 | -7.4947229 | 112.7216833 | 0.848 |
| **68** | 61 | -7.4942286 | 112.7145774 | -7.4947229 | 112.7216833 | 0.792 |
| **68** | 65 | -7.4942286 | 112.7145774 | -7.4975703 | 112.7126795 | 0.427 |
| **50** | 70 | -7.4884217 | 112.7173893 | -7.4828688 | 112.7238632 | 0.949 |

**3.** Testing Radius 3 Km

Figure 4.5 *Testing Results Radius 3 Km*

From the results of testing a radius of 3 km in Figure 4.5 obtained 14 housing data with the calculation of the distance based on a radius of 3 km. The color of the lines on the 3 Km radius test is distinguished based on the radius level. 1 km radius in green. 2 km radius in red, 3 km radius in blue. To calculate the rarity between residential nodes using the Euclidean Distance formula, here are the search results for a radius of 3 km.

| Vertex M | | | | | | | |
|---|---|---|---|---|---|---|---|
| Vertex Get out of the Queue | Vertex Get into the Queue | | | | | | |
| **0 (L0)** | 0(L0) | | | | | | |
| **46 (L1)** | 46(L1) | 49(L1) | | | | | |
| **49 (L1)** | 49(L1) | 65(L2) | 175(L2) | 181(L2) | | | |
| **50 (L2)** | 50(L2) | 54(L2) | 65 (L2) | 68 (L2) | 175 (L2) | 181 (L2) | |
| **54 (L2)** | 54(L2) | 65(L2) | 68 (L2) | 175(L2) | 181 (L2) | 61 (L3) | 70 (L3) |
| **65 (L2)** | 65(L2) | 68(L2) | 175(L2) | 181(L2) | 61 (L3) | 183 (L3) | |
| **68 (L2)** | 68(L2) | 175(L2) | 181(L2) | 176(L3) | 183 (L3) | | |
| **175 (L2)** | 175(L2) | 181(L2) | 61 (L3) | | | | |
| **181 (L2)** | 181(L2) | 174(L3) | 176(L3) | 178(L3) | | | |
| **61 (L3)** | 61 (L3) | 70 (L3) | 174(L3) | 176(L3) | 178 (L3) | 183 (L3) | |
| **70 (L3)** | 70 (L3) | 174(L3) | 176(L3) | 178(L3) | 183 (L3) | | |
| **174 (L3)** | 174(L3) | 176(L3) | 178(L3) | 183(L3) | | | |
| **176 (L3)** | 176(L3) | 178(L3) | 183(L3) | | | | |
| **178 (L3)** | 178(L3) | 183(L3) | | | | | |
| **183 (L3)** | 183(L3) | | | | | | |

Figure 4.6. *Result of Vertex M Radius 3 Km*

In Figure 4.6, the result of Vertex M with a radius of 3 km is the result of map drawing with the Breadth First Search method using a technique where the first step is the expansion of the root node, after which all successors of the root node are also expanded. This continues to be done repeatedly until the leaf (the node at the lowest level has no successor anymore). To calculate the infrequency between housing nodes using the Euclidean Distance formula, here are the search results for a radius of 3 km.

Table 4.3. 3 Km Radius Calculation Results.

| ***NODE*** | | **Koordinat *Node*(1) (x,y)** | | **Koordinat *Node*(2) (x,y)** | | **KM Distance** |
|---|---|---|---|---|---|---|
| **0** | 46 | -7.4884028 | 112.705509 | -7.4966869 | 112.7049868 | 0.924 |
| **0** | 49 | -7.4884028 | 112.705509 | -7.4872951 | 112.7124319 | 0.78 |
| **49** | 50 | -7.4872951 | 112.7124319 | -7.4884217 | 112.7173893 | 0.565 |
| **49** | 54 | -7.4872951 | 112.7124319 | -7.4948051 | 112.7166897 | 0.961 |
| **46** | 65 | -7.4966869 | 112.7049868 | -7.4975703 | 112.7126795 | 0.861 |
| **49** | 68 | -7.4872951 | 112.7124319 | -7.4942286 | 112.7145774 | 0.807 |
| **46** | 175 | -7.4966869 | 112.7049868 | -7.5051839 | 112.7064416 | 0.959 |
| **46** | 181 | -7.4966869 | 112.7049868 | -7.5032509 | 112.7018842 | 0.808 |
| **50** | 61 | -7.4884217 | 112.7173893 | -7.4947229 | 112.7216833 | 0.848 |
| **54** | 61 | -7.4948051 | 112.7166897 | -7.4947229 | 112.7216833 | 0.555 |
| **68** | 61 | -7.4942286 | 112.7145774 | -7.4947229 | 112.7216833 | 0.792 |
| **50** | 70 | -7.4884217 | 112.7173893 | -7.4828688 | 112.7238632 | 0.949 |
| **175** | 174 | -7.5051839 | 112.7064416 | -7.5056096 | 112.7000048 | 0.718 |
| **181** | 174 | -7.5032509 | 112.7018842 | -7.5056096 | 112.7000048 | 0.335 |
| **65** | 176 | -7.4975703 | 112.7126795 | -7.5052552 | 112.7132548 | 0.857 |
| **175** | 176 | -7.5051839 | 112.7064416 | -7.5052552 | 112.7132548 | 0.758 |
| **175** | 178 | -7.5051839 | 112.7064416 | -7.5089945 | 112.7129722 | 0.841 |
| **54** | 183 | -7.4948051 | 112.7166897 | -7.5021308 | 112.7190587 | 0.857 |
| **65** | 183 | -7.4975703 | 112.7126795 | -7.5021308 | 112.7190587 | 0.872 |

The method used to measure the distance in table 4.3 above is the Euclidean Distance method, which is a method of finding the proximity of 2 variables, apart from being easy, this method is also more time efficient, and a fast process.

**4.** Testing Radius 4 Km

Figure 4.7 *Testing Results Radius 4 Km*

From the 4 km radius test results in Figure 4.7, 17 housing data are obtained by calculating the distance based on a 4 km radius. The color of the lines on the 4 Km Radius Test is distinguished based on the radius of the level. 4 km radius in purple color.

| Vertex M | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Vertex Get out of the Queue | Vertex Get into the Queue | | | | | | | |
| **0(L0)** | 0 (L0) | | | | | | | |
| **10 (L1)** | 10 (L1) | 12 (L1) | 28 (L1) | | | | | |
| **12 (L1)** | 12 (L1) | 28 (L1) | | | | | | |
| **28 (L1)** | 28 (L1) | 1 (L2) | 14 (L2) | | | | | |
| **1 (L2)** | 1 (L2) | 9 (L2) | 14 (L2) | 95 (L2) | 111 (L2) | | | |
| **9 (L2)** | 9 (L2) | 14 (L2) | 95 (L2) | 111(L2) | 11 (L3) | 31(L3) | | |
| **14 (L2)** | 14 (L2) | 95 (L2) | 111 (L2) | 24 (L3) | 97 (L3) | 108(L3) | | |
| **95 (L2)** | 95 (L2) | 111 (L2) | | | | | | |
| **111 (L2)** | 111 (L2) | 97 (L3) | 108 (L3) | | | | | |
| **11 (L3)** | 11 (L3) | 24 (L3) | 31 (L3) | 87 (L3) | 97 (L3) | 108(L3) | 113(L3) | 116(L3) |
| **24 (L3)** | 24 (L3) | 31 (L3) | 87 (L3) | 97 (L3) | 108 (L3) | 113(L3) | 116(L3) | 35 (L4) |
| **31 (L3)** | 31 (L3) | 87 (L3) | 97 (L3) | 108(L3) | 113 (L3) | 116(L3) | 25 (L4) | |
| **87 (L3)** | 87 (L3) | 97 (L3) | 108 (L3) | 113(L3) | 116 (L3) | | | |
| **97 (L3)** | 97 (L3) | 108 (L3) | 113 (L3) | 116(L3) | 96 (L4) | 100(L4) | | |
| **108 (L3)** | 108 (L3) | 113 (L3) | 116 (L3) | 29 (L4) | 98 (L4) | | | |
| **113 (L3)** | 113 (L3) | 116 (L3) | 29 (L4) | 98 (L4) | | | | |
| **116 (L3)** | 116 (L3) | 96 (L4) | 100 (L4) | 117(L4) | | | | |
| **25 (L4)** | 25 (L4) | 29 (L4) | 35 (L4) | 90 (L4) | 96 (L4) | 98 (L4) | 100(L4) | 117(L4) |
| **29 (L4)** | 29 (L4) | 35 (L4) | 90 (L4) | 96 (L4) | 98 (L4) | 100(L4) | 117(L4) | |
| **35 (L4)** | 35 (L4) | 90 (L4) | 96 (L4) | 98 (L4) | 100 (L4) | 117(L4) | | |
| **90 (L4)** | 90 (L4) | 96 (L4) | 98 (L4) | 100(L4) | 117 (L4) | | | |
| **96 (L4)** | 96 (L4) | 98 (L4) | 100 (L4) | 117(L4) | | | | |
| **98 (L4)** | 98 (L4) | 100 (L4) | 117 (L4) | | | | | |
| **100 (L4)** | 100 (L4) | 117 (L4) | | | | | | |
| **117 (L4)** | 117 (L4) | | | | | | | |

Figure 4.8. *Result Vertex M Radius 4 Km*

In Figure 4.8, the result of the Vertex M radius of 4 km is the result of map drawing with the Breadth First Search method using a technique where the first step is the expansion of the root node, after which all successors of the root node are also expanded. This continues to be done repeatedly until the leaf (the node at the lowest level that no longer has a successor). To calculate the rarity between housing nodes using the Euclidean Distance formula, here are the results of the search for a radius of 4 km.

Table 4.4. *Radius Calculation Results 4 Km*

| ***NODE*** | | **Koordinat *Node*(1) (x,y)** | | **Koordinat *Node*(2) (x,y)** | | **KM Distance** |
|---|---|---|---|---|---|---|
| **0** | 10 | -7.43724321 | 112.6891659 | -7.4389935 | 112.685581 | 0.444 |
| **0** | 12 | -7.43724321 | 112.6891659 | -7.431064 | 112.685259 | 0.813 |
| **0** | 28 | -7.43724321 | 112.6891659 | -7.436595 | 112.6927383 | 0.404 |
| **12** | 1 | -7.431064 | 112.685259 | -7.4255243 | 112.6793584 | 0.9 |
| **28** | 9 | -7.436595 | 112.6927383 | -7.442978 | 112.6987763 | 0.978 |
| **12** | 14 | -7.431064 | 112.685259 | -7.4266277 | 112.6837561 | 0.521 |
| **28** | 95 | -7.436595 | 112.6927383 | -7.4371607 | 112.6995163 | 0.757 |
| **28** | 111 | -7.436595 | 112.6927383 | -7.4343586 | 112.699801 | 0.824 |
| **1** | 11 | -7.4255243 | 112.6793584 | -7.4265167 | 112.6705908 | 0.982 |
| **9** | 24 | -7.442978 | 112.6987763 | -7.4463345 | 112.6923156 | 0.81 |
| **1** | 31 | -7.4255243 | 112.6793584 | -7.4323542 | 112.6745094 | 0.932 |
| **111** | 87 | -7.4343586 | 112.699801 | -7.4259988 | 112.6968647 | 0.986 |
| **9** | 97 | -7.442978 | 112.6987763 | -7.4388137 | 112.703012 | 0.661 |
| **95** | 97 | -7.4371607 | 112.6995163 | -7.4388137 | 112.703012 | 0.43 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **111** | 97 | -7.4343586 | 112.699801 | -7.4388137 | 112.703012 | 0.611 |
| **9** | 108 | -7.442978 | 112.6987763 | -7.4406166 | 112.7040418 | 0.642 |
| **95** | 108 | -7.4371607 | 112.6995163 | -7.4406166 | 112.7040418 | 0.633 |
| **111** | 108 | -7.4343586 | 112.699801 | -7.4406166 | 112.7040418 | 0.841 |
| **111** | 113 | -7.4343586 | 112.699801 | -7.4260372 | 112.6978196 | 0.952 |
| **111** | 116 | -7.4343586 | 112.699801 | -7.4291968 | 112.7069266 | 0.979 |
| **24** | 25 | -7.4463345 | 112.6923156 | -7.4549121 | 112.6911032 | 0.964 |
| **97** | 29 | -7.4388137 | 112.703012 | -7.44649 | 112.707578 | 0.994 |
| **108** | 29 | -7.4406166 | 112.7040418 | -7.44649 | 112.707578 | 0.763 |
| **11** | 35 | -7.4265167 | 112.6705908 | -7.4318864 | 112.665191 | 0.847 |
| **116** | 90 | -7.4291968 | 112.7069266 | -7.4270085 | 112.7081073 | 0.276 |
| **87** | 96 | -7.4259988 | 112.6968647 | -7.4250041 | 112.7057725 | 0.997 |
| **113** | 96 | -7.4260372 | 112.6978196 | -7.4250041 | 112.7057725 | 0.892 |
| **116** | 96 | -7.4291968 | 112.7069266 | -7.4250041 | 112.7057725 | 0.484 |
| **97** | 98 | -7.4388137 | 112.703012 | -7.4346549 | 112.7100555 | 0.91 |
| **108** | 98 | -7.4406166 | 112.7040418 | -7.4346549 | 112.7100555 | 0.942 |
| **116** | 98 | -7.4291968 | 112.7069266 | -7.4346549 | 112.7100555 | 0.7 |
| **87** | 100 | -7.4259988 | 112.6968647 | -7.4227581 | 112.69559 | 0.387 |
| **113** | 100 | -7.4260372 | 112.6978196 | -7.4227581 | 112.69559 | 0.441 |
| **113** | 117 | -7.4260372 | 112.6978196 | -7.4228317 | 112.705926 | 0.97 |
| **116** | 117 | -7.4291968 | 112.7069266 | -7.4228317 | 112.705926 | 0.717 |

The method used to measure the distance in table 4.4 above is the Euclidean Distance method, which is a method of finding the proximity of 2 variables, besides being easy, this method is also more time efficient, and a fast process.

**5.** Testing Radius 5 Km

Figure 4.9 *Testing Results Radius 5 Km*

From the results of testing a radius of 5 km in Figure 4.9, 33 housing data obtained by calculating the distance based on a radius of 5 km. The color of the lines on the 5 Km Radius Testing is distinguished based on the radius of the level. 5 km radius in black.

| Vertex M | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Vertex Get out of the Queue | Vertex Get into the Queue | | | | | | | |
| 0(L0) | 0(L0) | | | | | | | |
| 93 (L1) | 93 (L1) | 94(L1) | 104(L1) | 114(L1) | | | | |
| 94 (L1) | 94 (L1) | 104(L1) | 114(L1) | 23 (L2) | 37(L2) | 98(L2) | 101(L2) | 109(L2) |
| 104(L1) | 104(L1) | 114(L1) | 23 (L2) | 37 (L2) | 98(L2) | 101(L2) | 109(L2) | |
| 114(L1) | 114(L1) | 2(L2) | 3(L2) | | | | | |
| 2 (L2) | 2(L2) | 3 (L2) | 23(L2) | 37 (L2) | 98(L2) | 101(L2) | 109(L2) | |
| 3 (L2) | 3(L2) | 23 (L2) | 37 (L2) | 98 (L2) | 101(L2) | 109(L2) | 36(L3) | |
| 23 (L2) | 23(L2) | 37 (L2) | 98 (L2) | 101(L2) | 109(L2) | 36(L3) | | |
| 37 (L2) | 37(L2) | 98 (L2) | 101(L2) | 109(L2) | 29(L3) | | | |
| 98 (L2) | 98 (L2) | 101(L2) | 109(L2) | 29(L3) | | | | |
| 101(L2) | 101(L2) | 109(L2) | 90 (L3) | 97 (L3) | 108(L3) | 116(L3) | | |
| 109(L2) | 109(L2) | | | | | | | |
| 29 (L3) | 29 (L3) | 36 (L3) | 90 (L3) | 97 (L3) | 108(L3) | 116(L3) | | |
| 36 (L3) | 36 (L3) | 90 (L3) | 97 (L3) | 108(L3) | 116(L3) | | | |
| 90 (L3) | 90 (L3) | 97 (L3) | 108(L3) | 116(L3) | 102(L4) | | | |
| 97 (L3) | 97 (L3) | 108(L3) | 116(L3) | 96 (L4) | 117(L4) | | | |
| 108(L3) | 108(L3) | 116(L3) | 9 (L4) | 95 (L4) | 111(L4) | | | |
| 116(L3) | 116(L3) | 9 (L4) | 95 (L4) | 111(L4) | | | | |
| 9 (L4) | 9 (L4) | 95 (L4) | 96 (L4) | 102(L4) | 111(L4) | 117(L4) | | |
| 95 (L4) | 95 (L4) | 96 (L4) | 102(L4) | 111(L4) | 117(L4) | 24 (L5) | 28 (L5) | |
| 96 (L4) | 96 (L4) | 102(L4) | 111(L4) | 117(L4) | 28 (L5) | | | |
| 102(L4) | 102(L4) | 111(L4) | 117(L4) | 87 (L5) | 113(L5) | | | |
| 111(L4) | 111(L4) | 117(L4) | 99 (L5) | | | | | |
| 117(L4) | 117(L4) | 28 (L5) | 87 (L5) | 113(L5) | | | | |
| 24 (L5) | 24 (L5) | 28 (L5) | 87 (L5) | 88 (L5) | 99 (L5) | 112(L5) | 113(L5) | |
| 28 (L5) | 28 (L5) | 87 (L5) | 88 (L5) | 99 (L5) | 112(L5) | 113(L5) | | |
| 87 (L5) | 87 (L5) | 88 (L5) | 99 (L5) | 112(L5) | 113(L5) | | | |
| 88 (L5) | 88 (L5) | 99 (L5) | 112(L5) | 113(L5) | | | | |
| 99 (L5) | 99 (L5) | 112(L5) | 113(L5) | | | | | |
| 112(L5) | 112(L5) | 113(L5) | | | | | | |
| 113(L5) | 113(L5) | | | | | | | |

Figure 4.10 *Vertex M Radius 5 Km*

In Figure 4.10, the results of the Vertex M radius of 5 km are the results of map drawing with the Breadth First Search method using a technique where the first step is the expansion of the root node, after which all successors of the root node are also expanded. This continues to be done repeatedly until the leaf (the node at the lowest level no longer has a successor). To calculate the rarity between residential nodes using the Euclidean Distance formula, in table 4.5 below are the results of the search for a radius of 5 km.

Table 4.5 *Calculation Results of a 5 Km Radius*

| ***NODE*** | | **Koordinat *Node*(1) (x,y)** | | **Koordinat *Node*(2) (x,y)** | | **KM Distance** |
|---|---|---|---|---|---|---|
| **0** | 93 | -7.429606206 | 112.7205629 | -7.4353324 | 112.7137881 | 0.987 |
| **0** | 94 | -7.429606206 | 112.7205629 | -7.4351766 | 112.7144362 | 0.921 |
| **0** | 104 | -7.429606206 | 112.7205629 | -7.4298675 | 112.7253196 | 0.53 |
| **0** | 114 | -7.429606206 | 112.7205629 | -7.4365316 | 112.7225124 | 0.8 |
| **104** | 2 | -7.4298675 | 112.7253196 | -7.4365516 | 112.7279435 | 0.799 |
| **114** | 2 | -7.4365316 | 112.7225124 | -7.4365516 | 112.7279435 | 0.604 |
| **104** | 3 | -7.4298675 | 112.7253196 | -7.438072 | 112.7250027 | 0.913 |
| **114** | 3 | -7.4365316 | 112.7225124 | -7.438072 | 112.7250027 | 0.325 |
| **93** | 23 | -7.4353324 | 112.7137881 | -7.4400654 | 112.7134364 | 0.528 |
| **94** | 23 | -7.4351766 | 112.7144362 | -7.4400654 | 112.7134364 | 0.555 |
| **93** | 37 | -7.4353324 | 112.7137881 | -7.4430857 | 112.7130962 | 0.866 |
| **94** | 37 | -7.4351766 | 112.7144362 | -7.4430857 | 112.7130962 | 0.892 |
| **93** | 98 | -7.4353324 | 112.7137881 | -7.4346549 | 112.7100555 | 0.422 |
| **94** | 98 | -7.4351766 | 112.7144362 | -7.4346549 | 112.7100555 | 0.491 |
| **93** | 101 | -7.4353324 | 112.7137881 | -7.4397912 | 112.7187636 | 0.743 |
| **94** | 101 | -7.4351766 | 112.7144362 | -7.4397912 | 112.7187636 | 0.704 |
| **114** | 101 | -7.4365316 | 112.7225124 | -7.4397912 | 112.7187636 | 0.553 |
| **93** | 109 | -7.4353324 | 112.7137881 | -7.4389128 | 112.7163001 | 0.486 |
| **94** | 109 | -7.4351766 | 112.7144362 | -7.4389128 | 112.7163001 | 0.464 |
| **114** | 109 | -7.4365316 | 112.7225124 | -7.4389128 | 112.7163001 | 0.74 |
| **23** | 29 | -7.4400654 | 112.7134364 | -7.44649 | 112.707578 | 0.967 |
| **37** | 29 | -7.4430857 | 112.7130962 | -7.44649 | 112.707578 | 0.721 |
| **2** | 36 | -7.4365516 | 112.7279435 | -7.4377062 | 112.7315377 | 0.42 |
| **3** | 36 | -7.438072 | 112.7250027 | -7.4377062 | 112.7315377 | 0.728 |
| **98** | 90 | -7.4346549 | 112.7100555 | -7.4270085 | 112.7081073 | 0.878 |
| **98** | 97 | -7.4346549 | 112.7100555 | -7.4388137 | 112.703012 | 0.91 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **98** | 108 | -7.4346549 | 112.7100555 | -7.4406166 | 112.7040418 | 0.942 |
| **98** | 116 | -7.4346549 | 112.7100555 | -7.4291968 | 112.7069266 | 0.7 |
| **97** | 9 | -7.4388137 | 112.703012 | -7.442978 | 112.6987763 | 0.661 |
| **108** | 9 | -7.4406166 | 112.7040418 | -7.442978 | 112.6987763 | 0.642 |
| **97** | 95 | -7.4388137 | 112.703012 | -7.4371607 | 112.6995163 | 0.43 |
| **108** | 95 | -7.4406166 | 112.7040418 | -7.4371607 | 112.6995163 | 0.633 |
| **90** | 96 | -7.4270085 | 112.7081073 | -7.4250041 | 112.7057725 | 0.342 |
| **116** | 96 | -7.4291968 | 112.7069266 | -7.4250041 | 112.7057725 | 0.484 |
| **36** | 102 | -7.4377062 | 112.7315377 | -7.4316014 | 112.7370408 | 0.914 |
| **97** | 111 | -7.4388137 | 112.703012 | -7.4343586 | 112.699801 | 0.611 |
| **108** | 111 | -7.4406166 | 112.7040418 | -7.4343586 | 112.699801 | 0.841 |
| **116** | 111 | -7.4291968 | 112.7069266 | -7.4343586 | 112.699801 | 0.979 |
| **90** | 117 | -7.4270085 | 112.7081073 | -7.4228317 | 112.705926 | 0.524 |
| **116** | 117 | -7.4291968 | 112.7069266 | -7.4228317 | 112.705926 | 0.717 |
| **9** | 24 | -7.442978 | 112.6987763 | -7.4463345 | 112.6923156 | 0.81 |
| **9** | 28 | -7.442978 | 112.6987763 | -7.436595 | 112.6927383 | 0.978 |
| **95** | 28 | -7.4371607 | 112.6995163 | -7.436595 | 112.6927383 | 0.757 |
| **111** | 28 | -7.4343586 | 112.699801 | -7.436595 | 112.6927383 | 0.824 |
| **96** | 87 | -7.4250041 | 112.7057725 | -7.4259988 | 112.6968647 | 0.997 |
| **111** | 87 | -7.4343586 | 112.699801 | -7.4259988 | 112.6968647 | 0.986 |
| **117** | 88 | -7.4228317 | 112.705926 | -7.4144723 | 112.7075303 | 0.947 |
| **102** | 99 | -7.4316014 | 112.7370408 | -7.4235189 | 112.7341958 | 0.953 |
| **117** | 112 | -7.4228317 | 112.705926 | -7.4149731 | 112.7085324 | 0.921 |
| **96** | 113 | -7.4250041 | 112.7057725 | -7.4260372 | 112.6978196 | 0.892 |
| **111** | 113 | -7.4343586 | 112.699801 | -7.4260372 | 112.6978196 | 0.952 |
| **117** | 113 | -7.4228317 | 112.705926 | -7.4260372 | 112.6978196 | 0.97 |

4.2. Analysis of test results

From the Breadth First Search test applied in this application it can run well, testing is done by testing searches based on a 1 km radius, 2 km radius, 3 km radius, 4 km radius, 5 km radius. Each radius in the search results is different because the starting point of the test is at five different location points. The following is in table 4.6 Processing Time Test Results with data displayed housing data and the search process time.

Table 4.6 *Processing Time Test Results*

| Testing Name | Housing Yield | Processing Time |
|---|---|---|
| Radius 1 Km | 5 | 230 ms |
| Radius 2 Km | 6 | 230 ms |
| Radius 3 Km | 14 | 230 ms |
| Radius 4 Km | 17 | 390 ms |
| Radius 5 Km | 33 | 290 ms |

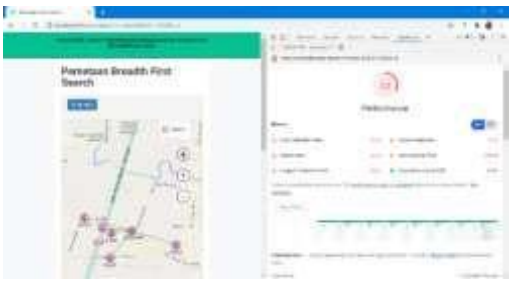


Figure 4.11. Results of Testing Time Radius 1 Km

Figure 4.12. *Result of Testing Time Radius 2 Km*

Figure 4.13. *Result of Testing Time Radius 3 Km*

Figure 4.14. *Testing Results Radius 4 Km Processing Time*

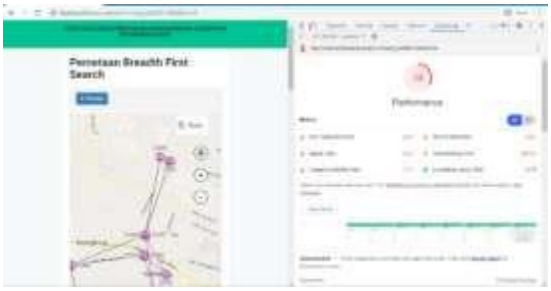


Figure 4.15. *Results of Testing Time Radius 5 Km*

## 5. CONCLUSIONS AND SUGGESTIONS

### 5.1. Conclusion

Based on the results of the research and discussion that has been done, it can be concluded that:

[1]. Breadth-first search (BFS) performs the searching process on all nodes that are at the same level or hierarchy first before continuing the searching process on the nodes at the next level.
[2]. The application can display Breadth-first search (BFS) and Breadth-first search (BFS) nodes which is traversed according to the start and end points defined by the user.
[3]. Applications can measure path lines using the Euclidean Distance formula method.

### 5.2. Suggestion

After evaluating the application as a whole, it is hoped that the results of this research can be further developed with development suggestions as follows:

[1]. For the development of this information system, the mapping of housing locations should be enlarged, not only for housing in Sidoarjo.
[2]. Implementing this geographic information system concept on the Android platform in order to take advantage of the geolocation or GPS feature and increase system mobility.
[3]. For the creation of Djkstra nodes and paths, so that they are made closer to the curves of the road, so that the resulting difference is not too far from the distance shown by Google Maps.

## 6. REFERENCES

[1]. Dharwiyanti, S., Wahono, .S.R., 2003, Pengantar Unified Modelling Language (UML), IlmuComputer.org, terakhir diakses pada 20 Desember 2018.
[2]. Kadir, A., 1999, Konsep dan Tuntunan Praktis Basis Data, Andi, Yogyakarta.
[3]. Kusrini, 2007, Strategi Perancangan dan Pengelolaan Database, Edisi I, Andi, Yogyakarta.
[4]. Nanda Bagus Maha Putra (2015), Rancang Bangun Aplikasi Android Wisata Kota Semarang Menggunakan Algoritma BFS, ( eprints.undip.ac.id/21134/1/ , diakses tanggal 22 Mei 2018 ).
[5]. Delima Zai (2016), Simulasi Rute Terpendek Lokasi Pariwisata Di Nias Dengan Metode Breadth First Search Dan Tabu Search, (eprints.undip.ac.id/31231/2/, diakses tanggal 30 Juni 2018 ).

[6]. Budianto (2016), Implementasi Algoritma Breadth First Search Dan Obstacle Detection Dalam Penelusuran Labirin Dinamis Menggunakan Robot Lego, (eprints.undip.ac.id/534321/5/, diakses tanggal 30 Juni 2018).