

SISTEM INFORMASI GEOGRAFIS PEMETAAN LOKASI PERUMAHAN MENGGUNAKAN ALGORITMA BREADTH FIRST SEARCH BERBASIS WEB (Studi Kasus : Kabupaten Sidoarjo)

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

^{1,2,3}Program Studi Teknik Informatika, Fakultas Teknik

Universitas Bhayangkara – Surabaya

e-mail: dwibudim@gmail.com

ABSTRAK

Kabupaten Sidoarjo sebagai salah satu kota dengan keadaan penduduk yang makin padat dan kebutuhan masyarakat yang ingin mencari informasi tentang perumahan dengan cepat, membuat Sistem Informasi Geografis sangat diperlukan, terutama dalam hal pencarian lokasi perumahan. Oleh karena itu, penulis membuat sistem informasi geografis perumahan di Kabupaten Sidoarjo berbasis web menggunakan Breadth First Search untuk merancang antarmuka dan logika sistem, MySQL untuk perancangan database sistem dan Bing Api untuk pemetaan. Informasi yang disajikan berupa nama perumahan, alamat, foto perumahan, gambar tipe rumah, denah rumah, ketersediaan dan harga dari tiap tipe rumah. Dengan adanya SIG (Sistem Informasi Geografis) ini diharapkan masyarakat dapat lebih mudah dan cepat mendapatkan informasi perumahan di Kabupaten Sidoarjo. Hasil penelitian ini menampilkan detail lokasi perumahan dari lokasi user sesuai radius yang akan di pilih dan mengetahui perkiraan jarak waktu perjalanan menuju ke lokasi perumahan tersebut.

Kata Kunci: Berbasis Web, Bing API, Perumahan, SIG

1. PENDAHULUAN

Kabupaten Sidoarjo merupakan Kabupaten yang terletak di Provinsi Jawa Timur. Letak geografis Kabupaten Sidoarjo berada sangat dekat dan yang berbatasan langsung dengan kota metropolitan sekaligus Ibukota Provinsi Jawa Timur yaitu kota Surabaya. Sebagai kota penyanggah dari kota Surabaya sudah tentu menjadikan Kabupaten Sidoarjo menjadikan lahan berinvestasi yang sangat baik dan strategis di Kawasan Jawa Timur.

Banyaknya *developer property* di Kabupaten Sidoarjo membuat pembeli perumahan harus menyeleksi banyak kriteria yang diinginkan, proses untuk membeli rumah menjadi hal yang sangat penting bagi calon pembeli tidak salah mengambil keputusan yang salah. Untuk mengatasi membantu calon pembeli memilih kriteria yang diinginkan diperlukan *system* informasi yang mampu mengintegrasikan dan mengolah data non spasial maupun spasial, khususnya pemetaan perumahan dengan sarana umum, gedung pendidikan dan gedung kesehatan. Hal ini lah yang menyebabkan calon pembeli masih terbatas informasi untuk membeli rumah. Di butuhnya media yang dapat menyampaikan informasi yang berkaitan dengan pemetaan wilayah pembangunan perumahan, supaya permasalahan seperti pencarian perumahan bersifat dinamis. Penggunaan teknologi *Geographic Information System (GIS)* sangat membantu dalam pemetaan / penentuan titik perumahan yang ada pada Kabupaten Sidoarjo.

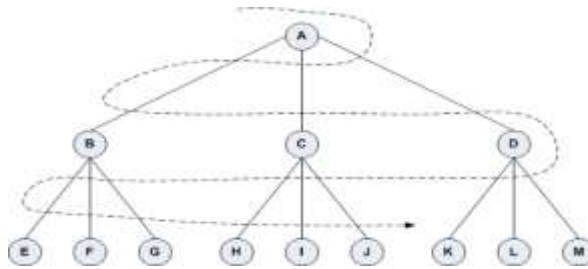
2. LANDASAN TEORI

2.1 Pengertian Sistem Informasi Geografis

Geographic Information System atau lebih dikenal dengan sebutan Sistem Informasi Geografis, merupakan suatu sistem informasi yang terintegrasi dan secara khusus digunakan untuk mengelola berbagai data yang mempunyai suatu informasi dalam bentuk spasial keruangan, dimana teknologi Sistem Informasi Geografis ini dapat digunakan untuk investigasi ilmiah, pengelolaan sumber daya, perencanaan pembangunan, kartografi, bahkan data juga digunakan untuk melakukan perencanaan terhadap rute. Secara praktisnya, kita bisa menyebutkan bahwa Sistem Informasi Geografis adalah suatu sistem komputerisasi yang mempunyai kemampuan untuk membangun, mengelola, menganalisa, menyimpan, dan menampilkan suatu Sistem Informasi Geografis dalam bentuk pemetaan dimana *user* yang membangun data serta mengoperasikannya juga termasuk dari bagian sistem tersebut.

2.2 Metode *Breadth-First Search* (BFS)

Algoritma *Breadth-First Search* (BFS) atau dikenal juga dengan nama algoritma pencarian melebar adalah algoritma yang melakukan pencarian secara melebar yang mengunjungi simpul secara preorder yaitu mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga dengan simpul tersebut terlebih dahulu. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya. Jika graf berbentuk pohon berakar, maka semua simpul pada aras d dikunjungi lebih dahulu sebelum simpul-simpul pada aras $d+1$.

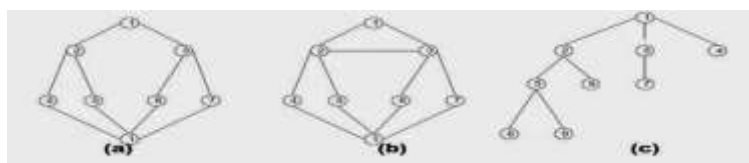


Gambar 2.1. Contoh Graf Pada Algoritma *Dijkstra*

Dalam algoritma BFS, simpul anak yang telah dikunjungi disimpan dalam suatu antrian. Antrian ini digunakan untuk mengacu simpul-simpul yang bertetangga dengannya yang akan dikunjungi kemudian sesuai urutan pengantrian. Untuk memperjelas cara kerja algoritma BFS beserta antrian yang digunakannya, berikut langkah-langkah algoritma BFS :

- Masukkan simpul ujung (akar) ke dalam antrian.
- Ambil simpul dari awal antrian, lalu cek apakah simpul merupakan solusi.
- Jika simpul merupakan solusi, pencarian selesai dan hasil dikembalikan.
- Jika simpul bukan solusi, masukkan seluruh simpul yang bertetangga dengan simpul tersebut (simpul anak) ke dalam antrian.
- Jika antrian kosong dan setiap simpul sudah dicek, pencarian selesai dan mengembalikan hasil solusi tidak ditemukan.
- Ulangi pencarian dari langkah kedua.

Contohnya terlihat dibawah ini:

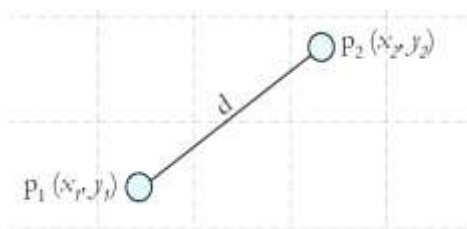


Gambar 2.2 Contoh BFS

Untuk melakukan proses searching pada semua node yang berada pada level atau hirarki yang sama terlebih dahulu sebelum melanjutkan proses searching pada node di level berikutnya. Urutan proses searching BFS ditunjukkan dalam contoh Gambar 3.3 Maka untuk urutan penyelesaiannya Gambar (a) BFS (1): 1, 2, 3, 4, 5, 6, 7, 1. Gambar (b) BFS (1): 1, 2, 3, 4, 5, 6, 7, 1 Gambar (c) BFS (1): 1, 2, 3, 4, 5, 6, 7, 8, 9.

2.3 Metode *Euclidean Distance*

Euclidean Distance adalah perhitungan jarak dari 2 buah titik dalam *Euclidean Space*. *Euclidean Space* diperkenalkan oleh *Euclid*, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. *Euclidean* ini berkaitan dengan Teorema Pythagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. *Euclidean Distance* merupakan fungsi heuristik yang diperoleh berdasarkan jarak langsung bebas hambatan seperti untuk mendapatkan nilai dari panjang garis diagonal pada segitiga. Tetapi sebelum mendapatkan hasil kedua titik harus direpresentasikan ke dalam koordinat 2 dimensi (x, y). Dua buah titik $p_1 = (x_1, y_1)$ dan $p_2 = (x_2, y_2)$ menjadi persamaan berikut.



$$\text{Euclidean Distance } (d) = (\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2})$$

Rumus *Euclidean*

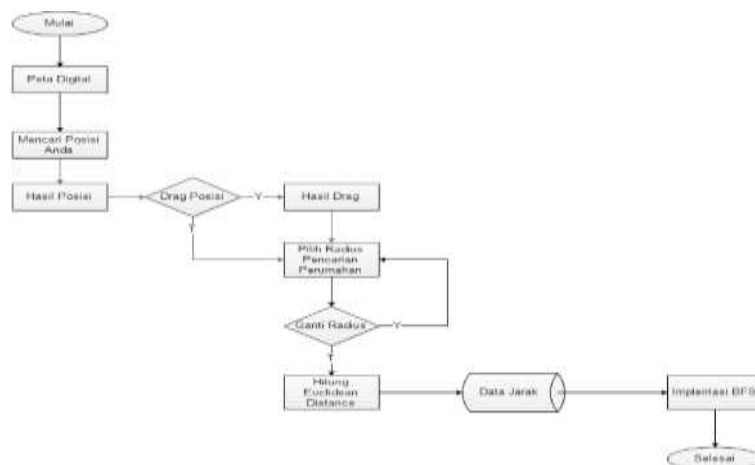
Sehingga dari Formula diatas kita dapat implementasi

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

3. ANALISIS DAN DESAIN SISTEM

3.1 Flowchart Sistem

Flowchart sistem adalah penggambaran secara grafis dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong analisis dan programmer untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian program. Berikut ini gambaran dari *flowchart* sistem yang akan dibuat.



Gambar 3.1. *Flowchart* sistem

3.2 Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD adalah suatu diagram yang menggambarkan susunan tabel beserta atribut-atributnya dan menentukan relasi antar tabel. ERD juga menerangkan relasi antara atribut dengan tabelnya, dimana atribut mempunyai fungsi untuk mendeskripsikan karakteristik dari tabel tersebut. Berikut ini gambaran ERD dari sistem yang akan dibuat:



Gambar 3.2.ERD

3.3 Data Flow Diagram (DFD)

Data flow diagram yang disingkat DFD atau diagram arus data adalah suatu diagram yang menjelaskan tentang aliran data di dalam suatu sistem. Dalam sistem ini, terdapat beberapa level DFD yang dijelaskan dibawah ini.

3.3.1. Context Diagram (CD)

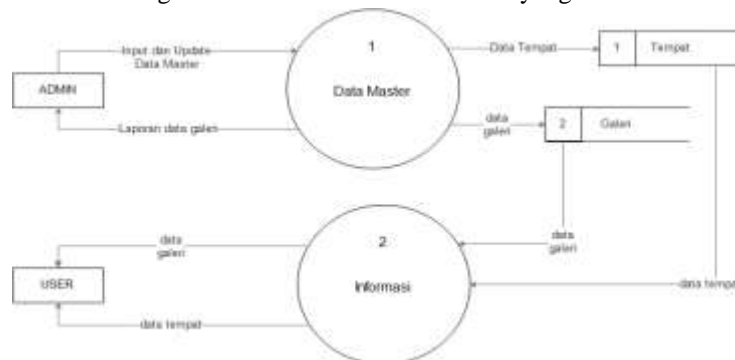
Context Diagram (CD) atau Diagram konteks adalah tingkatan DFD yang paling tinggi. Diagram ini menggambarkan aliran data pada sistem secara global. menurut Afyenni, Diagram konteks harus menggambarkan satu proses saja, tidak boleh lebih, dan tidak menggambarkan data store (2014). Diagram konteks ini juga menggambarkan kesatuan luar (*external entity*) dengan sistem secara umum. Berikut ini gambaran dari diagram konteks sistem yang akan dibuat:



Gambar 3.3. Context Diagram/Diagram Konteks

3.3.2. Data Flow Diagram Level 0 (DFD 0)

Data Flow Diagram level 0 atau DFD 0 adalah sebuah DFD yang menjabarkan proses-proses yang ada didalam diagram konteks. Berikut ini gambaran dari DFD 0 dari sistem yang akan dibuat.

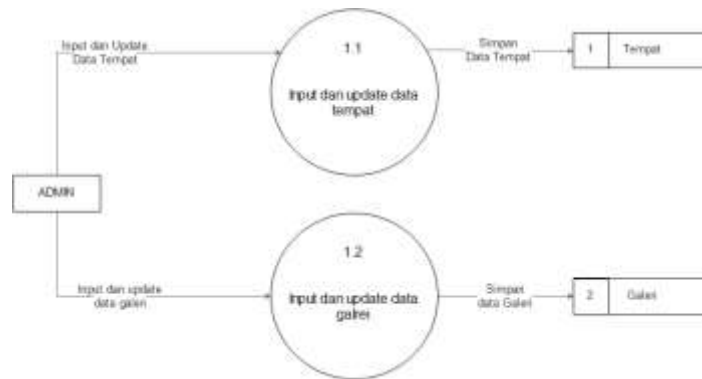


Gambar 3.4. DFD 0

Gambar diatas adalah gambar data flow diagram level 0 yang memiliki 2 proses yang penjelasannya sebagai berikut:

- a. Data master
Proses data master adalah proses memasukan data yang dilakukan oleh admin data yang dimasukan adalah data perumahan yang disimpan pada tabel tempat dan data galeri disimpan pada tabel galeri.
- b. Pemetaan BFS
Proses pemetaan BFS adalag proses pemetaan dengan perhitungan BFS proses ini dilakukan oleh user dan data pemetaan ini diambil dari tabel tempat.

3.3.3. Data Flow Diagram Level 1 (DFD 1)

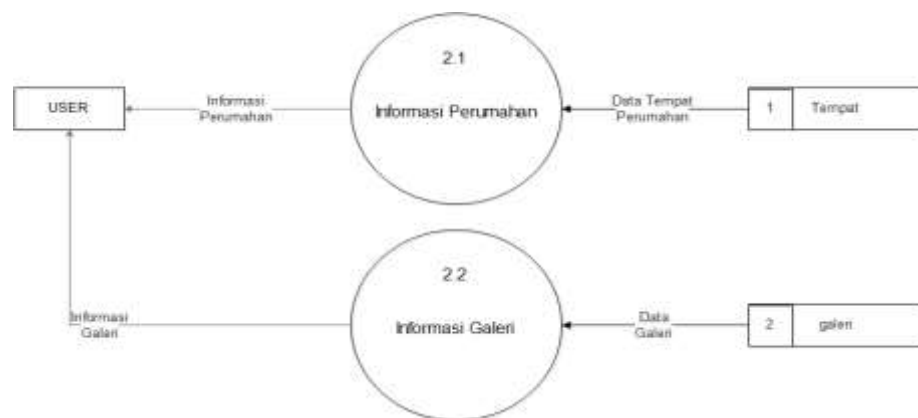


Gambar 3.5. Data Flow Diagram Level 1

Gambar diatas adalah gambar data flow diagram level 1 yang memiliki 2 proses yang penjelasannya sebagai berikut:

- a. *Input dan update* data tempat
Proses *Input dan update* data tempat adalah proses yang dilakukan oleh admin untuk menambah dan merubah data perumahan dan tabel yang digunakan untuk menyimpan adalah tabel tempat.
- b. *Input dan update* data Galeri
Proses *Input dan update* data Galeri adalah proses yang dilakukan oleh admin untuk menambah dan merubah data Galeri dan tabel yang digunakan untuk menyimpan adalah tabel Galeri.

3.3.4. Data Flow Diagram Level 2 (DFD 2)



Gambar 3.6. Data Flow Diagram Level 2

Gambar diatas adalah gambar data flow diagram level 2 yang memiliki 2 proses yang penjelasannya sebagai berikut:

- a. Infomasi Perumahan
Proses informasi perumahan adalah proses untuk melihat infomasi perumahan.
- b. Informasi Galeri
Proses informasi galeri adalah proses untuk melihat infomasi galeri perumahan.

4. PENGUJIAN DAN HASIL

4.1. Pengujian Algoritma *Breadth First Search*

Pengujian performa aplikasi adalah menguji bagaimana performa aplikasi dalam menjalankan metode yang diterapkan dalam aplikasi ini. Dalam pengujian kali ini, hasil dari keluaran jalur aplikasi secara sistem akan dibandingkan dengan perhitungan manual metode *Breadth First Search*. Dalam pengujian ini, akan diambil 5 radius dari titik lokasi *user*, dan kemudian akan dicocokkan dengan hasil aplikasi dan hitung-hitungan secara manual.

1. Pengujian Radius 1 Km.



Gambar 4.1 Hasil Pengujian Radius 1 Km

Dari hasil pengujian radius 1 km pada gambar 4.1 didapatkan 5 data perumahan dengan dengan perhitungan jarak berdasarkan radius 1 Km. Warna garis pada pengujian radius 1 Km dibedakan berdasarkan radius levelnya. Radius 1 km dengan warna hijau.

Vertex M					
Vertex Keluar dari Antrian	Vertex Masuk ke Antrian				
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)				

Gambar 4.2 Hasil Vertex M Radius 1 Km

Pada gambar 4.2 hasil Vertex M radius 1 km adalah hasil penggambaran peta dengan metode *Breadth First Search* menggunakan teknik dimana langkah pertamanya adalah *root node* diekspansi, setelah itu

dilanjutkan semua *successor* dari *root node* juga di-*expand*. Hal ini terus dilakukan berulang-ulang hingga *leaf* (*node* pada level paling bawah yang sudah tidak mempunyai *successor* lagi). Untuk menghitung jarak antar *node* perumahan menggunakan rumus *Euclidean Distance*, berikut ini perhitungan manual dan pada tabel 4.1 merupakan hasil pencarian radius 1 km.

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

$$\text{Node } 0-56 = 0.614$$

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

$$\text{Node } 0-56 = 0.364$$

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

$$\text{Node } 0-57 = 0.246$$

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

$$\text{Node } 0-75 = 0.431$$

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

$$\text{Node } 0-79 = 0.614$$

Tabel 4.1 Hasil Perhitungan Radius 1 Km

NODE		Koordinat Node(1) (x,y)		Koordinat Node(2) (x,y)		Jarak KM
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. Pengujian Radius 2 Km.



Gambar 4.3 Hasil Pengujian Radius 2 Km

Dari hasil pengujian radius 2 km pada gambar 4.3 didapatkan 6 data perumahan dengan dengan perhitungan jarak berdasarkan radius 2 Km. warna garis pada pengujian radius 2 Km dibedakan berdasarkan radius levelnya. Radius 1 km dengan warna hijau, radius 2 km dengan warna merah.

Vertex M				
Vertex Keluar dari Antrian	Vertex Masuk ke Antrian			
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)			

Gambar 4.4 Hasil Vertex M Radius 2 Km

Pada Gambar 4.4 hasil Vertex M radius 2 km adalah hasil penggambaran peta dengan metode *Breadth First Search* menggunakan teknik dimana langkah pertamanya adalah *root node* diekspansi, setelah itu dilanjutkan semua *successor* dari *root node* juga di-*expand*. Hal ini terus dilakukan berulang-ulang hingga *leaf* (*node* pada level paling bawah yang sudah tidak mempunyai *successor* lagi). Untuk menghitung jarak antar *node* perumahan menggunakan rumus *Euclidean Distance*, pada table 4.2 berikut ini adalah hasil pencarian radius 2 km.

Tabel 4.2 Hasil Perhitungan Radius 2 Km

NODE		Koordinat Node(1) (x,y)		Koordinat Node(2) (x,y)		Jarak KM
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. Pengujian Radius 3 Km.



Gambar 4.5 Hasil Pengujian Radius 3 Km

Dari hasil pengujian radius 3 km pada gambar 4.5 didapatkan 14 data perumahan dengan dengan perhitungan jarak berdasarkan radius 3 Km. warna garis pada pengujian radius 3 Km dibedakan berdasarkan radius levelnya. Radius 1 km dengan warna hijau. Radius 2 km dengan warna merah, Radius 3 km dengan warna biru. Untuk menghitung jarak antar *node* perumahan menggunakan rumus *Euclidean Distance*, berikut ini hasil pencarian radius 3 km.

Vertex M							
Vertex Keluar dari Antrian	Vertex Masuk ke Antrian						
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)						

Gambar 4.6. Hasil Vertex M Radius 3 Km

Pada gambar 4.6 hasil Vertex M radius 3 km adalah hasil penggambaran peta dengan metode *Breadth First Search* menggunakan teknik dimana langkah pertamanya adalah *root node* diekspansi, setelah itu dilanjutkan semua *successor* dari *root node* juga di-*expand*. Hal ini terus dilakukan berulang-ulang hingga *leaf* (*node* pada level paling bawah yang sudah tidak mempunyai *successor* lagi). Untuk menghitung jarak antar *node* perumahan menggunakan rumus *Euclidean Distance*, berikut ini hasil pencarian radius 3 km.

Tabel 4.3. Hasil Perhitungan Radius 3 Km

NODE		Koordinat Node(1) (x,y)		Koordinat Node(2) (x,y)		Jarak KM
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

Metode yang digunakan untuk mengukur jarak pada table 4.3 diatas adalah metode *Euclidean Distance* merupakan suatu metode pencarian kedekatan jarak dari 2 buah variabel, selain mudah metode ini juga lebih efisien waktu, dan proses yang cepat.

4. Pengujian Radius 4 Km.



Gambar 4.7 Hasil Pengujian Radius 4 Km

Dari hasil pengujian radius 4 km pada gambar 4.7 didapatkan 17 data perumahan dengan dengan perhitungan jarak berdasarkan radius 4 Km. warna garis pada Pengujian Radius 4 Km dibedakan berdasarkan radius levelnya. Radius 4 km dengan warna ungu.

Vertex M								
Vertex Keluar dari Antrian	Vertex Masuk ke Antrian							
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)							

Gambar 4.8. Hasil Vertex M Radius 4 Km

Pada gambar 4.8 hasil Vertex M radius 4 km adalah hasil penggambaran peta dengan metode *Breadth First Search* menggunakan teknik dimana langkah pertamanya adalah *root node* diekspansi, setelah itu dilanjutkan semua *successor* dari *root node* juga di-*expand*. Hal ini terus dilakukan berulang-ulang hingga *leaf* (*node* pada level paling bawah yang sudah tidak mempunyai *successor* lagi). Untuk menghitung jarak antar *node* perumahan menggunakan rumus *Euclidean Distance*, berikut ini hasil pencarian radius 4 km.

Tabel 4.4. Hasil Perhitungan Radius 4 Km

NODE		Koordinat Node(1) (x,y)		Koordinat Node(2) (x,y)		Jarak KM
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

Metode yang digunakan untuk mengukur jarak pada table 4.4 diatas adalah metode *Euclidean Distance* merupakan suatu metode pencarian kedekatan jarak dari 2 buah variabel, selain mudah metode ini juga lebih efisien waktu, dan proses yang cepat.

5. Pengujian Radius 5 Km.



Gambar 4.9 Hasil Pengujian Radius 5 Km

Dari hasil pengujian radius 5 km pada gambar 4.9 didapatkan 33 data perumahan dengan dengan perhitungan jarak berdasarkan radius 5 Km. warna garis pada Pengujian Radius 5 Km dibedakan berdasarkan radius levelnya. Radius 5 km dengan warna hitam.

Vertex M								
Vertex Keluar dari Antrian	Vertex Masuk ke Antrian							
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)							

Gambar 4.10 Hasil Vertex M Radius 5 Km

Pada gambar 4.10 hasil Vertex M radius 5 km adalah hasil penggambaran peta dengan metode *Breadth First Search* menggunakan teknik dimana langkah pertamanya adalah *root node* diekspansi, setelah itu dilanjutkan semua *successor* dari *root node* juga di-*expand*. Hal ini terus dilakukan berulang-ulang hingga *leaf (node* pada level paling bawah yang sudah tidak mempunyai *successor* lagi). Untuk menghitung jarak antar *node* perumahan menggunakan rumus *Euclidean Distance*, pada table 4.5 berikut ini merupakan hasil pencarian radius 5 km.

Tabel 4.5 Hasil Perhitungan Radius 5 Km

NODE		Koordinat Node(1) (x,y)		Koordinat Node(2) (x,y)		Jarak KM
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

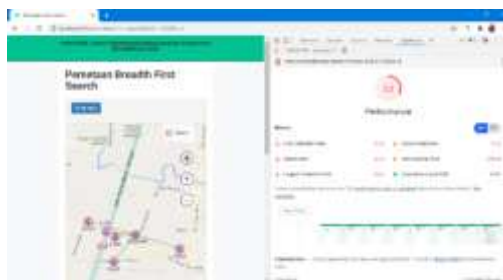
116	117	-7.4291968	112.7069266	-7.4228317	112.705926	0.717
9	24	-7.442978	112.6987763	-7.4463345	112.6923156	0.81
9	28	-7.442978	112.6987763	-7.436595	112.6927383	0.978
95	28	-7.4371607	112.6995163	-7.436595	112.6927383	0.757
111	28	-7.4343586	112.699801	-7.436595	112.6927383	0.824
96	87	-7.4250041	112.7057725	-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. Analisa Hasil pengujian

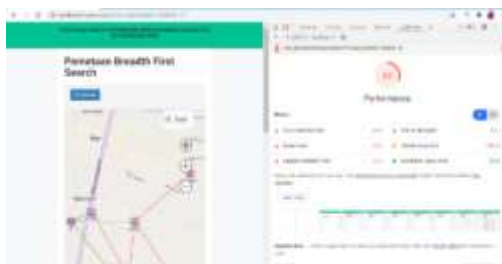
Dari pengujian *Breadth First Search* yang diterapkan dalam aplikasi ini dapat berjalan dengan baik, pengujian dilakukan dengan menguji pencarian berdasarkan radius 1 km, radius 2 km, radius 3 km, radius 4 km, radius 5 km. setiap radius dalam pencarian hasilnya berbeda-beda karena titik lokasi awal pengujian di lima titik lokasi yang berbeda. Berikut ini pada table 4.6 Hasil Pengujian Waktu Proses dengan data yang diampilkkan data perumahan dan proses waktu pencarian.

Tabel 4.6 Hasil Pengujian Waktu Proses

Nama Pengujian	Hasil Perumahan	Waktu Proses
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



Gambar 4.11. Hasil Pengujian Waktu Proses Radius 1 Km



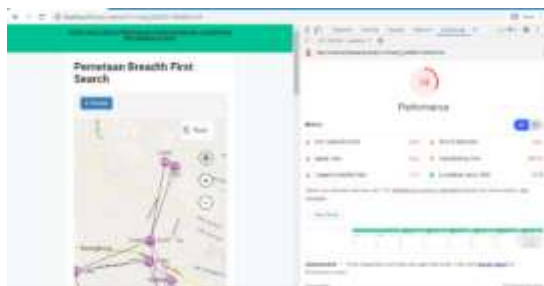
Gambar 4.12. Hasil Pengujian Waktu Proses Radius 2 Km



Gambar 4.13. Hasil Pengujian Waktu Proses Radius 3 Km



Gambar 4.14. Hasil Pengujian Waktu Proses Radius 4 Km



Gambar 4.15. Hasil Pengujian Waktu Proses Radius 5 Km

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa:

1. *Breadth-first search* (BFS) melakukan proses searching pada semua node yang berada pada level atau hirarki yang sama terlebih dahulu sebelum melanjutkan proses searching pada node di level berikutnya.
2. Aplikasi dapat menampilkan garis jalur *Breadth-first search* (BFS) dan *node Breadth-first search* (BFS) yang dilalui sesuai dengan titik awal dan akhir yang telah ditentukan oleh pengguna.
3. Aplikasi dapat mengukur garis jalur dengan menggunakan metode *Euclidean Distance* formula.

5.2. Saran

Setelah melakukan evaluasi terhadap aplikasi secara keseluruhan, diharapkan dari hasil penelitian ini dapat dikembangkan lebih lanjut dengan saran – saran pengembangan sebagai berikut :

1. Untuk pengembangan sistem informasi ini pemetaan lokasi perumahan diperbesar jangkauannya tidak hanya perumahan di Sidoarjo saja.
2. Mengimplementasikan konsep sistem informasi geografis ini pada platform android agar dapat memanfaatkan fitur geolocation atau gps serta meningkatkan mobilitas sistem.
3. Untuk pembuatan node dan jalur Dijkstra, agar dibuat mendekati bentuk lekuk-lekuk jalan, agar selisih jarak yang dihasilkan tidak terlalu jauh dengan jarak yang ditunjukkan oleh Google Maps.

DAFTAR PUSTAKA

- Dharwiyanti, S., Wahono, .S.R., 2003, *Pengantar Unified Modelling Language (UML)*, IlmuComputer.org, terakhir diakses pada 20 Desember 2018.
- Kadir, A., 1999, *Konsep dan Tuntunan Praktis Basis Data*, Andi, Yogyakarta.
- Kusrini, 2007, *Strategi Perancangan dan Pengelolaan Database, Edisi I*, Andi, Yogyakarta.
- 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).
- 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).
- 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).

