

Perbandingan Hasil Rute Terdekat Antar Rumah Sakit di Samarinda Menggunakan Algoritma A*(star) dan Floyd-Warshall

Arief Bramato Wicaksono Putra^[1], Achmad Aulia Rachman^[2], Adji Santoso^[3], Mulyanto^{[4]*}

Program Studi Teknik Informatika Multimedia, Jurusan Teknologi Informasi^{[1],[2],[3],[4]}

Politeknik Negeri Samarinda

Samarinda, Indonesia

email : ariefbram@gmail.com^[1], yanto1294@gmail.com^[4]

Abstract— Transportation of patients between hospitals requires the fastest route, the shortest route is one of the best options. The shortest path search is the search process to the destination node with the optimal weight possible, the weight used is the mileage. The Distance is obtained from the calculation process using the Haversine formula. Floyd-Warshall and A*(star) algorithms are the search algorithms that can be used to find the shortest path. The calculation of the algorithm Floyd-Warshall acquired distance and nodes between Samarinda Medika Citra hospitals to the area Abdul Wahab Sjahranie General Hospital through 12 vertices with a distance of 2.987 kilometers heuristics. From the hospital Dirgahayu headed to the District General Hospital of Abdul Wahab Sjahranie through 11 vertices with a heuristic distance of 3.764 kilometers. The results of using A * (STAR) algorithm obtained A heuristic distance of 4.733 kilometers from the hospital Samarinda Medika Citra to the area Abdul Wahab Sjahranie General Hospital through 15 vertices. From the hospital Dirgahayu to the district general hospital Abdul Wahab Sjahranie through 18 vertices with a heuristic distance of 10.607 kilometers.

Keywords— Search Algorithm, Hospitals, Floyd-Warshall, A*(star), Haversine

Abstrak— Transportasi pasien antar rumah sakit membutuhkan jalur tercepat, rute terdekat merupakan salah satu opsi terbaik. Pencarian jalur terdekat merupakan proses pencarian menuju simpul tujuan dengan bobot seoptimal mungkin, bobot yang digunakan berupa jarak tempuh. Jarak didapat dari proses perhitungan menggunakan rumus Haversine. Algoritma Floyd-Warshall dan A*(star) merupakan algoritma pencarian yang dapat digunakan untuk mencari jalur terdekat. Dari perhitungan menggunakan algoritma Floyd-Warshall didapat jarak dan simpul antar rumah sakit Samarinda Medika Citra menuju rumah sakit umum daerah Abdul Wahab Sjahranie melalui 12 simpul dengan jarak heuristik 2,987 kilometer, dan dari rumah sakit Dirgahayu menuju rumah sakit umum daerah Abdul Wahab Sjahranie melalui 11 simpul dengan jarak heuristik 3,764 kilometer. Hasil menggunakan algoritma A*(star) diperoleh jarak heuristik 4,733 kilometer dari rumah sakit Samarinda Medika Citra menuju rumah sakit umum daerah Abdul Wahab Sjahranie melalui 15 simpul, dan dari rumah sakit Dirgahayu menuju rumah sakit umum daerah

Abdul Wahab Sjahranie melalui 18 simpul dengan jarak heuristik 10,607 kilometer.

Kata Kunci— Algoritma Pencarian, Rumah sakit, Floyd-Warshall, A*(star), Haversine

I. PENDAHULUAN

Rumah sakit merupakan institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna yang menyediakan pelayanan rawat inap, rawat jalan dan rawat gawat darurat [1]. Untuk menuju rumah sakit, dibutuhkan jalur terdekat menuju rumah sakit tujuan agar pasien dapat segera ditangani. Pencarian jalur merupakan proses pencarian jalur paling optimal antara dua atau beberapa simpul yang terhubung satu sama lain. Persoalan mencari lintasan terdekat di dalam graf merupakan salah satu persoalan optimasi. Persoalan ini biasanya direpresentasikan dalam bentuk graf. Graf yang digunakan dalam pencarian lintasan terdekat atau shortest path adalah graf berbobot (weighted graph), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot [2]. Bobot pada graf dapat berupa jarak, kemacetan, maupun kepadatan lalu lintas.

Untuk menemukan jarak terdekat, maka hal pertama yang dilakukan adalah membuat peta dari rumah sakit awal menuju rumah sakit tujuan. Peta merupakan gambaran permukaan bumi dengan skala tertentu yang digambar pada permukaan datar [3]. Dari peta yang telah dibuat, kemudian peta tersebut dibaca untuk mendapatkan struktur peta tersebut. Pada proses pembacaan, didapat struktur peta berupa geometri, garis bujur (longitude) dan garis lintang (latitude). Geometri yang didapat pada peta berupa titik (point) dan garis (line).

Tahapan selanjutnya adalah mencari jarak pada jalur-jalur yang terdapat pada peta. Untuk mendapatkan jarak, rumus yang digunakan adalah rumus haversine. Rumus haversine akan menghasilkan jarak antara dua titik berdasarkan input berupa garis bujur (longitude) dan garis lintang (latitude) [4]. Hasil yang didapat berupa jarak untuk masing-masing jalur yang terdapat pada peta.

Jarak yang didapat digunakan sebagai bobot untuk perhitungan menggunakan algoritma Floyd-Warshall dan A*(star).

Algoritma Floyd-Warshall akan mencari rute terdekat dengan melakukan iterasi sebanyak n kali dimana n mengikuti jumlah simpul yang ada. Hasil setiap iterasi yang diperoleh akan digunakan sebagai acuan untuk tahapan iterasi selanjutnya. Iterasi akan terus dilakukan sampai simpul terakhir. Hasil perhitungan adalah rute terdekat dan jarak terpendek dari semua pasangan simpul.

Algoritma A*(star) melakukan pencarian rute terdekat dengan melakukan perhitungan heuristik. Dari heuristik yang diperoleh, kemudian mencari nilai f(n). Nilai f(n) yang diperoleh akan dimasukkan kedalam open list. Tahap selanjutnya adalah mempertimbangkan bobot antar simpul dimana simpul dengan bobot terkecil yang akan dilalui. Simpul dengan bobot terkecil akan dimasukkan kedalam closed list dimana simpul dalam closed list akan dipilih sebagai simpul yang akan dilalui.

II. TEORI PENUNJANG

A. Algoritma Floyd-Warshall

Algoritma Floyd-Warshall merupakan salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan penyelesaian masalah dengan melihat solusi yang akan didapatkan sebagai suatu keputusan yang saling terhubung, dimana solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan memiliki kemungkinan solusi lebih dari satu.

Untuk mencari rute terdekat menggunakan algoritma Floyd-Warshall, langkah pertama adalah membuat matriks berdasarkan graph. Persamaan untuk membuat matriks adalah sebagai berikut [5] :

$$w_{ij} \begin{cases} 0 & \text{jika } i = j \\ w_{ij} & \text{jika } i \neq j \text{ dan } i \text{ terhubung dengan } j \\ \infty & \text{jika } i \neq j \text{ dan } i \text{ tidak terhubung dengan } j \end{cases} \quad (1)$$

Output berupa matriks jarak $W = [w_{ij}]$ berdimensi $n \times n$

Langkah selanjutnya adalah membuat matriks Z yang berfungsi untuk menentukan simpul-simpul yang harus dilalui untuk menuju simpul tujuan. Persamaan untuk membuat matriks Z adalah sebagai berikut [5] :

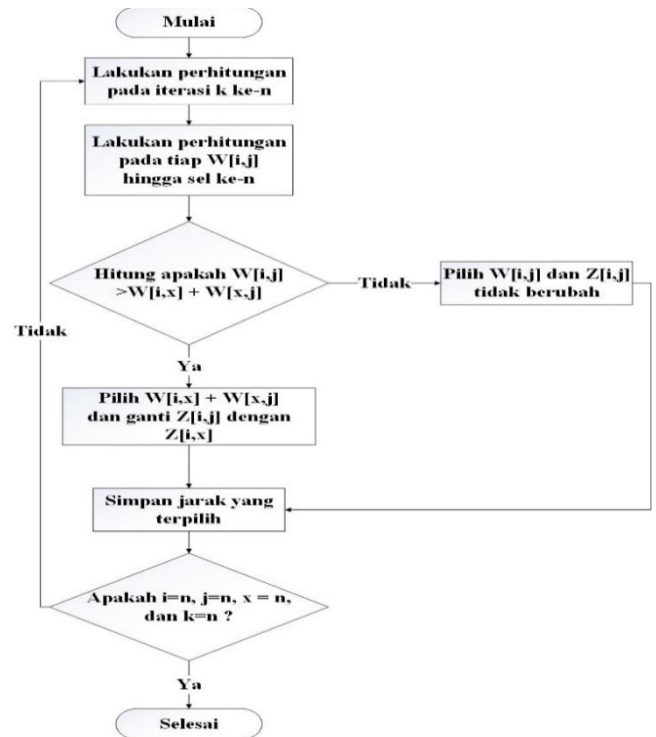
$$\text{Inisialisasi } Z_{ij}^{(0)} = \begin{cases} j & \text{jika } W_{ij}^{(0)} \neq \infty \\ 0 & \text{jika } W_{ij}^{(0)} = \infty \end{cases} \quad (2)$$

Kemudian, dilakukan iterasi pada matriks dengan menggunakan persamaan berikut [5] :

$$d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)}, d_{ix}^{(k-1)} + d_{xj}^{(k-1)}\} \quad (3)$$

Pada algoritma Floyd-Warshall, iterasi dilakukan mulai

dari iterasi pertama sampai n dimana n mengikuti jumlah simpul yang ada. Pada setiap iterasi, setiap blok matriks akan dilakukan perbandingan antara jarak $W_{[i,j]}$ dengan jarak $W_{[i,x]} + W_{[x,j]}$. Jika jarak $W_{[i,j]}$ lebih besar dari jarak $W_{[i,x]} + W_{[x,j]}$, maka ganti jarak $W_{[i,j]}$ dengan jarak $W_{[i,x]} + W_{[x,j]}$ dan ganti $Z_{[i,j]}$ dengan $Z_{[i,x]}$. Flowchart untuk algoritma Floyd-Warshall ditunjukkan oleh Gambar 1 berikut :



Gambar 1. Flowchart algoritma Floyd-Warshall

B. Algoritma A*(star)

Algoritma A* (A star) dikenal sebagai salah satu algoritma yang paling sering digunakan untuk pencarian jalur (path finding) dan penerusan grafis (graph traversal), yaitu proses plotting jalur yang paling efisien antar titik, yang disebut dengan nodes. Metode A* adalah metode yang merupakan hasil pengembangan dari metode dasar Best First Search. Metode ini mengevaluasi setiap titik dengan mengombinasikan dengan g(n), nilai untuk mencapai titik n dari titik awal, dan h(n), nilai perkiraan untuk mencapai tujuan dari titik n tersebut. Algoritma ini menggunakan fungsi distance - plus - cost (biasanya di notasikan dengan f(x)) untuk menentukan urutan kunjungan pencarian node di dalam tree. Gabungan jarak - plus - biaya merupakan penjumlahan dari dua fungsi, yaitu fungsi path - cost (selalu dinotasikan dengan g(x), dimungkinkan bernilai heuristik), dan sebuah kemungkinan penerimaan atas "perkiraan heuristik" jarak ke titik tujuan (dinotasikan dengan h(x)). Fungsi path- cost g(x) adalah jumlah biaya yang harus dikeluarkan dari node awal menuju node tujuan.

Dengan terlebih dahulu mencari rute yang tampaknya mempunyai kemungkinan besar untuk menuju ke arah tujuan, algoritma ini mengambil jarak perjalanan ke arah tujuan (dimana g(x) bagian dari heuristik adalah biaya dari awal).

Beberapa terminologi dasar yang terdapat pada algoritma ini adalah starting point, simpul (nodes), A, open list, closed list, harga (cost), halangan (unwalkable) [6].

Sebelum melakukan pencarian rute, maka langkah pertama adalah menghitung heuristik menggunakan Persamaan 4 berikut [7] :

$$h(n) = \sqrt{(xn - xgoal)^2 + (yn - ygoal)^2} \quad (4)$$

Dimana :

- h(n) = nilai heuristik untuk node /titik n.
- xn = nilai koordinat x dari node /titik n.
- yn = nilai koordinat y dari node /titik n.
- xgoal = nilai koordinat x dari node /titik tujuan.
- ygoal = nilai koordinat y dari node / titik tujuan.

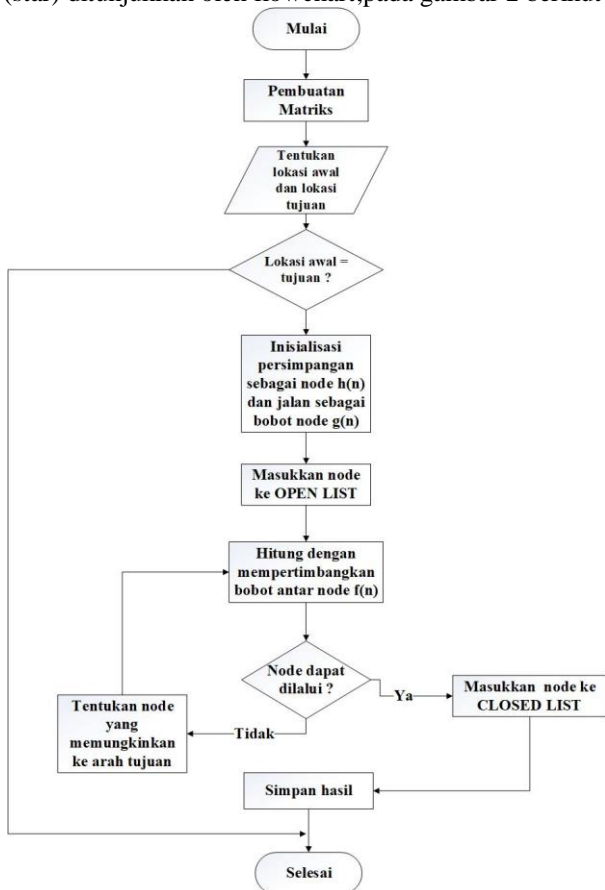
Langkah berikutnya adalah mencari nilai f(n) menggunakan Persamaan 5 berikut [8] :

$$f(n) = h(n) + g(n) \quad (5)$$

Keterangan:

- h (n) = Nilai Heuristik antara Koordinat.
- g (n) = Jarak Koordinat ke titik tujuan.
- f (n) = Perkiraan solusi dengan jarak terpendek melalui n.

Langkah – langkah perhitungan menggunakan algoritma A*(star) ditunjukkan oleh flowchart, pada gambar 2 berikut :



Gambar 2. Flowchart algoritma A*(star)

1) *Pembuatan Matriks*

Tahapan pertama dalam algoritma ini adalah membuat matriks untuk lintasan jalur node – node yang akan dilalui.

2) *Menentukan lokasi awal dan tujuan*

Selanjutnya dalam algoritma ini adalah menentukan lokasi awal dan lokasi tujuan, untuk mengetahui dimana rute awal dan akhir dari jalur tersebut.

3) *Inisialisasi persimpangan*

Jika sudah menentukan lokasi awal dan tujuan, tahapan selanjutnya adalah inisialisasi persimpangan, yaitu dimana setiap simpangan akan dijadikan node h(n) dan jalan antar node sebagai bobot node g(n).

4) *Masukkan node ke open list*

Setelah mendapatkan perkiraan jarak terpendek antar node f(n), maka setiap node yang akan dilalui menuju ke titik tujuan akan dimasukkan ke open list, dan open list berisi node – node yang masih memiliki peluang untuk terpilih sebagai node terbaik.

5) *Hitung dengan mempertimbangkan bobot*

Tahap selanjutnya setelah memasukkan node ke openlist adalah mempertimbangkan bobot antar node mana yang bisa dilalui dan memiliki nilai yang kecil akan dilalui.

6) *Masukkan node ke closed list*

Jika telah mempertimbangkan bobot antar node, node yang memiliki nilai terkecil dapat dilalui akan dimasukkan ke closed list, dan node yang sudah dimasukkan ke closed list tidak bisa di lewati lagi dan peluang untuk terpilih sudah tertutup.

7) *Tentukan node yang memungkinkan ke arah tujuan*

Dalam tahap ini jika node tidak dapat dilalui akan menentukan yang memungkinkan menuju arah tujuan dan kembali menghitung ulang dengan mempertimbangkan bobot antar node.

8) *Simpan hasil*

Jika sudah mendapatkan node – node yang telah dimasukkan ke closed list hingga tujuan, maka kita akan mendapatkan rute terdekat yang akan dilalui.

C. *Formula Haversine*

Formula Haversine adalah persamaan penting dalam sistem navigasi, nantinya Formula Haversine akan menghasilkan jarak terpendek antara dua titik, misalnya pada bola yang diambil dari garis bujur (longtitude) dan garis lintang (latitude). Formula ini pertama kali ditemukan oleh Jamez Andrew di tahun 1805, dan digunakan pertam kali oleh Josef de Mendoza y Ríos di tahun 1801.

Formula Haversine adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Formula Haversine merupakan suatu metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan. Rumus haversine sebagai berikut [9] :

$$\Delta lat = lat2 - lat1 \quad (6)$$

$$\Delta\text{long} = \text{long}2 - \text{long}1 \quad (7)$$

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat}1) \cdot \cos(\text{lat}2) \cdot \sin^2(\Delta\text{long}/2) \quad (8)$$

$$c = 2 \cdot \text{atan}2(\sqrt{a}, \sqrt{1-a}) \quad (9)$$

$$d = R \cdot c \quad (10)$$

Dimana :

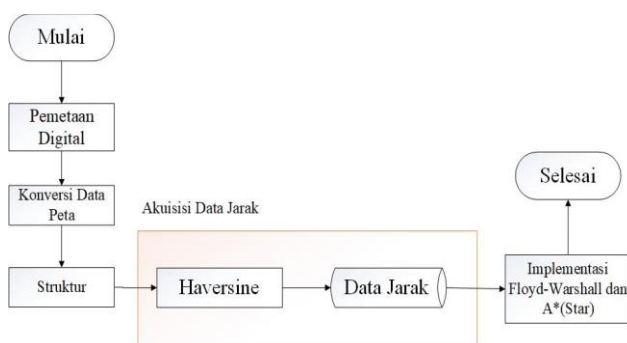
- R = jari-jari bumi sebesar 6371(km)
- Δlat = besaran perubahan latitude
- Δlong = besaran perubahan longitude
- c = kalkulasi perpotongan sumbu
- d = jarak (km)

D. Latitude dan Longitude

Latitude adalah garis yang melintang di antara kutub utara dan kutub selatan, yang menghubungkan antara sisi timur dan barat bagian bumi. Garis ini memiliki posisi membentangi bumi, sama halnya seperti garis equator (khatulistiwa), tetapi dengan kondisi nilai tertentu. Garis lintang inilah yang dijadikan ukuran dalam mengukur sisi utara-selatan koordinat suatu titik di belahan bumi. Sedangkan, longitude adalah garis membujur yang menghubungkan antara sisi utara dan sisi selatan bumi (kutub). Garis bujur ini digunakan untuk mengukur sisi barat-timur koordinat suatu titik di belahan bumi. Sama seperti equator pada latitude yang berada di tengah dan memiliki nilai 0 (no) derajat, pada longitude, garis tengah yang bernilai 0 (no) derajat disebut garis prime meridian (garis bujur). Sedangkan, garis yang berada paling kiri memiliki nilai -90 derajat, dan yang paling kanan memiliki nilai 90 derajat [10].

III. METODOLOGI PENELITIAN

Metodologi penelitian ditunjukkan pada Gambar 3 berikut:



Gambar 3. Metodologi penelitian

A. Pemetaan Digital

Pemetaan digital dilakukan dengan menggunakan Google MyMaps. Langkah yang dilakukan terlebih dahulu adalah menentukan titik awal dan titik tujuan serta persimpangan

yang akan dilalui. Dalam penelitian ini, titik awalnya adalah rumah sakit Samarinda Medika Citra dan rumah sakit Dirgahayu, sedangkan titik tujuannya adalah rumah sakit umum daerah Abdul Wahab Sjahranie. Tiap persimpangan yang dilalui akan diberi tanda berupa titik biru. Langkah selanjutnya adalah membuat jalur yang menyambungkan setiap titik dengan mengikuti jalur yang ada pada peta.

B. Konversi Data Peta

Proses konversi data peta bertujuan untuk mendapatkan struktur peta. Konversi dilakukan dengan mengubah setiap jalur yang telah dibuat menjadi file berformat KML. File-file KML tersebut kemudian dibaca menggunakan tool MATLAB dengan memakai fungsi kml2struct. Hasil pembacaan berupa struktur dari jalur yang telah dibuat.

C. Akuisisi Data Jarak

Pada proses akuisisi data jarak, penelitian ini menggunakan rumus *haversine* yang akan mencari jarak antara garis bujur (*longitude*) dan garis lintang (*latitude*) pada jalur peta menggunakan Persamaan 6 sampai Persamaan 10.

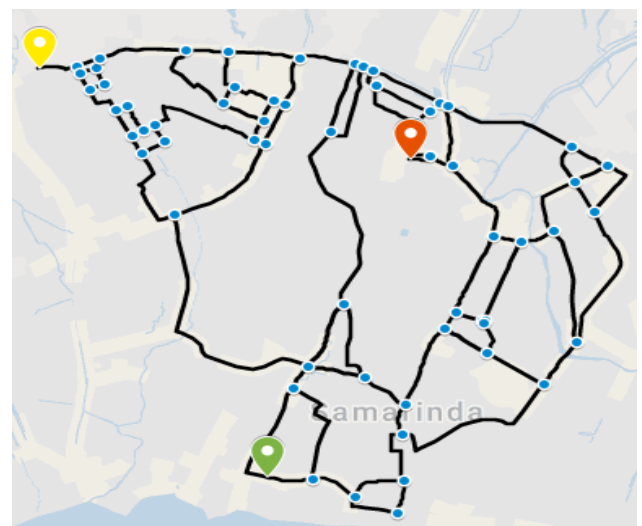
D. Implementasi Floyd-Warshall dan A*(star)

Tahapan terakhir adalah implementasi algoritma Floyd-Warshall dan algoritma A*(star). Hasil dari implementasi kedua algoritma tersebut adalah rute terdekat antar rumah sakit.

IV. HASIL DAN PEMBAHASAN

A. Pemetaan Digital

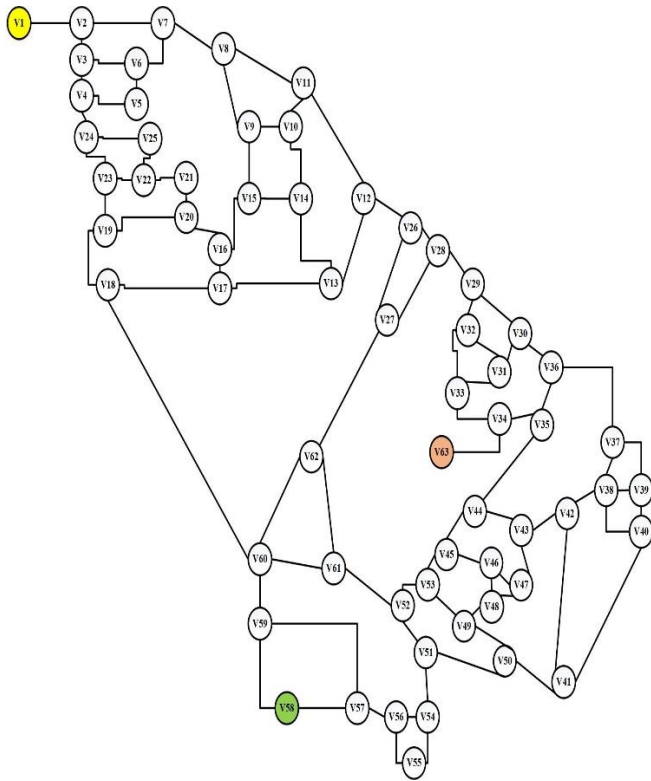
Hasil proses pemetaan digital dapat dilihat pada gambar 4 berikut ini :



Gambar 4. Peta antar rumah sakit

Ikon berwarna kuning merupakan rumah sakit Samarinda Medika Citra, ikon berwarna hijau merupakan rumah sakit Dirgahayu, sedangkan ikon berwarna merah merupakan rumah sakit umum daerah Abdul Wahab Sjahranie. Titik-titik biru merupakan simpangan. Jumlah simpangan berjumlah 63 simpangan dan jumlah jalur adalah 91 jalur. Dari peta yang

dibuat, peta kemudian dibuat menjadi model graph yang ditunjukkan oleh gambar 5.



Gambar 5. Graph peta antar rumah sakit

B. Konversi Data Peta

Dari proses konversi data peta menggunakan fungsi kml2struct, didapat struktur file KML yang dapat dilihat pada model gambar 6 berikut:

Field	Value
Geometry	'Line'
Name	'Path 1'
Description	''
Lon	[117.1248;117.1251;117.1254;117.1258;117.12...
Lat	[-0.4731;-0.4730;-0.4730;-0.4730;-0.4731;-0.4...
BoundingBox	[117.1248,-0.4731;117.1267,-0.4730]

Gambar 6. Struktur file KML

Dari gambar diatas, file KML dengan nama “Path 1” disimpan kedalam variabel dengan nama “p1”. Struktur variabel “p1” berupa geometri, nama, deskripsi, longitude, latitude dan BoundingBox. Setiap jalur akan dibaca dan disimpan kedalam variabel-variabel tersendiri. Jumlah variabel yang tersimpan berjumlah 91 buah sesuai dengan jumlah jalur.

C. Akuisisi data jarak

Pada Gambar 6, variabel “p1” masih belum memiliki jarak. Untuk mendapatkan jarak, rumus haversine digunakan

untuk mencari jarak antar titik koordinat pada jalur. Gambar 7 menunjukkan jalur “Path 1” yang akan dicari jaraknya :



Gambar 7. Jalur “Path 1”

Pada Gambar 7, terlihat bahwa “Path 1” memiliki titik-titik putih berjumlah 8 titik. Penggunaan rumus haversine akan menghasilkan jarak antar titik putih tersebut. Jarak antar titik kemudian dijumlahkan untuk menghasilkan jarak keseluruhan pada suatu jalur. Rumus haversine yang digunakan adalah Persamaan 6 sampai Persamaan 10. Dibawah ini adalah contoh perhitungan untuk mencari jarak pada 2 titik pertama pada jalur “Path 1” :

9) Titik koordinat pertama :

$$\begin{aligned} \text{Longitude 1} &= 117,124771 \\ \text{Latitude 1} &= -0,473112 \\ \text{Radian longitude 1} &= 117,124771 * \pi/180 \\ &= 2,04421289 \\ \text{Radian latitude 1} &= -0,473112 * \pi/180 \\ &= -0,008257362 \end{aligned}$$

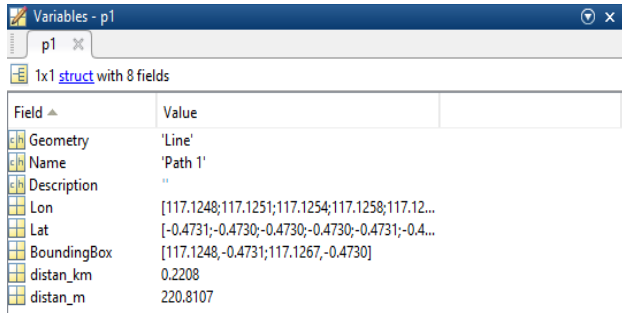
10) Titik koordinat kedua :

$$\begin{aligned} \text{Longitude 2} &= 117,125056 \\ \text{Latitude 2} &= -0,472985 \\ \text{Radian longitude 2} &= 117,125056 * \pi/180 \\ &= 2,044217864 \\ \text{Radian latitude 2} &= -0,472985 * \pi/180 \\ &= -0,008255146 \end{aligned}$$

$$\begin{aligned} 11) \Delta\text{long} &= 2,044217864 - 2,04421289 = 0,000004974 \\ 12) \Delta\text{lat} &= (-0,008255146) - (-0,008257362) = 0,000002217 \\ 13) a &= \sin^2(0,000002217/2) + \cos(-0,008257362) * \cos(-0,008255146) * \sin^2(0,000004974/2) = 0,000000000007414 \\ 14) c &= 2 * \text{atan2}(\sqrt{0,000000000007414}, \sqrt{1-0,000000000007414}) = 0,000005446 \\ 15) d &= 6371 * 0,000005446 = 0,0347 \text{ km} \end{aligned}$$

Dari contoh perhitungannya, didapat jarak sepanjang 0,0347 kilometer untuk 2 titik pertama pada jalur “Path 1”. Cara yang sama digunakan untuk mencari jarak antar titik yang lainnya. Jarak antar titik kemudian dijumlahkan sehingga didapat jarak pada jalur “Path 1” sepanjang 0,2208 kilometer.

Rumus haversine akan dimasukkan kedalam fungsi kml2struct. Pada saat fungsi dijalankan, terdapat 2 buah field baru yang berisikan jarak dalam satuan kilometer dan meter. Gambar 8 menunjukkan hasil pencarian jarak menggunakan fungsi kml2struct yang telah dimasukkan rumus haversine :



Gambar 8. Hasil pencarian jarak

Pada Gambar 8, dapat dilihat bahwa jalur “Path 1” memiliki jarak 0,2208 kilometer. Untuk jarak pada jalur – jalur lainnya, dapat dilihat pada Tabel 1 berikut :

TABLE 1. JARAK SETIAP JALUR

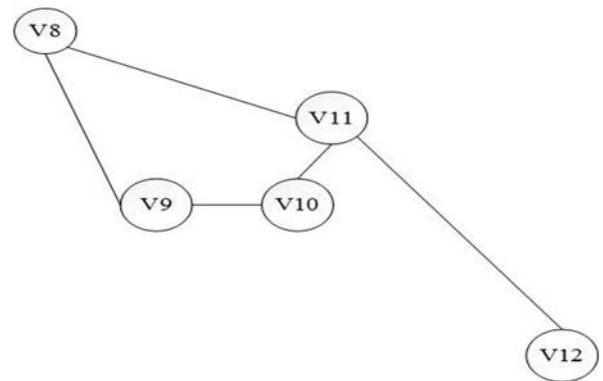
Jalur	Simpangan	Jarak (KM)
Path 1	V1 - V2	0,22
Path 2	V2 - V3	0,05
Path 3	V3 - V4	0,13
Path 4	V4 - V5	0,10
Path 5	V5 - V6	0,13
Path 6	V3 - V6	0,10
Path 7	V6 - V7	0,09
Path 8	V2 - V7	0,15
Path 9	V4 - V24	0,26
Path 10	V23 - V24	0,21
Path 11	V22 - V23	0,08
Path 12	V22 - V25	0,20
Path 13	V24 - V25	0,08
Path 14	V19 - V23	0,14
Path 15	V19 - V20	0,16
Path 16	V20 - V21	0,13
Path 17	V21 - V22	0,08
Path 18	V18 - V19	0,64
Path 19	V17 - V18	0,78
Path 20	V16 - V21	0,78
Path 21	V16 - V17	0,07
Path 22	V15 - V16	0,15
Path 23	V9 - V15	0,26
Path 46	V58 - V59	0,77
Path 47	V57 - V58	0,27
Path 48	V57 - V59	0,81
Path 49	V59 - V60	0,18
Path 50	V60 - V61	0,34
Path 51	V18 - V60	1,74
Path 52	V60 - V62	0,50
Path 53	V27 - V62	1,38
Path 54	V26 - V27	0,50
Path 55	V27 - V28	0,61
Path 56	V56 - V57	0,30
Path 57	V54 - V56	0,35
Path 58	V55 - V56	0,33
Path 59	V54 - V55	0,24
Path 60	V51 - V54	0,43
Path 61	V51 - V52	0,21
Path 62	V50 - V51	1,15
Path 63	V52 - V61	0,31
Path 64	V52 - V53	0,59
Path 65	V49 - V53	0,30
Path 66	V45 - V53	0,13
Path 67	V49 - V50	0,41
Path 68	V48 - V49	0,24

Jalur	Simpangan	Jarak (KM)
Path 24	V9 - V10	0,13
Path 25	V10 - V14	0,27
Path 26	V14 - V15	0,16
Path 27	V8 - V9	0,57
Path 28	V7 - V8	0,52
Path 29	V8 - V11	0,25
Path 30	V10 - V11	0,28
Path 31	V11 - V12	0,42
Path 32	V12 - V13	0,35
Path 33	V13 - V14	0,08
Path 34	V13 - V17	0,29
Path 35	V12 - V26	0,33
Path 36	V26 - V28	0,06
Path 37	V28 - V29	0,06
Path 38	V29 - V30	0,46
Path 39	V29 - V32	0,13
Path 40	V31 - V32	0,37
Path 41	V32 - V33	0,39
Path 42	V30 - V31	0,09
Path 43	V31 - V33	0,11
Path 44	V33 - V34	0,31
Path 45	V34 - V63	0,15
Path 69	V47 - V48	0,02
Path 70	V46 - V48	0,03
Path 71	V46 - V47	0,02
Path 72	V45 - V46	0,16
Path 73	V44 - V45	0,58
Path 74	V35 - V44	0,60
Path 75	V34 - V35	0,15
Path 76	V35 - V36	0,43
Path 77	V30 - V36	0,05
Path 78	V36 - V37	0,81
Path 79	V37 - V39	0,25
Path 80	V37 - V38	0,27
Path 81	V38 - V39	0,22
Path 82	V39 - V40	0,46
Path 83	V38 - V40	0,25
Path 84	V40 - V41	1,03
Path 85	V41 - V50	0,35
Path 86	V41 - V42	0,82
Path 87	V38 - V42	0,46
Path 88	V42 - V43	0,21
Path 89	V43 - V44	0,17
Path 90	V43 - V47	0,59
Path 91	V61 - V62	0,60

D. Implementasi

1) Algoritma Floyd-Warshall

Jarak yang diperoleh akan digunakan sebagai bobot untuk mencari jarak terpendek. Karena banyaknya jumlah simpul pada graph peta, maka diambil 5 buah simpul untuk dijadikan sebagai contoh perhitungan. Simpul-simpul yang digunakan sebagai contoh adalah simpul V1, V2, V3, V4 dan V5. Yang ditunjukkan pada gambar 9 berikut :



Gambar 9. Graph 5 simpul

Dari graph contoh diatas, kemudian dibuat kedalam bentuk matriks menggunakan Persamaan (1) :

TABLE II. MATRIKS JARAK AWAL

	V8	V9	V10	V11	V12
V8	0	0,57	∞	0,25	∞
V9	0,57	0	0,13	∞	∞
V10	∞	0,13	0	0,28	∞
V11	0,25	∞	0,28	0	0,42
V12	∞	∞	∞	0,42	0

Tabel 2 dapat dijelaskan seperti berikut : blok berwarna kuning merupakan titik awal (i) sedangkan blok berwarna biru merupakan titik tujuan [2]. Selanjutnya adalah membuat matriks Z menggunakan Persamaan 2. Fungsi matriks Z adalah untuk mencari simpul-simpul yang harus dilalui ditunjukkan pada tabel 3:

TABLE III. MATRIKS Z AWAL

	V8	V9	V10	V11	V12
V8	8	9	0	11	0
V9	8	9	10	0	0
V10	0	9	10	11	0
V11	8	0	10	11	12
V12	0	0	0	11	12

Tahapan selanjutnya adalah melakukan iterasi menggunakan Persamaan 3. Proses iterasi dilakukan dengan membandingkan antara jarak simpul awal menuju simpul tujuan secara langsung, dengan jarak simpul awal menuju simpul tujuan melalui titik perantara. Setiap simpul pada graph akan digunakan sebagai simpul perantara (x). proses iterasi dari awal hingga akhir ditunjukkan oleh tabel 4 sampai tabel 7. Untuk iterasi pertama, simpul perantaranya adalah simpul V1.

TABLE IV. HASIL PERHITUNGAN JARAK ITERASI PERTAMA

	V8	V9	V10	V11	V12
V8	0	0,57	∞	0,25	∞
V9	0,57	0	0,13	0,82	∞
V10	∞	0,13	0	0,28	∞
V11	0,25	0,82	0,28	0	0,42
V12	∞	∞	∞	0,42	0

TABLE V. HASIL PERHITUNGAN JALUR ITERASI PERTAMA

	V8	V9	V10	V11	V12
V8	8	9	0	11	0
V9	8	9	10	8	0
V10	0	9	10	11	0
V11	8	8	10	11	12
V12	0	0	0	11	12

Dari proses iterasi pertama, didapat jarak heuristik dari simpul V9 menuju simpul V11 yaitu 0,82 kilometer. Namun, masih terdapat simpul yang jaraknya belum ditemukan. Untuk itu, proses iterasi dilanjutkan ke iterasi kedua sampai iterasi terakhir dengan simpul perantaranya adalah simpul V9 dan seterusnya.

TABLE VI. HASIL PERHITUNGAN JARAK ITERASI KELIMA

	V8	V9	V10	V11	V12
V8	0	0,57	0,53	0,25	0,67
V9	0,57	0	0,13	0,41	0,83
V10	0,53	0,13	0	0,28	0,70
V11	0,25	0,41	0,28	0	0,42
V12	0,67	0,83	0,70	0,42	0

TABLE VII. HASIL PERHITUNGAN JALUR ITERASI KELIMA

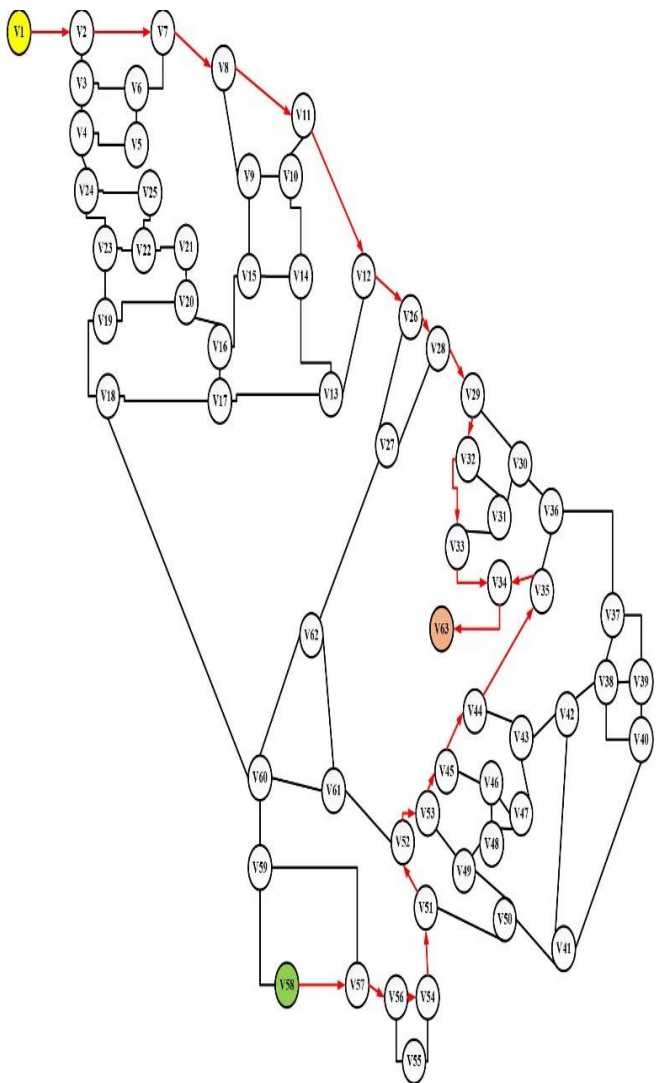
	V8	V9	V10	V11	V12
V8	8	9	11	11	11
V9	8	9	10	10	10
V10	11	9	10	11	11
V11	8	10	10	11	12
V12	11	11	11	11	12

Dari proses iterasi sebanyak 5 kali, didapat jalur paling optimal untuk semua jalur beserta simpul-simpul yang harus dilalui. Sebagai contoh, jarak heuristik dari simpul V8 menuju simpul V12 adalah 0,67 kilometer dengan simpul yang dilalui adalah V8 -> V11 -> V12.

Pada kasus sebenarnya, akan dicari rute terdekat dari simpul V1 menuju simpul V63 dan dari simpul V58 menuju simpul V63. Dari uraian penjelasan sebelumnya, graph pada Gambar 3 akan diubah kedalam bentuk matriks menggunakan Persamaan 1. Matriks yang dihasilkan berukuran 63 x 63. Kemudian, dibuat matriks Z menggunakan Persamaan 3. Sama seperti pembahasan sebelumnya, kemudian dilakukan proses iterasi sebanyak simpul yang ada. Pada kasus sebenarnya, iterasi dilakukan sebanyak 63 kali. Dari 63 kali proses iterasi, didapat rute terdekat antar rumah sakit sebagai berikut :

- 16) Rute terdekat dari rumah sakit Samarinda Medika Citra (V1) menuju rumah sakit umum daerah Abdul Wahab Sjahranie (V63) adalah V1 – V2 – V7 – V8 – V11 – V12 – V26 – V28 – V29 – V32 – V33 – V34 – V63 dengan jarak heuristik 2,987 kilometer.
- 17) Rute terdekat dari rumah sakit Dirgahayu (V58) menuju rumah sakit umum daerah Abdul Wahab Sjahranie (V63) adalah V58 – V57 – V56 – V54 – V51 – V52 – V53 – V45 – V44 – V35 – V34 – V63 dengan jarak heuristik 3,764 kilometer.

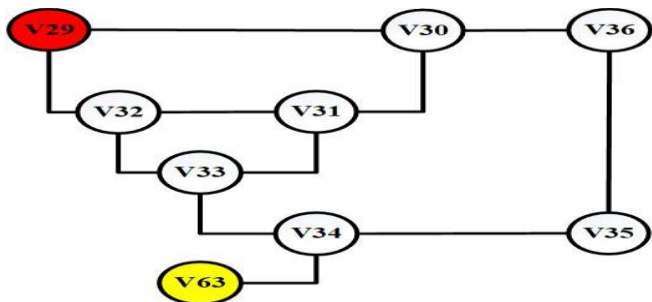
Jalur antar rumah sakit menggunakan algoritma Floyd-Warshall ditunjukkan oleh Gambar 10 berikut :



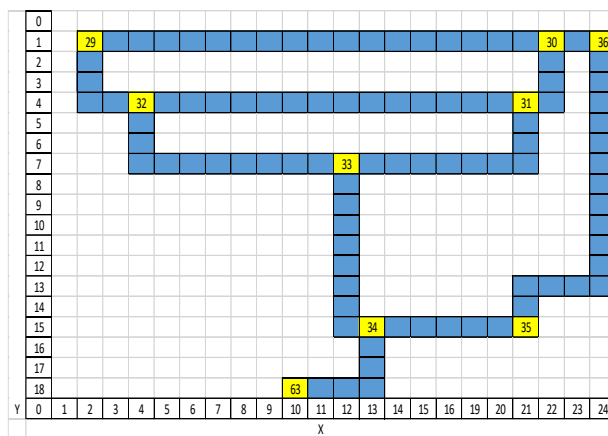
Gambar 10. Hasil pencarian Floyd-Warshall

2) *Algoritma A*(star)*

Sebagai contoh perhitungan menggunakan algoritma A*(star), graph pada Gambar 5 yang berupa graph peta rumah sakit, disederhanakan dengan hanya menggunakan 9 simpul yaitu V29, V30, V31, V32, V33, V34, V35, V36, dan V63. Kesembilan simpul tersebut kemudian diubah kedalam bentuk matriks dengan satu matriks bernilai jarak 25 meter, yang ditunjukkan pada gambar 11 dan selanjutnya dirubah dalam bentuk matrik yang ditunjukkan pada gambar 12 berikut :



Gambar 11. Graph 9 simpul



Gambar 12. Matriks 9 simpul

Dari matriks yang dibuat sesuai dengan gambar 12, langkah selanjutnya adalah menentukan posisi awal dan akhir node serta menunjukkan posisi node pada matriks yang ditunjukkan tabel 8 berikut :

TABLE VIII. POSISI NODE PADA MATRIKS

Node	X	Y
29	2	1
30	22	1
31	21	4
32	4	4
33	12	7
34	13	15
35	21	16
36	24	1
63	10	18

Langkah selanjutnya adalah mencari nilai heuristik menggunakan Persamaan 4, dapat dilihat pada tabel 9 :

TABLE IX. HASIL PERHITUNGAN HEURISTIK

Jalur	H(n)
V29 - V30	20,00
V29 - V32	3,61
V31 - V32	17,00
V32 - V33	8,54
V30 - V31	3,16
V31 - V33	9,49
V33 - V34	8,06
V34 - V63	4,24
V34 - V35	8,06
V35 - V36	15,30
V30 - V36	2,00

Setelah nilai heuristik diperoleh, maka selanjutnya adalah melakukan perhitungan nilai $f(n)$ menggunakan Persamaan 5. Adapun langkah – langkah untuk menemukan nilai $f(n)$, yaitu :

- 18) Langkah pertama adalah buat tabel open node dan closed node.
- 19) Menghitung nilai $f(n)$ dengan menjumlahkan nilai heuristik $h(n)$, dengan jarak $g(n)$.
- 20) Memasukkan nilai $f(n)$ masing masing node yang mempunyai hubungan atau ketetanggaan dengan titik awal ke dalam open node
- 21) Melakukan cek nilai $f(n)$ dan memasukkan nilai $f(n)$ terkecil kedalam closed node.

Lakukan berulang ulang sampai titik goal tercapai. Hasilnya ditunjukkan pada tabel 10 berikut :

TABLE X. HASIL PERHITUNGAN ALGORITMA A-STAR

Node	X	Y	Jalur	H(n)	G(n)	F(n)
29	2	1	V29-V30	20,00	18	38,00
30	22	1	V29 - V32	3,61	5	8,61
31	21	4	V31 - V32	17,00	15	32,00
32	4	4	V32 - V33	8,54	11	19,54
33	12	7	V30 - V31	3,16	4	7,16
34	13	15	V31 - V33	9,49	10	19,49
35	21	16	V33 - V34	8,06	9	17,06
36	24	1	V34 - V63	4,24	6	10,24
63	10	18	V34 - V35	8,06	6	14,06
			V35 - V36	15,30	17	32,30
			V30 - V36	2,00	2	4,00

Pada tabel 11 dapat dijelaskan prosesnya sebagai berikut : setelah melakukan perhitungan untuk menemukan nilai $f(n)$, langkah selanjutnya adalah melakukan cek nilai $f(n)$ dan memasukkan nilai $f(n)$ terkecil kedalam closed node dan dilakukan berulang ulang sampai titik goal tercapai.

TABLE XI. PERHITUNGAN OPEN NODE

Open node		
tujuan	f(n)	dari
30	38,00	29
31	32,00	32
31	19,49	33
35	14,06	34

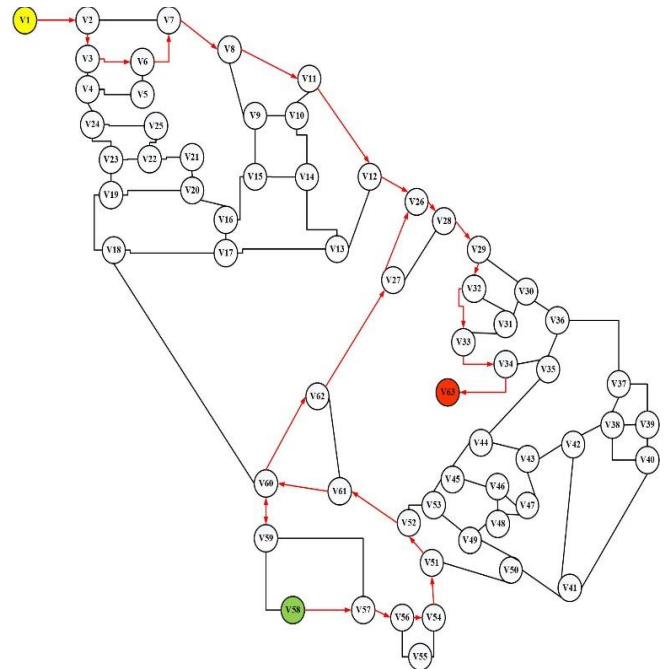
TABLE XII. PERHITUNGAN CLOSED NODE

Closed node		
tujuan	f(n)	Dari
32	8,61	29
33	19,54	32

Closed node		
34	17,06	33
63	10,24	34
Total	55,45	

Dari Tabel 12, didapat rute terdekat dari simpul V29 menuju V63 adalah V29 – V32 – V33 – V34 – V63 dengan total perkiraan solusi jarak heuristik terpendek adalah 55,45 x 25 meter = 1,386 kilometer.

Pada kasus sebenarnya, graph pada Gambar 5 akan diubah kedalam bentuk matriks seperti dan menghasilkan pencarian seperti pada Gambar 13 berikut.



Gambar 13. Hasil pencarian A*(star)

Gambar 13 dapat diuraikan seperti berikut : diawali menentukan posisi awal dan akhir node. Kemudian mencari nilai heuristik menggunakan Persamaan 4. Dari nilai heuristik yang diperoleh, kemudian dicari nilai $f(n)$ menggunakan Persamaan 5. Setelah nilai $f(n)$ ditemukan, lalu langkah selanjutnya adalah melakukan cek nilai $f(n)$ dan memasukkan nilai $f(n)$ terkecil kedalam closed node dan dilakukan berulang ulang sampai titik goal tercapai.

Dari perhitungan menggunakan algoritma A*(star), diperoleh rute terdekat dari rumah sakit Samarinda Medika Citra menuju rumah sakit umum daerah Abdul Wahab Sjahranie adalah V1 – V2 – V3 – V6 – V7 – V8 – V11 – V12 – V26 – V28 – V29 – V32 – V33 – V34 – V63 dengan perkiraan jarak heuristik terpendek 4,733 kilometer, dan rute terdekat dari rumah sakit Dirgahayu menuju rumah sakit umum daerah Abdul Wahab Sjahranie adalah V58 – V57 – V56 – V54 – V51 – V52 – V61 – V60 – V59 – V62 – V27 – V26 – V28 – V29 – V32 – V33 – V34 – V63 dengan perkiraan jarak heuristik terpendek 10,607 kilometer.

V. KESIMPULAN

Dari hasil penelitian yang dilakukan, dapat disimpulkan bahwa algoritma Floyd-Warshall dan algoritma A*(star) dapat digunakan untuk mencari rute terdekat antar rumah sakit. Dengan menggunakan algoritma Floyd-Warshall, diperoleh rute terdekat dari rumah sakit Samarinda Medika Citra (V1) menuju rumah sakit Abdul Wahab Sjahranie (V63) melalui 12 simpul dengan jarak heuristik 2,987 kilometer, dan dari rumah sakit Dirgahayu (V58) menuju rumah sakit Abdul Wahab Sjahranie (V63) melalui 11 simpul dengan jarak heuristik 3,764 kilometer.

Dengan menggunakan algoritma A*(star), diperoleh rute terdekat dari rumah sakit Samarinda Medika Citra (V1) menuju rumah sakit umum daerah Abdul Wahab Sjahranie (V63) melalui 15 simpul dengan perkiraan jarak heuristik terpendek 4,733 kilometer, dan rute terdekat dari rumah sakit Dirgahayu (V58) menuju rumah sakit umum daerah Abdul Wahab Sjahranie (V63) melalui 18 simpul dengan perkiraan jarak heuristik terpendek 10,607 kilometer.

Dari perhitungan menggunakan algoritma Floyd-Warshall dan algoritma A*(star), dapat disimpulkan bahwa algoritma Floyd-Warshall bekerja lebih efektif dalam melakukan pencarian rute terdekat antar rumah sakit dengan selisih 3 simpul untuk rute dari rumah sakit Samarinda Medika Citra (V1) menuju rumah sakit umum daerah Abdul Wahab Sjahranie (V63), dan selisih 7 simpul untuk rute dari rumah sakit Dirgahayu (V58) menuju rumah sakit umum daerah Abdul Wahab Sjahranie (V63).

REFERENCES

- [1] S. E. Rikomah, *Farmasi Rumah Sakit*: Deepublish, 2017.
- [2] A. Juniansyah And Mesterjon, "Aplikasi Penentuan Rute Terpendek Untuk Bagian Pemasaran Produk Roti Surya Dengan Metode Best First Search," *Jurnal Media Infotama* Vol. 12 No. 1, P. 10, 2016.
- [3] T. S. Nusantara, *Top No.1 Sukses Pendalaman Materi Smp/Mts Kelas Vii*: Gramedia Widiasarana, 2016.
- [4] D. M. Khairina, F. W. Ramadhinata, And H. R. Hatta, "Pencarian Lokasi Jalur Nugraha Ekakurir (Jne) Terdekat Menggunakan Haversine Formula (Studi Kasus Kota Samarinda)," *Seminar Nasional Inovasi Dan Aplikasi Teknologi Di Industri*, P. 5, 2017.
- [5] F. W. Ningrum And T. Andrasto, "Penerapan Algoritma Floyd-Warshall Dalam Menentukan Rute Terpendek Pada Pemodelan Jaringan Pariwisata Di Kota Semarang," *Jurnal Teknik Elektro*, Vol. Volume 8 No. 1, P. 4, 2016.
- [6] Y. Syukriyah, Falahah, And H. Solihin, "Penerapan Algoritma A* (Star) Untuk Mencari Rute Tercepat Dengan Hambatan," *Seminar Nasional Telekomunikasi Dan Informatika (Selisik 2016)*, P. 6, 2016.
- [7] M. Yamin And M. B. Talai, "Aplikasi Pencarian Jalur Terpendek Pada Rumah Sakit Umum Bahteramas Menggunakan Algoritma A* (A-Star)" *Jurnal Informatika*, Vol. Volume 9 No. 2, P. 14, 2015.
- [8] P. Purnamasari And W. Suyitno, "Pencarian Jalur Terpendek Dari Rumah Menuju Candi Jiwa Batujaya Menggunakan Algoritma A-Star," *Knit-2 Nusa Mandiri* P. 6, 2016.
- [9] Yulianto, Ramadiani, And A. H. Kridalaksana, "Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal," *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer* Vol. 13, 2018.
- [10] E. Kusuma Jefri And H. Agung, "Aplikasi Perhitungan Dan Visualisasi Jarak Terpendek Berdasarkan Data Coordinate Dengan Algoritma Dijkstra Dalam Kasus Pengantaran Barang Di Kawasan Jabodetabek," *Jurnal Sisfokom*, Vol. Volume 8 Nomor 1, P. 10, 2019.