

ANALISA PERBANDINGAN ALGORITMA FLOYD-WARSHALL DAN ALGORITMA DIJKSTRA UNTUK PENENTUAN RUTE TERDEKAT

Ahmada Maghfirotul Inayah¹, Nalsa Cintya Resti^{2*}, Nur Fadilatul Ilmiyah³

IAIN Kediri^{1,2,3}

nalsacintya@iainkediri.ac.id

Received: 8 April 2023

Accepted: 23 Juni 2023

Published : 8 Desember 2023

Abstract

The aims of this study is to find a solution for applying the Floyd-Warshall Algorithm and Dijkstra's Algorithm in determining the shortest route among hospitals in Kediri City. The steps of the research include 1) making a weighted graph of inter-hospital road routes in Kediri City; 2) explaining the completion of the Implementation of the Floyd-Warshall Algorithm and Dijkstra's Algorithm; 3) comparing between the Floyd-Warshall Algorithm and Dijkstra's Algorithm based on the results and processing. Based on the research results, there are 9 points of public hospitals in Kediri City and the results are the same between the two algorithms in the form of 72 shortest routes and their trajectories. These two algorithms have the shortest distance and the same route. Even though, each algorithms has characteristics and different ways of calculating including principles, methods, speed, complexity and results as well as effectiveness. In this study, the Floyd-Warshall Algorithm is more effective than Dijkstra's Algorithm for determining the shortest route among hospitals in Kediri City.

Keywords: *dijkstra's algorithm, floyd-warshall algorithm, nearest route*

Abstrak

Penelitian ini bertujuan untuk menemukan penyelesaian dari penerapan Algoritma *Floyd-Warshall* dan Algoritma *Dijkstra* dalam menentukan rute terpendek antar- rumah sakit di Kota Kediri satu sama lain. Langkah dari penelitian meliputi 1) membuat graf berbobot rute jalan antar-rumah sakit di Kota Kediri; 2) menjelaskan penyelesaian dari Implementasi Algoritma *Floyd-Warshall* dan Algoritma *Dijkstra*; 3) membandingkan antara Algoritma *Floyd-Warshall* dan Algoritma *Dijkstra* berdasarkan hasil dan proses pengerjaan. Berdasarkan hasil penelitian terdapat 9 titik rumah sakit umum di Kota Kediri dan diperoleh hasil yang sama antara kedua algoritma berupa 72 rute terpendek beserta lintasannya. Kedua algoritma ini memiliki hasil jarak terdekat dan rute tempuh yang sama. Meskipun begitu, masing-masing algoritma memiliki karakteristik bahkan cara penghitungan yang berbeda meliputi prinsip, metode, kecepatan, kerumitan dan hasil serta keefektifan. Dalam penelitian ini, Algoritma *Floyd-Warshall* lebih efektif dibandingkan dengan Algoritma *Dijkstra* untuk menentukan rute terdekat antar-Rumah Sakit di Kota Kediri.

Kata Kunci: *algoritma dijkstra, algoritma floyd-warshall, rute terdekat*

Sitasi artikel ini:

Inayah A., M., Resti, N. C., & Ilmiyah, N. F. (2023). Analisa Perbandingan Algoritma Floyd-Warshall dan Algoritma Dijkstra untuk Penentuan Rute Terdekat. *Jurnal Ilmiah Matematika Realistik*, Vol(1), 146-155.

PENDAHULUAN

Rumah sakit adalah salah satu layanan kesehatan yang sangat penting peranannya bagi masyarakat. Salah satu layanan yang disediakan rumah sakit untuk menunjang kebutuhan pasien adalah mobil ambulans. Mobil ambulans adalah suatu kendaraan yang digunakan untuk mengantar atau memindahkan pasien ke suatu tempat untuk mendapatkan pengobatan. Menurut (Peraturan Menteri Kesehatan Republik Indonesia Nomor 11 Tahun 2017. Tentang Keselamatan Pasien Rumah Sakit, 2017) dalam Peraturan Menteri Kesehatan RI tentang keselamatan pasien rumah sakit, keselamatan pasien adalah prioritas utama yang harus diprogramkan di rumah sakit. Kondisi pasien dengan berbagai macam keluhan mengharuskan para tenaga medis untuk selalu cepat dan

cekatan dalam hal penanganan. Penanganan yang penting untuk dilakukan salah satunya adalah membawa pasien ke rumah sakit terdekat. Tetapi, terdapat kendala bahwa terkadang rumah sakit yang dituju tidak memiliki kamar kosong atau tidak melayani pertolongan medis yang diinginkan. Untuk mendapatkan pelayanan yang diinginkan, tenaga medis perlu terampil dalam menentukan jarak terpendek untuk pasien transfer rumah sakit agar segera mendapatkan perawatan.

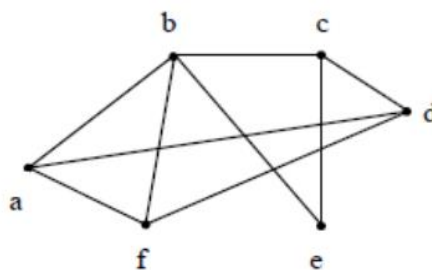
Dalam matematika, terdapat ilmu yang mempelajari tentang cara mencari rute tercepat dari suatu lokasi menggunakan graf. Pencarian jalur terpendek dalam penelitian ini menggunakan graf berarah dan berbobot yang mana setiap titiknya dihubungkan oleh sisi yang memiliki arah dan nilai bobot (jarak tiap-tiap titik pada graf). Terdapat beberapa contoh algoritma untuk mencari jalur terpendek dari suatu titik ke titik lain. Algoritma tersebut ialah Algoritma *Floyd-Warshall* dan Algoritma *Dijkstra* yang berfungsi untuk menentukan jalur terpendek dari suatu titik ke titik lain secara maksimal.

Algoritma *Dijkstra* merupakan salah satu metode yang dapat dioperasikan secara menyeluruh terhadap titik-titiknya dengan asumsi bobotnya non-negatif. Kasus jalur terpendek dapat diselesaikan oleh Algoritma *Dijkstra* antara lain jalur terpendek antara simpul tertentu ke semua titik yang lain, jalur terpendek antara dua buah titik, jalur terpendek antara semua pasangan titik, dan jalur terpendek antara dua buah titik melalui beberapa titik tertentu (Yusuf et al., 2017). Algoritma ini memiliki cara kerja dengan mencoba memilih nilai optimum lokal untuk setiap langkah secara umum dan mengarah pada nilai keseluruhan. Selanjutnya, menurut (Ningrum & Andrasto, 2016) terdapat Algoritma *Floyd-Warshall* yang memiliki cara kerja dengan menghitung setiap jalur yang menghubungkan pasangan titik secara sekaligus sampai titik tujuan dengan jumlah bobot yang paling minimum. Algoritma ini sederhana dan mudah diimplementasikan, akan tetapi proses penghitungannya lambat terutama untuk nilai bobot yang besar. Dari kedua penggunaan algoritma tersebut terdapat perbedaan mendasar yang mana *Dijkstra* memakai prinsip *greedy* atau mencari solusi terbaik setiap langkah dari titik terakhir yang sudah dipilih ke titik yang belum terpilih, sedangkan *Floyd-Warshall* memakai metode pencarian solusi terbaik rute terdekat antar keseluruhan titik satu sama lain secara terkait. Telah ada beberapa penelitian yang membahas topik mengenai Algoritma *Dijkstra* dan Algoritma *Floyd Warshall*, salah satunya adalah (Attamimi et al., 2017) membahas tentang penentuan jalur terpendek pada jaringan *openflow* menggunakan perbandingan Algoritma *Floyd-Warshall* dan *Dijkstra*. Hasil penelitian menunjukkan kedua algoritma tersebut tidak memiliki perbedaan yang signifikan satu sama lain. (Ningrum & Andrasto, 2016) meneliti tentang penentuan rute terpendek dengan menggunakan Algoritma *Floyd-Warshall* pada pemodelan jaringan pariwisata di Kota Semarang. Sebelum mengimplementasikan algoritma *Floyd-Warshall*, para wisatawan domestik belum mengetahui jalur terpendek yang harus dilalui agar dapat mengefisiensi jarak, waktu, dan biaya. Hasil yang diperoleh berupa titik graf yang merepresentasikan jaringan pariwisata Kota Semarang.

Berdasarkan penjelasan di atas peneliti tertarik untuk mempelajari lebih lanjut tentang perbandingan pencarian rute terdekat antar-rumah sakit di Kota Kediri. Oleh sebab itu, untuk efisiensi pelacakan rumah sakit, penelitian ini akan menerapkan perbandingan Algoritma *Dijkstra* dan Algoritma *Floyd-Warshall* sebagai jalan keluar menentukan rute atau jalur evakuasi terbaik bagi pasien darurat.

Graf

Menurut (Siang, n.d.) graf G adalah suatu himpunan yang terdiri dari himpunan titik dan himpunan garis. Elemen dari himpunan (tak kosong) titik disebut titik/*vertex*, sedangkan elemen dari himpunan garis disebut sisi/*edge*. Graf dapat ditulis dengan $G(V, E)$ dengan V adalah titik dan E adalah sisi. Perhatikan contoh berikut (Rahayuningsih, n.d.):



Gambar 1. Graf $G(V, E)$

Terdapat beberapa istilah terkait sebuah graf antara lain: Graf berbobot merupakan sebuah graf yang setiap sisi diberi nilai/bobot. Nilai pada graf berbobot dapat bisa saja berbeda sesuai dengan model permasalahan yang ingin diselesaikan. *Incident* (terkait/bersisian). Apabila a adalah sisi yang memiliki simpul yaitu b dan c yang ditulis $a = (b, c)$. Maka b dan c dinamakan terletak pada a sedangkan a dinamakan incident/bersisian dengan b dan c . *Adjacent* (bertetangga), Jika terdapat 2 simpul pada graf yang tidak berarah, maka dua simpul tersebut disebut *adjacent* apabila terdapat sisi yang menghubungkan keduanya. *Degree*(derajat), *indegree* dan *outdegree*, *Degree*/derajat dari sebuah simpul merupakan banyaknya sisi yang bersisian dengan simpul tersebut. *Indegree* adalah simpul yang memiliki sisi dengan arah masuk/*in* pada simpulnya (terjadi pada graf berarah), *Outdegree* adalah kebalikan dari *indegree* yaitu simpul yang memiliki sisi dengan arah keluar/*out* pada simpulnya (terjadi pada graf berarah)

Algoritma Dijkstra

Menurut (Munir, 2012) Algoritma dapat didefinisikan sebagai urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Menurut (Aulia et al., 2015) Algoritma *Dijkstra* adalah salah satu metode penentuan jalur terpendek yang digunakan untuk graf berbobot dan terarah. Dijkstra adalah nama penemu algoritma ini yaitu *Edsger Wybe Dijkstra*. Algoritma *Dijkstra* dapat menemukan jalur terpendek dalam beberapa langkah. Algoritma ini memakai prinsip serakah/*greedy*, hal ini memiliki arti bahwa pada setiap langkah akan dipilih sisi dengan bobot paling sedikit dan mencatatnya ke dalam himpunan solusi.

Algoritma *Dijkstra* merupakan salah satu algoritma yang paling efektif untuk menyediakan rute terpendek dari satu tempat ke tempat lain. Prinsip dari algoritma *Dijkstra* adalah mencari letak suatu titik dengan mencari dua jalur terpendek. Untuk setiap iterasinya jarak dari titik yang diketahui (dari titik awal) diperbarui jika ditemukan titik baru yang memberikan jarak terpendek. Syarat untuk algoritma ini adalah bahwa bobot sisi dari graf harus memiliki bobot positif (≤ 0) (Satyananda, 2012).

Adapun cara mencari bobot dari tiap-tiap sisi adalah:

$$X[ij] = \begin{cases} 0, & \text{jika } i = j \\ X[ij], & \text{jika } i \neq j \text{ dan } [ij] \in E \\ \infty, & \text{jika } i \neq j \text{ dan } [ij] \notin E \end{cases} \quad (1)$$

$X[ij]$ adalah bobot dari titik pada baris ke- i ke titik yang terdapat pada kolom ke- j dan E adalah edge atau sisi.

Secara umum, algoritma Dijkstra menurut (Wilson, 2010) bekerja sebagai berikut: 1) penentuan titik awal, pemberian bobot jarak pada titik pertama menuju titik terdekat secara satu per satu, kemudian dilakukan pengembangan; 2) pemberian nilai bobot (jarak) dari setiap titik ke titik lainnya, lalu beri nilai 0 pada titik awal dan nilai tak hingga terhadap titik lain yang belum terisi; 3) atur semua titik yang belum dilalui dan atur titik awal sebagai "Titik keberangkatan"; 4) dari titik keberangkatan, pertimbangkan titik tetangga yang belum dilalui dan hitung jaraknya dari titik keberangkatan. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru; 5) saat selesai mempertimbangkan setiap jarak terhadap titik tetangga, tandai titik yang telah dilalui sebagai "titik dilewati". Titik yang dilewati tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya; 6) ulangi langkah 4 dan 5

Algoritma Floyd Warshall

Menurut (Suriyanto & Adiya, 2020) Algoritma *Floyd-Warshall* adalah jenis pemrograman dinamis. Pemrograman dinamis sendiri adalah cara pemecahan masalah dengan melihat solusi yang diperoleh sebagai keputusan yang koheren atau saling berkaitan. Solusi yang terbentuk diperoleh dari solusi yang diturunkan dari tahap sebelumnya dan memiliki kemungkinan solusi yang diperoleh lebih dari satu. Yang membedakan dengan Algoritma Dijkstra adalah Algoritma Dijkstra bersifat serakah (*greedy*), mengeksekusi tiap-tiap langkah hanya berdasarkan nilai optimal yang diperoleh tanpa memikirkan konsekuensi masa depan jika sebuah langkah dipilih dalam sebuah skenario. Oleh karena itu, Algoritma Dijkstra dikhawatirkan belum menawarkan solusi terbaik. Berbeda dengan Algoritma *Floyd-Warshall*, Algoritma *Floyd-Warshall* mencoba menawarkan solusi memikirkan konsekuensi yang mungkin timbul di masa depan jika suatu langkah diambil.

Secara umum, algoritma *Floyd-Warshall* menurut (Ramadhani, n.d.) adalah sebagai berikut:

- a. $X = X_0$
- b. Untuk $k = 1$ hingga n , lakukan:
 Untuk $i = 1$ hingga n , lakukan:
 Untuk $j = 1$ hingga n , lakukan:
 Jika $X[i, j] > X[i, k] + X[k, j]$
 Tukar $X[i, j]$ dengan $X[i, k] + X[k, j]$ (2)
- c. $X^* = X$
 Keterangan:
 X = Matriks
 X_0 = Matriks hubung graf mula-mula
 X^* = Hasil matriks setelah perbandingan
 k = iterasi 1 sampai ke- n
 i = titik awal pada v_i
 j = titik awal pada v_j

Jika pada iterasi ke- k , hasil $X[i, k] + X[k, j]$ lebih sedikit dari $X[i, j]$ maka bobot sisi yang terdapat dalam $X[i, j]$ dapat diganti dengan hasil $X[i, k] + X[k, j]$. Selanjutnya, untuk iterasi iterasi ke- $k + 1$ akan digunakan tabel hasil iterasi iterasi ke- k yang sudah dihitung sebelumnya.

METODE PENELITIAN

Pendekatan yang dipakai dalam penelitian ini ialah pendekatan kuantitatif deskriptif. Menurut (Achmadi & Narbuko, 2015) penelitian deskriptif adalah penelitian yang bertujuan untuk menjelaskan solusi berbasis data untuk masalah saat ini dengan menyajikan, menganalisis, dan menafsirkannya. Jenis penelitian yang dipakai dalam penelitian ini yaitu studi kasus. Dalam hal ini topik tertentu tersebut ialah “Perbandingan Algoritma *Dijkstra* dan Algoritma *Floyd-Warshall* untuk Menentukan Rute Terdekat Antar- rumah sakit Di Kota Kediri”.

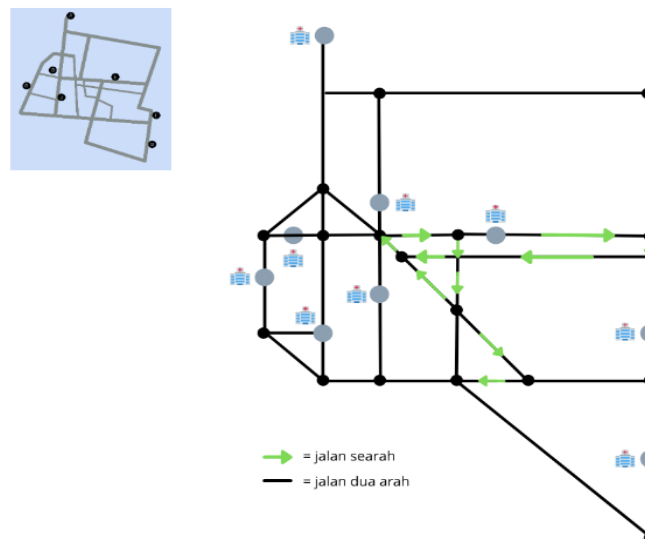
Data dalam penelitian diambil dari situs web *google maps* terkait penentuan jarak dan rute antar-rumah sakit. Sedangkan, objek penelitian dalam penelitian ini berupa beberapa rumah sakit besar yang ada di Kota Kediri yaitu RS Muhammadiyah Ahmad Dahlan, RS DKT Tk. IV, RS Umum Lirboyo, RS Umum Daha Husada, RS Umum Bhayangkara, RS Umum Daerah Kilisuci, RS Umum Ratih, RS Baptis, dan RS Umum Daerah Gambiran dengan mempertimbangkan kelengkapan layanan dan fasilitas yang ada pada rumah sakit tersebut secara umum.

Data dan sumber data yang digali dalam penelitian ini berupa data non-manusia berupa data yang sudah siap diolah antara lain: Dari *google maps*, didapatkan data berupa rute antar- rumah sakit di Kota Kediri. Selain itu, dapat mengukur jarak melalui aplikasi tersebut beserta jarak tempuh secara optimal tanpa harus mengukur secara langsung. Data Kementerian Kesehatan Republik Indonesia, terdapat data dari beberapa web Kemenkes (RI, 2018) yang membagikan informasi tentang daftar rumah sakit yang ada di Kota Kediri lengkap dengan jenis, tipe, beserta alamat lengkapnya.

Teknik analisis data yang digunakan untuk membandingkan Algoritma *Dijkstra* dan Algoritma *Floyd-Warshall* dalam menentukan rute terdekat antar- rumah sakit di Kota Kediri ialah sebagai berikut: 1) Menentukan 9 titik rumah sakit umum di Kota Kediri; 2) Menentukan rute-rute antar- rumah sakit di Kota Kediri dengan memperhatikan batasan masalah; 3) Mengubah jalur yang sudah ditentukan dari *google maps* menjadi ilustrasi peta jaringan rute antar- rumah sakit di Kota Kediri dengan bantuan *software Corel draw* dan *Canva*; 4) Menjadikan ilustrasi peta jaringan menjadi sebuah graf. Terdapat 9 titik utama yang menginterpretasikan rumah sakit umum di Kota Kediri dan beberapa titik penghubung berupa persimpangan jalan; 5) Memberi bobot dan arah pada graf. Bobot pada graf menggambarkan jarak yang ditempuh sedangkan arah merupakan rute yang dapat dilewati mempertimbangkan keadaan lapangan berupa jalan searah dan jalan dua arah; 6) Bentuk graf yang sudah jadi akan dihitung dengan Algoritma *Dijkstra* dan Algoritma *Floyd-Warshall*. Penghitungan akan dilakukan dengan metode tabel sebagai alternatif untuk menentukan jalur terpendek antar-rumah sakit di Kota Kediri.

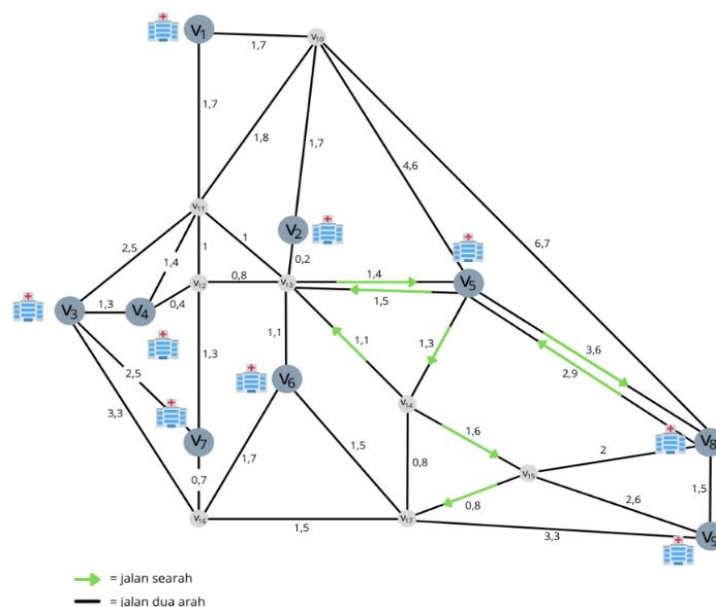
HASIL DAN PEMBAHASAN

Data rute wilayah Kota Kediri dapat diambil dari *google maps* dan *google earth*. Dari *google maps* didapatkan rute lalu lintas jalan raya secara langsung sedangkan *google earth* membantu peneliti untuk melihat kondisi dan melakukan perjalanan secara virtual di rumah sakit yang ingin dikunjungi. Bersumber dari *google maps* tersebut ditentukan rute yang menghubungkan 9 rumah sakit umum serta Panjang lintasan jalur yang ada pada rute tersebut. Ilustrasi peta jaringan rute antar-rumah sakit ditunjukkan pada Gambar 2.



Gambar 2. Ilustrasi Peta Jaringan Route Antar-rumah sakit Umum Di Kota Kediri

Ilustrasi peta jaringan pada Gambar 2 yang selanjutnya diubah menjadi bentuk graf dengan tiap-tiap rumah sakit merupakan titik dengan panjang lintasan adalah sisi dari graf tersebut. Selanjutnya untuk menentukan panjang lintasan dari tiap-tiap sisi, graf diubah menjadi graf berbobot. Graf berbobot memerlukan tiga data utama antara lain titik berupa letak rumah sakit dan persimpangan jalan, sisi berupa jalan yang menghubungkan antara satu titik ke titik lainnya yang berdekatan, dan bobot sisi berupa panjang lintasan jalan pada sisi dalam kilometer (km). Representasi graf berbobot disajikan pada Gambar 3.



Gambar 3. Representasi Graf Berbobot (dalam kilometer)

Berikut adalah 9 daftar rumah sakit umum di Kota Kediri yang disajikan dalam graf berbobot pada Gambar 2: RS Muhammadiyah Ahmad Dahlan (v_1), RS DKT Tk. IV (v_2), RS Umum Lirboyo (v_3), RS Umum Daha Husada (v_4), RS Umum Bhayangkara (v_5), RS Umum Daerah Kilisuci (v_6), RS Umum Ratih (v_7), RS Baptis (v_8), dan RS Umum Daerah Gambiran (v_9). Adapun v_{10} , v_{11} , v_{12} , v_{13} , v_{14} , v_{15} , v_{16} , v_{17} merupakan persimpangan jalan.

Implementasi Algoritma Dijkstra dalam Pencarian Rute Terdekat antar- Rumah Sakit Di Kota Kediri

Langkah pertama dalam mengerjakan implementasi Algoritma Dijkstra adalah mengubah graf ke dalam bentuk tabel berisi bobot sisi graf berdasarkan Gambar 3. Adapun tabel bobot sisi graf rute rumah sakit di Kota Kediri disajikan dalam Tabel 1.

Tabel 1. Bobot Sisi Graf Rute Rumah Sakit di Kota Kediri

X	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}
v_1	0	∞	∞	∞	∞	∞	∞	∞	∞	1,7	1,7	∞	∞	∞	∞	∞	∞
v_2	∞	0	∞	∞	∞	∞	∞	∞	∞	1,7	∞	∞	0,2	∞	∞	∞	∞
v_3	∞	∞	0	1,3	∞	∞	2,5	∞	∞	∞	2,5	∞	∞	∞	∞	3,3	∞
v_4	∞	∞	1,3	0	∞	∞	∞	∞	∞	∞	1,4	0,4	∞	∞	∞	∞	∞
v_5	∞	∞	∞	∞	0	∞	∞	3,6	∞	4,6	∞	∞	1,5	1,3	∞	∞	∞
v_6	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	1,1	∞	∞	1,7	1,5
v_7	∞	∞	2,5	∞	∞	∞	0	∞	∞	∞	∞	1,3	∞	∞	∞	0,7	∞
v_8	∞	∞	∞	∞	2,9	∞	∞	0	1,5	6,7	∞	∞	∞	∞	2	∞	∞
v_9	∞	∞	∞	∞	∞	∞	∞	1,5	0	∞	∞	∞	∞	∞	2,6	∞	3,3
v_{10}	1,7	1,7	∞	∞	4,6	∞	∞	6,7	∞	0	1,8	∞	∞	∞	∞	∞	∞
v_{11}	1,7	∞	2,5	1,4	∞	∞	∞	∞	∞	1,8	0	1	1	∞	∞	∞	∞
v_{12}	∞	∞	∞	0,4	∞	∞	1,3	∞	∞	∞	1	0	0,8	∞	∞	∞	∞
v_{13}	∞	0,2	∞	∞	1,4	1,1	∞	∞	∞	∞	1	0,8	0	∞	∞	∞	∞
v_{14}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,1	0	1,6	∞	0,8
v_{15}	∞	∞	∞	∞	∞	∞	∞	2	2,6	∞	∞	∞	∞	∞	0	∞	0,8
v_{16}	∞	∞	3,3	∞	∞	1,7	0,7	∞	∞	∞	∞	∞	∞	∞	∞	0	1,5
v_{17}	∞	∞	∞	∞	∞	1,5	∞	∞	3,3	∞	∞	∞	∞	0,8	∞	1,5	0

Dari Tabel 1 terlihat bahwa nilai-nilai tersebut merupakan bobot jarak antar 2 rumah sakit yang terhubung secara langsung. Jika terdapat rumah sakit yang tidak terhubung secara langsung maka diberi bobot ∞ . Algoritma Dijkstra dihitung dengan sederhana dan dituliskan dalam bentuk tabel secara tahap demi tahap berdasarkan perhitungan menggunakan persamaan (1) yang disajikan dalam Tabel 2.

Tabel 1. Hasil Penghitungan Algoritma Dijkstra dalam Pencarian Rute Terdekat antar-RS di Kota Kediri

X	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
v_1	0	2,9	4,2	3,1	4,1	3,8	4	7,7	8,6
v_2	2,9	0	2,7	1,4	1,6	1,3	2,3	5,2	6,1
v_3	4,2	2,7	0	1,3	3,9	3,6	2,5	7,5	8
v_4	3,1	1,4	1,3	0	2,6	2,3	1,7	6,2	7,1
v_5	4,2	1,7	4	2,7	0	2,6	3,6	3,6	5,1
v_6	3,8	1,3	3,6	2,3	2,5	0	2,4	5,9	4,8
v_7	4	2,3	2,5	1,7	3,5	2,4	0	6,6	5,5
v_8	7,1	4,6	6,9	5,6	2,9	4,3	5	0	1,5
v_9	7,9	5,4	7,7	6,4	4,4	4,8	5,5	1,5	0

Selain mendapatkan jarak rute terdekat, Algoritma Dijkstra juga menampilkan rute yang dilewati pada tahap penghitungannya. Dibawah ini dijelaskan hasil algoritma Dijkstra RS Muhammadiyah Ahmad Dahlan (v_1) sebagai titik awal sebagai berikut:

Tabel 2. Hasil Algoritma Dijkstra RS Muhammadiyah Ahmad Dahlan sebagai Titik Awal

Titik Awal	Titik Akhir	Jarak Terdekat	Rute Tempuh
v_1	v_2	2,9	$v_1 - v_{11} - v_{13} - v_2$
	v_3	4,2	$v_1 - v_{11} - v_3$
	v_4	3,1	$v_1 - v_{11} - v_4$
	v_5	4,1	$v_1 - v_{11} - v_{13} - v_5$
	v_6	3,8	$v_1 - v_{11} - v_{13} - v_6$
	v_7	4	$v_1 - v_{11} - v_{12} - v_7$
	v_8	7,7	$v_1 - v_{11} - v_{13} - v_5 - v_8$
	v_9	8,6	$v_1 - v_{11} - v_{13} - v_6 - v_{17} - v_9$

Dari **Tabel 3** dapat dijelaskan bahwa jarak terdekat dari RS Muhammadiyah Ahmad Dahlan (v_1) menuju RS DKT Tk IV (v_2) adalah sejauh 2,9 km dengan rute tempuh RS Muhammadiyah Ahmad Dahlan (v_1)-persimpangan jalan (v_{11}) - persimpangan jalan (v_{13})- RS DKT Tk IV (v_2).

Implementasi Algoritma *Floyd Warshall* dalam Pencarian Rute Terdekat antar- Rumah Sakit Di Kota Kediri

Pada Tabel 1 sudah ditampilkan bobot sisi graf rute Rumah Sakit di Kota Kediri. Dari tabel tersebut, dilakukan penghitungan Algoritma Floyd-Warshall dengan iterasi (pengulangan proses penghitungan) sebanyak jumlah titik pada graf yaitu 17 kali iterasi menggunakan rumus pada persamaan (2). Untuk mempermudah penghitungan, pada Tabel 1 terlihat bahwa banyak sel yang berisi ∞ atau tak hingga menandakan titik i ke j tidak terhubung secara langsung dalam satu sisi. Ketika sel pada $X[i, k]$ atau $X[k, j]$ bernilai ∞ , dapat dipastikan hasil dari $X[i, k] + X[k, j]$ juga akan bernilai ∞ sehingga tidak akan terpenuhi syarat pada rumus. Berikut adalah hasil iterasi terakhir penghitungan Algoritma Floyd-Warshall untuk menentukan rute terdekat antar-RS di Kota Kediri.

Tabel 4. Hasil Penghitungan Algoritma Floyd-Warshall dalam Pencarian Rute Terdekat Antar-RS di Kota Kediri

X	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}
v_1	0	2,9	4,2	3,1	4,1	3,8	4	7,7	8,6	1,7	1,7	2,7	2,7	5,4	7	4,7	5,3
v_2	2,9	0	2,7	1,4	1,6	1,3	2,3	5,2	6,1	1,7	1,2	1	0,2	2,9	4,5	3	2,8
v_3	4,2	2,7	0	1,3	3,9	3,6	2,5	7,5	8	4,3	2,5	1,7	2,5	5,2	6,8	3,2	4,7
v_4	3,1	1,4	1,3	0	2,6	2,3	1,7	6,2	7,1	3,1	1,4	0,4	1,2	3,9	5,5	2,4	3,8
v_5	4,2	1,7	4	2,7	0	2,6	3,6	3,6	5,1	3,4	2,5	2,3	1,5	1,3	2,9	3,6	2,1
v_6	3,8	1,3	3,6	2,3	2,5	0	2,4	5,9	4,8	3	2,1	1,9	1,1	2,3	3,9	1,7	1,5
v_7	4	2,3	2,5	1,7	3,5	2,4	0	6,6	5,5	4	2,3	1,3	2,1	3	4,6	0,7	2,2
v_8	7,1	4,6	6,9	5,6	2,9	4,3	5	0	1,5	6,3	5,4	5,2	4,4	3,6	2	4,3	2,8
v_9	7,9	5,4	7,7	6,4	4,4	4,8	5,5	1,5	0	7,1	6,2	6	5,2	4,1	2,6	4,8	3,3
v_{10}	1,7	1,7	4,3	3,1	3,3	3	4	6,7	7,8	0	1,8	2,7	1,9	4,6	6,2	4,7	4,5
v_{11}	1,7	1,2	2,5	1,4	2,4	2,1	2,3	6	6,9	1,8	0	1	1	3,7	5,3	3	3,6
v_{12}	2,7	1	1,7	0,4	2,2	1,9	1,3	5,8	6,7	2,7	1	0	0,8	3,5	5,1	2	3,4
v_{13}	2,7	0,2	2,5	1,2	1,4	1,1	2,1	5	5,9	1,9	1	0,8	0	2,7	4,3	2,8	2,6
v_{14}	3,8	1,3	3,6	2,3	2,5	2,2	3	3,6	4,1	3	2,1	1,9	1,1	0	1,6	2,3	0,8
v_{15}	5,4	2,9	5,2	3,9	4,1	2,3	3	2	2,6	4,6	3,7	3,5	2,7	1,6	0	2,3	0,8
v_{16}	4,7	3	3,2	2,4	4,2	1,7	0,7	5,9	4,8	4,7	3	2	2,8	2,3	3,9	0	1,5
v_{17}	4,6	2,1	4,4	3,1	3,3	1,5	2,2	4,4	3,3	3,8	2,9	2,7	1,9	0,8	2,4	1,5	0

Pada iterasi terakhir penghitungan Algoritma Floyd-Warshall, diperoleh hasil berupa jarak terdekat tanpa disertai rute tempuh. Meskipun begitu, tabel hasil yang ditampilkan merupakan jarak dari keseluruhan titik secara bersamaan satu sama lain. Hal ini memudahkan peneliti untuk mengecek rute tempuh yang dilalui dari titik satu ke titik lain melalui proses iterasi yang sudah dilakukan sebelumnya. Sehingga didapatkan rute tempuh penghitungan Algoritma Floyd-Warshall untuk menentukan jarak terdekat antar RS di Kota Kediri yang disajikan dalam Tabel 5.

Tabel 5. Hasil Jarak Terdekat dan Rute Tempuh Penghitungan Algoritma Floyd-Warshall untuk RS Muhammadiyah Ahmad Dahlan

Titik Awal	Titik Akhir	Jarak Terdekat	Rute Tempuh
v_1	v_2	2,9	$v_1 - v_{11} - v_{13} - v_2$
	v_3	4,2	$v_1 - v_{11} - v_3$
	v_4	3,1	$v_1 - v_{11} - v_4$
	v_5	4,1	$v_1 - v_{11} - v_{13} - v_5$
	v_6	3,8	$v_1 - v_{11} - v_{13} - v_6$
	v_7	4	$v_1 - v_{11} - v_{12} - v_7$
	v_8	7,7	$v_1 - v_{11} - v_{13} - v_5 - v_8$
	v_9	8,6	$v_1 - v_{11} - v_{13} - v_6 - v_{17} - v_9$

Dari Tabel 5 dapat dijelaskan bahwa jarak terdekat dari RS Muhammadiyah Ahmad Dahlan (v_1) menuju RS DKT Tk IV (v_2) adalah sejauh 2,9 km dengan rute tempuh RS Muhammadiyah Ahmad Dahlan (v_1)-persimpangan jalan (v_{11})-persimpangan jalan (v_{13})- RS DKT Tk IV (v_2). Hal ini menunjukkan hasil yang sama dengan Algoritma Dijkstra yang disajikan dalam **Tabel 3**.

Perbandingan Algoritma Dijkstra dan Algoritma Floyd Warshall dalam menentukan Rute Terdekat Antar-Rumah Sakit di Kota Kediri

Dari hasil penghitungan Algoritma Dijkstra pada Tabel 2 dan Algoritma Floyd-Warshall pada Tabel 4 diperoleh jarak terdekat beserta rute tempuh antar-Rumah Sakit di Kota Kediri bernilai sama. Meskipun memiliki hasil yang sama, terdapat beberapa hal mendasar yang dapat dibandingkan dari kedua algoritma tersebut antara lain.

1. Prinsip Algoritma

Prinsip dari Algoritma Dijkstra adalah setiap langkah yang dilakukan berdasarkan nilai optimum tanpa memikirkan konsekuensi setelah memilih langkah tersebut. Sedangkan prinsip dari Algoritma Floyd-Warshall ialah memandang solusi yang akan didapatkan sebagai keputusan yang saling terkait. (Munandar, 2022)

2. Metode

Perbedaan metode penghitungan yang paling terlihat dari Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam memperoleh jalur terpendek antar-Rumah Sakit di Kota Kediri ialah bahwa Algoritma Dijkstra menemukan jalur terpendek antara satu pasang titik sedangkan Algoritma Floyd-Warshall menemukan jalur terpendek antara semua titik.

3. Kecepatan

Dalam kasus menemukan jalur terpendek antara semua titik, Algoritma Dijkstra memerlukan jauh lebih banyak penghitungan dibandingkan Algoritma Floyd-Warshall yang artinya Algoritma Floyd-Warshall lebih cepat dibandingkan Algoritma Dijkstra.

Untuk menemukan semua jalur terpendek antar-Rumah Sakit di Kota Kediri Algoritma Dijkstra memerlukan 72 kali penghitungan, sedangkan pada algoritma Floyd-Warshall hanya memerlukan 17 kali iterasi (pengulangan proses penghitungan) diperoleh bukan hanya antar-rumah sakit tetapi juga antara semua titik termasuk titik pada persimpangan jalan.

4. Hasil

Penghitungan Algoritma Dijkstra dan Algoritma Floyd-Warshall untuk menemukan jalur terpendek antar-Rumah Sakit di Kota Kediri, diperoleh hasil yang sama antara penggunaan Algoritma Dijkstra dan algoritma Floyd-Warshall. Perbedaannya adalah Algoritma Dijkstra memiliki hasil akhir berupa jarak terdekat dan rute tempuh sekaligus sedangkan Algoritma Floyd-Warshall hanya diperoleh hasil akhir berupa jarak terdekat dengan rute tempuh yang dapat dicari secara terpisah dengan melihat kembali proses iterasi.

5. Kerumitan

Untuk menghitung jalur terpendek antar-Rumah Sakit di Kota Kediri, Algoritma Dijkstra lebih rumit dibandingkan dengan Algoritma Floyd-Warshall. Hal ini dikarenakan Algoritma Floyd-Warshall lebih efisien dan

mudah diterapkan karena setiap kali iterasi akan mengambil rute paling minimal antara semua titik dari rute sebelumnya. Sedangkan Dijkstra hanya fokus untuk menemukan jalur terdekat antara satu titik dengan titik lain dan akan mengulang penghitungan dari awal apabila ingin mencari titik lain yang belum ditemukan.

6. Keefektifan

Dengan diperoleh hasil yang sama dan dilihat dari beberapa aspek yang disebutkan sebelumnya berupa prinsip, metode, kecepatan dan kerumitan, Algoritma Floyd-Warshall lebih efektif dibandingkan dengan Algoritma Dijkstra dalam hal menentukan rute terdekat antar-Rumah Sakit di Kota Kediri.

SIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa Algoritma Dijkstra dan Algoritma Floyd-Warshall sangat membantu dalam menentukan rute terdekat. Dalam implementasinya untuk menentukan rute terdekat antar-Rumah Sakit di Kota Kediri, kedua algoritma ini memiliki hasil jarak terdekat dan rute tempuh yang sama. Meskipun begitu, masing-masing algoritma memiliki karakteristik bahkan cara penghitungan yang berbeda meliputi prinsip, metode, kecepatan, kerumitan dan hasil serta keefektifan. Dalam penelitian ini, Algoritma Floyd-Warshall lebih efektif dibandingkan dengan Algoritma Dijkstra untuk menentukan rute terdekat antar-Rumah Sakit di Kota Kediri.

REFERENSI

- Aulia, R., Syahputra, E. R., & Dafitri, H. (2015). *Sistem Pencarian Rumah Sakit Terdekat Menggunakan Algoritma Dijkstra Berbasis Android (Studi Kasus: Rumah Sakit di Kota Medan)*. 1, 150–155.
- Achmadi, A., & Narbuko. (2015). *Metodologi Penelitian*. Bumi Aksara.
- Attamimi, I., Yahya, W., & Hanafi, M. H. (2017). Analisis Perbandingan Algoritma Floyd-Warshall dan Dijkstra untuk Menentukan Jalur Terpendek Pada Jaringan Openflow. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK)*, 1(12), 1842–1849.
- Munandar, A. (2022). *Pengantar Matematika Diskrit dan Teori Graf*. Buku Pendidikan Deepublish.
- Munir, R. (2012). *Matematika Diskrit*. Penerbit Informatika.
- Ningrum, F. W., & Andrasto, T. (2016). Penerapan Algoritma Floyd-Warshall Dalam Menentukan Rute Terpendek Pada Pemodelan Jaringan Pariwisata Di Kota Semarang. *Jurnal Teknik Elektro Unnes*, 8(1), 21–24. <https://www.neliti.com/id/publications/140473/penerapan-algoritma-floyd-warshall-dalam-menentukan-rute-terpendek-pada-pemodela>
- Peraturan Menteri Kesehatan Republik Indonesia Nomor 11 Tahun 2017. Tentang Keselamatan Pasien Rumah Sakit, (2017).
- Rahayuningsih, S. (n.d.). *Teori Graph dan Penerapannya*. Universitas Wisnuwardhana Press Malang.
- Ramadhani, C. (n.d.). *Teori & Algoritma Graph dengan Bahasa Java*. Andi Publisher.
- RI, K. K. (2018). *Ditjen Yankes*. <https://sirs.kemkes.go.id/fo/login>
- Satyananda, D. (2012). *Struktur Data*. Universitas Negeri Malang.
- Siang, J. J. (n.d.). *Matematika diskrit dan Aplikasinya pada Ilmu Komputer*. Andi Offset.
- Surianto, & Adiya, M. H. (2020). *Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall Dalam Pencarian Lokasi Kuliner*. 2(3), 128–133.
- Wilson, R. J. (2010). *Pengantar Teori Graf*. Erlangga.
- Yusuf, M. S., Az-zahra, H. M., & Apriyanti, D. H. (2017). Implementasi Algoritma Dijkstra Dalam Menemukan Jarak Terdekat Dari Lokasi Pengguna Ke Tanaman Yang Di Tuju Berbasis Android (Studi Kasus di Kebun Raya Purwodadi). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 1(12), 1779–1781.