

## OPTIMASI RUTE PENDISTRIBUSIAN GAS ELPIJI MENGGUNAKAN ALGORITMA FLOYD WARSHALL DAN ALGORITMA GREEDY

Elia Resita Ningrum<sup>1</sup>, Ardhi Sanwidi<sup>2</sup>, Rachmadania Akbarita<sup>3</sup>, dan Muhammad Nur Haqul Qomaruddin<sup>4</sup>

<sup>1,2,3,4</sup>Universitas Nahdlatul Ulama Blitar

<sup>1</sup>eningrum66@gmail.com, <sup>2</sup>ardhisawidi@gmail.com, <sup>3</sup>dania.barita@gmail.com, haqquludin@gmail.com

### ABSTRACT

This study aims to find the results of the shortest route for LPG distribution using Floyd-Warshall and Greedy algorithms and to find the shortest path that is more optimal between those two algorithms. The distribution route optimization in this study is done by minimizing the distance that will be taken by the distribution vehicle. Floyd-Warshall algorithm and Greedy algorithm are optimization methods with a weighted graph that has vertices as location points and edges as distances. The instrument of this study is the Google Maps application which is arranged into the tables provided, and reports from PT Petro Jaya Gas which are also in the form of tables. In analyzing the data using the Floyd-Warshall algorithm, the vertices of the distribution route obtained are  $V_1 - V_2 - V_3 - V_8 - V_4 - V_5 - V_{10} - V_7 - V_{11} - V_6 - V_9 - V_{12} - V_{13} - V_{14}$  with the total distance of 13.89 km, and the travel time of 48 minutes. While in analyzing the data using the Greedy algorithm, the vertices of the distribution route obtained are  $V_1 - V_2 - V_3 - V_4 - V_5 - V_6 - V_7 - V_9 - V_8 - V_{12} - V_{13} - V_{14} - V_{11} - V_{10}$  with the total distance of 13.13 km, and the travel time of 45 minutes. Based on the result of analysis, it can be concluded that the most optimum method that can be used to solve the problem about the shortest route in distributing LPG of PT Petro Jaya Gas is using the Greedy algorithm.

**Keywords** : Floyd Warshall, Greedy, The Shortest Route Optimization.

### ABSTRAK

Penelitian ini bertujuan untuk mengetahui hasil rute terpendek menggunakan algoritma *Floyd Warshall* dan algoritma *Greedy* dan mengetahui lintasan terpendek yang lebih optimal diantara kedua algoritma tersebut. Optimasi rute pendistribusian dalam penelitian ini dilakukan dengan meminimumkan jarak yang akan ditempuh oleh kendaraan pendistribusian. Algoritma *Floyd Warshall* dan algoritma *Greedy* merupakan metode optimasi dengan graf berbobot yang memiliki *verteks* sebagai titik lokasi dan *edge* sebagai jarak tempuh. Adapun instrumen yang digunakan dalam penelitian ini antara lain aplikasi *google maps* yang disusun ke dalam tabel yang disediakan dan laporan dari PT Petro Jaya Gas dalam bentuk tabel. Dalam analisis data menggunakan algoritma *Floyd Warshall* diperoleh *verteks* rute pendistribusian yaitu  $V_1 - V_2 - V_3 - V_8 - V_4 - V_5 - V_{10} - V_7 - V_{11} - V_6 - V_9 - V_{12} - V_{13} - V_{14}$  dengan total jarak yaitu 13,89 km dan waktu tempuh 48 menit. Dalam analisis data menggunakan algoritma *Greedy* diperoleh *verteks* rute pendistribusian  $V_1 - V_2 - V_3 - V_4 - V_5 - V_6 - V_7 - V_9 - V_8 - V_{12} - V_{13} - V_{14} - V_{11} - V_{10}$  dengan total jarak yaitu 13,13 km dan waktu tempuh 45 menit. Berdasarkan hasil analisis tersebut,

dapat disimpulkan bahwa metode yang paling optimum untuk digunakan dalam pemecahan masalah rute terpendek pendistribusian gas elpiji PT Petro Jaya Gas adalah menggunakan algoritma *Greedy*.

**Kata Kunci** : Floyd Warshall, Greedy, Optimasi Rute Terpendek.

## I. PENDAHULUAN

LPG (*Liquified Petroleum Gas*) adalah bahan bakar gas yang terdiri dari gas propana ( $C_3H_8$ ) dan butana ( $C_4H_{10}$ ) yang dicairkan. Produk elpiji dijual oleh Pertamina sebagai pengganti penggunaan bahan bakar minyak tanah yang selama ini lebih banyak digunakan di rumah tangga dan industri (Subakdo & Nugroho, 2016). Penggunaan gas elpiji dinilai lebih praktis, lebih bersih, dan lebih cepat pemanasannya dibandingkan kayu bakar maupun minyak tanah (Akhir & Luckitasari, 2019). Gas elpiji merupakan kebutuhan yang penting karena digunakan sebagai alat bantu memasak. Banyaknya kebutuhan masyarakat terhadap gas elpiji diperlukan juga pendistribusian yang efektif agar gas elpiji tersedia ketika masyarakat membutuhkannya. Tahap pendistribusian gas elpiji memiliki peran penting dalam penyampaian gas elpiji kepada pihak konsumen.

Berbagai permasalahan memungkinkan pendistribusian gas menjadi kurang efektif. Permasalahan tersebut diantaranya adalah kepadatan lalu lintas, jarak dari SPPBE ke setiap agen dan jarak agen ke pangkalan yang terlalu jauh, jumlah kendaraan dan kapasitas kendaraan (Akhir & Luckitasari, 2019). Selain itu kurang efektifnya rute kendaraan juga dapat mengakibatkan biaya yang dikeluarkan akan lebih banyak (Amri, 2014). Pengoptimalan jarak pada setiap pendistribusian dapat memberi keuntungan kepada pihak agen karena memungkinkan dapat mengurangi waktu tempuh kendaraan pengangkut gas elpiji. Selain itu, selama melakukan rute pendistribusian, kendaraan pengangkut gas elpiji memiliki waktu dan biaya transportasi yang dapat di minimalisir.

PT Petro Jaya Gas merupakan agen yang bertempat di Kota Blitar yang berada dibawah naungan perusahaan Mayangkara Group. Agen ini melakukan pengisian gas elpiji di SPPBE PT Rama Manggala Gas. Penelitian ini mengambil agen di Kota Blitar karena lalu lintas di kota lebih padat dan karena banyaknya *outlet* sehingga diperlukan rute distribusi terpendek agar pendistribusian lebih optimal. Pencarian rute distribusi terpendek tersebut menggunakan algoritma *Floyd Warshall* dan algoritma *Greedy* sebagai solusi pemecahan masalah. Pengambilan data pada agen tersebut adalah alamat pangkalan (*outlet*) dan jadwal pendistribusian setiap pangkalan (*outlet*) yang akan diberikan pasokan gas elpiji oleh agen. Data tersebut adalah data yang digunakan untuk menentukan rute distribusi terpendek.

Penelitian sebelumnya terbukti bahwa hasil rute dengan menggunakan algoritma *Floyd Warshall* lebih optimal dibandingkan dengan rute distribusi yang dimiliki oleh agen PT Raja Gasindo, PT Rapy Ray Putratama dan PT X (Mukti dkk, 2018). Algoritma *Greedy* juga terbukti dapat menghasilkan pendistribusian produk AMDK ke outlet dengan jarak optimum (Jakaria, 2017). Kedua algoritma ini menggunakan graf berbobot. Graf berbobot merupakan graf yang memiliki bobot satuan

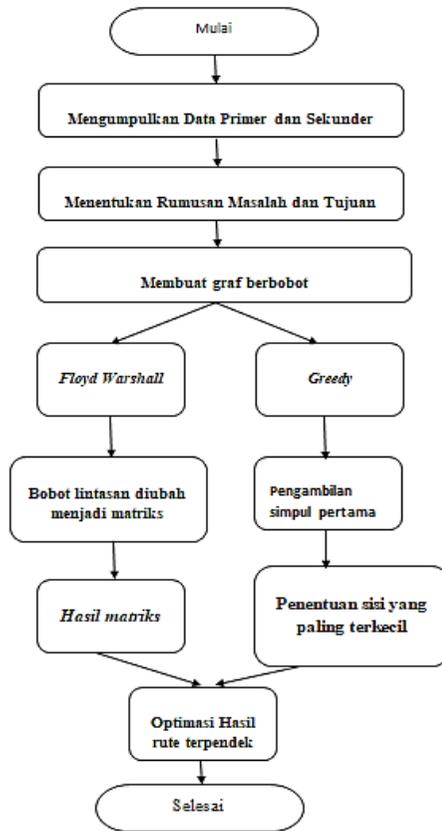
bisa berupa panjang, jarak, jumlah barang. Penelitian ini menggunakan SPPBE dan pangkalan (*outlet*) sebagai *verteks* (titik) dan jarak antar *verteks* sebagai *edge* (sisi). Kedua algoritma ini memiliki kelebihan yang sama-sama menonjol di dalam metode yang umum digunakan di dalam pencarian rute terpendek. Algoritma *Floyd Warshall* mampu menghasilkan rute terpendek secara akurat (Anggoro, 2015) dan algoritma *Greedy* mampu menyediakan solusi tercepat (Purnomo et al., n.d, 2011). Akan dicari algoritma yang lebih optimal/efektif diantara algoritma *Floyd Warshall* dan algoritma *Greedy* yang lebih menghemat waktu dan jarak tempuh. Hasil dari kedua algoritma tersebut adalah sama-sama berupa rute dan nanti akan terlihat algoritma yang memiliki rute yang lebih optimal. Rute yang paling optimal dilihat dari keseluruhan jumlah jarak yang paling minimum.

Pada penelitian ini menggunakan algoritma *Floyd Warshall* dan algoritma *Greedy* yang diimplementasikan pada satu objek sekaligus yaitu PT Petro Jaya Gas. Berbeda dengan penelitian sebelumnya yaitu mengimplementasikan setiap algoritma tersebut ke masing-masing objek yang berbeda. Merujuk pada penelitian oleh Anggoro (2015) dimana pemanfaatan Algoritma *Floyd Warshall* mampu mengoptimalkan pencarian rute terdekat untuk pencarian tempat. Algoritma *Floyd Warshall* memiliki solusi yang didapatkan sebagai keputusan yang saling terkait, algoritma ini melihat solusi yang dibentuk pada tahap sebelumnya (Darmawan & Ghaniy, 2018). Algoritma *Greedy* merupakan metode yang digunakan dalam suatu penyelesaian masalah optimasi yang diselesaikan secara bertahap (*step by step*), dengan harapan hasil yang didapatkan merupakan hasil terbaik yang dapat diperoleh (Kurniasari, 2006). Hasil dari algoritma *Greedy* bila dibandingkan dengan algoritma lainnya memiliki hasil optimasi yang bagus dengan waktu yang lebih cepat (Purnomo, 2011).

## II. METODE PENELITIAN

Penelitian ini menggunakan data primer dan data sekunder sebagai sumber data penelitian. Data primer tersebut merupakan data langsung yang diamati oleh penulis dari *google maps* berupa peta dan jarak antar lokasi. Data sekunder berupa alamat setiap pangkalan(*outlet*) dan jadwal pendistribusian setiap pangkalan(*outlet*) yang diperoleh dari PT. Petro Jaya Gas. Teknik pengumpulan datanya yaitu dengan observasi pencarian titik lokasi dan jarak yang diperoleh dari *google maps* dan dengan dokumentasi pada instansi terkait.

Penelitian yang dilakukan merupakan jenis penelitian murni sekaligus terapan dengan menggunakan algoritma *Floyd Warshall* dan algoritma *Greedy* yang bertujuan untuk mencari rute terpendek. Untuk algoritma *Floyd Warshall* bobot lintasan yang diperoleh di ubah menjadi matriks bujur sangkar  $n \times n$  ( $n$  adalah jumlah *verteks*) kemudian dianalisis. Untuk algoritma *Greedy* bobot lintasan digunakan untuk mengambil jarak optimal. Hasil dari algoritma *Floyd Warshall* dan algoritma *Greedy* berupa rute terpendek yang akan dilewati oleh kendaraan pengangkut gas elpiji. Gambaran alur penelitian dapat dilihat pada diagram alir yang di tunjukkan pada Gambar 1.



Gambar 1 : *Flowchart* alur penelitian

### III. HASIL DAN PEMBAHASAN

#### 3.1. Hasil Penelitian

##### 3.1.1. Perhitungan dengan Algoritma Floyd Warshall

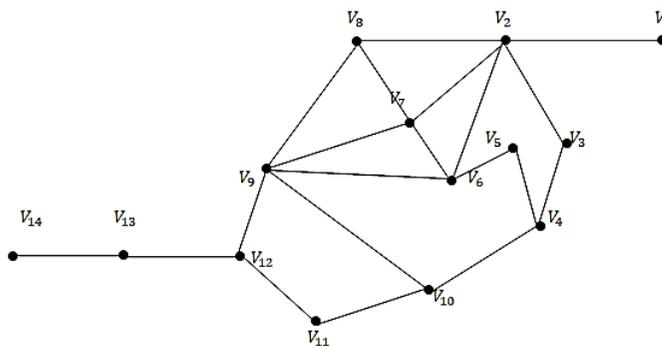
Dari data yang diperoleh dari *google maps* maka didapatkan peta wilayah dan titik lokasi sebagai gambaran umum penelitian. Titik lokasi tersebut jika diubah ke dalam bentuk *verteks* menjadi sebagai Tabel 1.

Tabel 1 : Representasi *verteks* terhadap titik lokasi

No	<i>Verteks</i>	Lokasi
1	$V_1$	PT Rama Manggala Gas
2	$V_2$	PT Mayangkara LPG
3	$V_3$	JI Galunggung RT 03 RW 02 Sukorejo, Sukorejo
4	$V_4$	JI Mayang Tengah RT 02 RW 05 Sukorejo, Sukorejo
5	$V_5$	JI Mayang Tengah RT 03 RW 05 Sukorejo, Sukorejo
6	$V_6$	Jl. Bakung Sukorejo, Sukorejo

7	$V_7$	Jl Kaliporong, Pakunden, Sukorejo
8	$V_8$	Jl. Bengawan Solo, Pakunden, Sukorejo
9	$V_9$	PT Hari Putra Santosa Jaya SPBU no 54.66103
10	$V_{10}$	Jl. Mawar, Sukorejo, Sukorejo
11	$V_{11}$	Toko Buwana
12	$V_{12}$	Jl. Jati GG 1 No 27 RT 02 RW 12, Sukorejo, Sukorejo
13	$V_{13}$	Jl. Jati No 65 RT 04 RW 12, Sukorejo, Sukorejo
14	$V_{14}$	Jl. Jati RT 02 RW 14, Sukorejo, Sukorejo

Kemudian didapatkan graf sebagai berikut dengan *verteks* sebagai titik lokasi dan *edge* sebagai jarak :



Gambar 2 : Graf berbobot pendistribusian

Pengolahan data dilakukan dengan memodelkan rute yang akan dilewati menjadi titik awal dan titik tujuan untuk mendapatkan rute terpendek. Titik direpresentasikan sebagai lokasi dan sisi menggambarkan jarak, maka algoritma *Floyd-Warshall* melakukan perhitungan terhadap semua kemungkinan bobot terkecil dari setiap titik. Dari graf pada Gambar 2 diperoleh data pada Tabel 2.

Tabel 2 : Jarak dan waktu tempuh antar *verteks*

No	Pasangan titik ( <i>Verteks</i> )	Jarak (km)	Waktu tempuh(menit)
1	$V_1 - V_2$	2,7	6
2	$V_2 - V_3$	2,1	7
3	$V_2 - V_6$	3,2	8
4	$V_2 - V_7$	3	7
5	$V_2 - V_8$	2,6	7
6	$V_3 - V_4$	0,65	3
7	$V_4 - V_5$	0,12	1
8	$V_3 - V_8$	0,9	3
9	$V_5 - V_6$	0,26	1

10	$V_5 - V_{10}$	0,26	2
11	$V_6 - V_7$	0,6	2
12	$V_7 - V_8$	1,2	2
13	$V_8 - V_9$	1,2	4
14	$V_8 - V_4$	0,6	2
15	$V_7 - V_9$	0,8	3
16	$V_7 - V_{10}$	1,12	5
17	$V_7 - V_{11}$	1,32	6
18	$V_{11} - V_6$	0,72	4
19	$V_6 - V_9$	1,4	5
20	$V_9 - V_{10}$	1,5	5
21	$V_4 - V_{10}$	0,14	1
22	$V_{10} - V_{11}$	0,2	1
23	$V_{11} - V_{12}$	1,2	4
24	$V_9 - V_{12}$	1,2	5
25	$V_{12} - V_{13}$	0,35	1
26	$V_{13} - V_{14}$	0,6	1

Dari Tabel 2 terbentuklah matriks  $n \times n$  (dengan  $n$  adalah jumlah *vertices*). Dengan ketentuan matriks  $w_{ij}$  bernilai 0 jika  $i = j$ , bernilai bobot edge  $(i,j)$  jika  $i \neq j$  dan  $(i,j) \in E$  dan bernilai  $\infty$  jika  $i \neq j$  dan  $(i,j) \notin E$ . Sehingga didapatkan matriks awal  $D^{(0)} =$

$$\begin{bmatrix} 0 & 2,7 & \infty \\ 2,7 & 0 & 2,1 & \infty & \infty & 3,2 & 3 & 2,6 & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 2,1 & 0 & 0,65 & \infty \\ \infty & \infty & 0,65 & 0 & 0,12 & \infty & \infty & \infty & \infty & 0,14 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0,12 & 0 & 0,26 & \infty \\ \infty & 3,2 & \infty & \infty & 0,26 & 0 & 0,6 & \infty & 1,4 & \infty & \infty & \infty & \infty & \infty \\ \infty & 3 & \infty & \infty & \infty & 0,6 & 0 & 1,2 & 0,8 & \infty & \infty & \infty & \infty & \infty \\ \infty & 2,6 & \infty & \infty & \infty & \infty & 1,2 & 0 & 1,2 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 1,4 & 0,8 & 1,2 & 0 & 1,5 & \infty & 1,2 & \infty & \infty \\ \infty & \infty & \infty & 0,14 & \infty & \infty & \infty & \infty & 1,5 & 0 & 0,2 & \infty & \infty & \infty \\ \infty & 0,2 & 0 & 1,2 & \infty & \infty \\ \infty & 1,2 & \infty & 1,2 & 0 & 0,35 & \infty \\ \infty & 0,35 & 0 & 0,6 \\ \infty & 0,6 & 0 \end{bmatrix}$$

Setelah didapatkan matriks awal kemudian dilakukan iterasi sampai sejumlah verteksnya yaitu 14 iterasi. Iterasi dilakukan dengan melihat sel pada setiap matriks yaitu dengan melihat nilai  $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$ . Kemudian didapatkan iterasi sebagai berikut :

**Iterasi k=1**

$D^{(1)} =$

0	2,7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2,7	0	2,1	∞	∞	3,2	3	2,6	∞	∞	∞	∞	∞	∞	∞
∞	2,1	0	0,65	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	∞	0,65	0	0,12	∞	∞	∞	0,14	∞	∞	∞	∞	∞	∞
∞	∞	∞	0,12	0	0,26	∞	∞	∞	∞	∞	∞	∞	∞	∞
∞	3,2	∞	∞	0,26	0	0,6	∞	1,4	∞	∞	∞	∞	∞	∞
∞	3	∞	∞	∞	0,6	0	1,2	0,8	∞	∞	∞	∞	∞	∞
∞	2,6	∞	∞	∞	∞	1,2	0	1,2	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	1,4	0,8	1,2	0	1,5	∞	1,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	1,5	0	0,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	0,2	0	1,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	1,2	∞	1,2	0	0,35	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,35	0	0,6
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6	0

**Iterasi k=2**

$D^{(2)} =$

0	2,7	4,8	∞	∞	∞	5,9	5,7	5,3	∞	∞	∞	∞	∞	∞
2,7	0	2,1	∞	∞	∞	3,2	3	2,6	∞	∞	∞	∞	∞	∞
4,8	2,1	0	0,65	∞	∞	5,3	5,1	4,7	∞	∞	∞	∞	∞	∞
∞	∞	0,65	0	0,12	∞	∞	∞	∞	0,14	∞	∞	∞	∞	∞
∞	∞	∞	0,12	0	0,26	∞	∞	∞	∞	∞	∞	∞	∞	∞
5,9	3,2	5,3	∞	0,26	0	0,6	5,8	1,4	∞	∞	∞	∞	∞	∞
5,7	3	5,1	∞	∞	0,6	0	1,2	0,8	∞	∞	∞	∞	∞	∞
5,3	2,6	4,7	∞	∞	∞	5,8	1,2	0	1,2	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	1,4	0,8	1,2	0	1,5	∞	1,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	1,5	0	0,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	0,2	0	1,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	1,2	∞	1,2	0	0,35	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,35	0	0,6
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6	0

**Iterasi k=3**

$D^{(3)} =$

0	2,7	4,8	5,45	∞	5,9	5,7	5,3	∞	∞	∞	∞	∞	∞	∞
2,7	0	2,1	2,75	∞	3,2	3	2,6	∞	∞	∞	∞	∞	∞	∞
4,8	2,1	0	0,65	∞	5,3	5,1	4,7	∞	∞	∞	∞	∞	∞	∞
5,45	2,75	0,65	0	0,12	5,95	5,75	5,35	∞	0,1	∞	∞	∞	∞	∞
∞	∞	∞	0,12	0	0,26	∞	∞	∞	∞	∞	∞	∞	∞	∞
5,9	3,2	5,3	5,95	0,26	0	0,6	5,8	1,4	∞	∞	∞	∞	∞	∞
5,7	3	5,1	5,75	∞	0,6	0	1,2	0,8	∞	∞	∞	∞	∞	∞
5,3	2,6	4,7	5,35	∞	5,8	1,2	0	1,2	∞	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	1,4	0,8	1,2	0	1,5	∞	1,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	1,5	0	0,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	0,2	0	1,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	1,2	∞	1,2	0	0,35	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,35	0	0,6
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6	0

**Iterasi k=4**

$D^{(4)} =$

0	2,7	4,8	5,45	5,57	5,9	5,7	5,3	∞	5,59	∞	∞	∞	∞	∞
2,7	0	2,1	2,75	2,87	3,2	3	2,6	∞	2,89	∞	∞	∞	∞	∞
4,8	2,1	0	0,65	0,77	1,03	5,1	4,7	∞	0,79	∞	∞	∞	∞	∞
5,45	2,75	0,65	0	0,12	5,95	5,75	5,35	∞	0,1	∞	∞	∞	∞	∞
∞	∞	∞	0,12	0	0,26	∞	∞	∞	∞	∞	∞	∞	∞	∞
5,57	2,87	0,77	0,12	0	0,26	5,87	5,47	∞	0,26	∞	∞	∞	∞	∞
5,9	3,2	5,3	5,95	0,26	0	0,6	5,8	1,4	6,09	∞	∞	∞	∞	∞
5,7	3	5,1	5,75	5,87	0,6	0	1,2	0,8	5,89	∞	∞	∞	∞	∞
5,3	2,6	4,7	5,35	5,47	5,8	1,2	0	1,2	5,49	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,4	0,8	1,2	0	1,5
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,4	0,8	1,2	0	1,5
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,5	0	0,2	∞	∞
5,59	2,89	0,79	0,14	0,26	6,09	5,89	5,49	1,5	0	0,2	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,2	0	1,2	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,2	∞	1,2	0	0,35
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,35	0
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6

**Iterasi k=5**

$D^{(5)} =$

0	2,7	4,8	5,45	5,57	5,83	5,7	5,3	∞	5,59	∞	∞	∞	∞	∞
2,7	0	2,1	2,75	2,87	3,13	3	2,6	∞	2,89	∞	∞	∞	∞	∞
4,8	2,1	0	0,65	0,77	1,03	5,1	4,7	∞	0,79	∞	∞	∞	∞	∞
5,45	2,75	0,65	0	0,12	0,38	5,75	5,35	∞	0,1	∞	∞	∞	∞	∞
∞	∞	∞	0,12	0	0,26	5,87	5,47	∞	0,26	∞	∞	∞	∞	∞
5,57	2,87	0,77	0,12	0	0,26	5,87	5,47	∞	0,26	∞	∞	∞	∞	∞
5,83	3,13	1,03	0,38	0,26	0	0,6	5,73	1,4	0,52	∞	∞	∞	∞	∞
5,7	3	5,1	5,75	5,87	0,6	0	1,2	0,8	5,89	∞	∞	∞	∞	∞
5,3	2,6	4,7	5,35	5,47	5,73	1,2	0	1,2	5,49	∞	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,4	0,8	1,2	0	1,5
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,5	0	0,2	∞	∞
5,59	2,89	0,79	0,14	0,26	0,52	5,89	5,49	1,5	0	0,2	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,2	0	1,2	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,2	∞	1,2	0	0,35
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,35	0
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6

**Iterasi k=6**

$D^{(6)} =$

0	2,7	4,8	5,45	5,57	5,83	5,7	5,3	7,23	3,59	∞	∞	∞	∞	∞
2,7	0	2,1	2,75	2,87	3,13	3	2,6	4,53	2,89	∞	∞	∞	∞	∞
4,8	2,1	0	0,65	0,77	1,03	1,63	4,7	2,43	0,79	∞	∞	∞	∞	∞
5,45	2,75	0,65	0	0,12	0,38	0,98	5,35	1,78	0,1	∞	∞	∞	∞	∞
∞	∞	∞	0,12	0	0,26	0,86	5,47	1,66	0,26	∞	∞	∞	∞	∞
5,57	2,87	0,77	0,12	0	0,26	0,86	5,47	1,66	0,26	∞	∞	∞	∞	∞
5,83	3,13	1,03	0,38	0,26	0	0,6	5,73	1,4	0,52	∞	∞	∞	∞	∞
5,7	3	1,63	0,98	0,86	0,6	0	1,2	0,8	1,12	∞	∞	∞	∞	∞
5,3	2,6	4,7	5,35	5,47	5,73	1,2	0	1,2	5,49	∞	∞	∞	∞	∞
7,23	4,53	2,43	1,78	1,66	1,4	0,8	1,2	0	1,5	∞	1,2	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,4	0,8	1,2	0	1,5
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,5	0	0,2	∞	∞
5,59	2,89	0,79	0,14	0,26	0,52	1,12	5,49	1,5	0	0,2	∞	∞	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,2	0	1,2	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1,2	∞	1,2	0	0,35
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,35	0
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6

**Iterasi k=7**

$D^{(7)} =$

0	2,7	4,8	5,45	5,57	5,83	5,7	5,3	6,5	5,59	∞	∞	∞	∞	∞
2,7	0	2,1	2,75	2,87	3,13	3	2,6	3,8	2,89	∞	∞	∞	∞	∞
4,8	2,1	0	0,65	0,77	1,03	1,63	2,83	2,43	0,79	∞	∞	∞	∞	∞
5,45	2,75	0,65	0	0,12	0,38	0,98	2,18	1,78	0,1	∞	∞	∞	∞	∞
∞	∞	∞	0,12	0	0,26	0,86	2,06	1,66	0,26	∞	∞	∞	∞	∞
5,57	2,87	0,77	0,12	0	0,26	0,86	2,06	1,66	0,26	∞	∞	∞	∞	∞
5,83	3,13	1,03	0,38	0,26	0	0,6	1,8	1,4	0,52	∞	∞	∞	∞	

### Iterasi k=9

$D^{(9)}=$

0	2,7	4,8	5,45	5,57	5,83	5,7	5,3	6,5	5,59	∞	7,7	∞	∞
2,7	0	2,1	2,75	2,87	3,13	3	2,6	3,8	2,89	∞	5	∞	∞
4,8	2,1	0	0,65	0,77	1,03	1,63	2,83	2,43	0,79	∞	3,63	∞	∞
5,45	2,75	0,65	0	0,12	0,38	0,98	2,18	1,78	0,1	∞	2,98	∞	∞
5,57	2,87	0,77	0,12	0	0,26	0,86	2,06	1,66	0,26	∞	2,86	∞	∞
5,83	3,13	1,03	0,38	0,26	0	0,6	1,8	1,4	0,52	∞	2,6	∞	∞
5,7	3	1,63	0,98	0,86	0,6	0	1,2	0,8	1,12	∞	2	∞	∞
5,3	2,6	2,83	2,18	2,06	1,8	1,2	0	1,2	2,32	∞	2,4	∞	∞
6,5	3,8	2,43	1,78	1,66	1,4	0,8	1,2	0	1,5	∞	1,2	∞	∞
5,59	2,89	0,79	0,14	0,26	0,52	1,12	2,32	1,5	0	0,2	2,7	∞	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	0,2	0	1,2	∞	∞
7,7	5	3,63	2,98	2,86	2,6	2	2,4	1,2	2,7	1,2	0	0,35	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,35	0	0,6	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6	0	∞

### Iterasi k=11

$D^{(11)}=$

0	2,7	4,8	5,45	5,57	5,83	5,7	5,3	6,5	5,59	5,79	6,99	∞	∞
2,7	0	2,1	2,75	2,87	3,13	3	2,6	3,8	2,89	3,09	4,29	∞	∞
4,8	2,1	0	0,65	0,77	1,03	1,63	2,83	2,29	0,79	0,99	2,19	∞	∞
5,45	2,75	0,65	0	0,12	0,38	0,98	2,18	1,64	0,1	0,34	1,54	∞	∞
5,57	2,87	0,77	0,12	0	0,26	0,86	2,06	1,66	0,26	0,46	1,66	∞	∞
5,83	3,13	1,03	0,38	0,26	0	0,6	1,8	1,4	0,52	0,72	1,92	∞	∞
5,7	3	1,63	0,98	0,86	0,6	0	1,2	0,8	1,12	1,32	2	∞	∞
5,3	2,6	2,83	2,18	2,06	1,8	1,2	0	1,2	2,32	2,52	2,4	∞	∞
6,5	3,8	2,29	1,64	1,66	1,4	0,8	1,2	0	1,5	1,7	1,2	∞	∞
5,59	2,89	0,79	0,14	0,26	0,52	1,12	2,32	1,5	0	0,2	1,4	∞	∞
5,79	3,09	0,99	0,34	0,46	0,72	1,32	2,52	1,7	0,2	0	1,2	∞	∞
6,99	4,29	2,19	1,54	1,66	1,92	2	2,4	1,2	1,4	1,2	0	0,35	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,35	0	0,6
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6	0

### Iterasi k=13

$D^{(13)}=$

0	2,7	4,8	5,45	5,57	5,83	5,7	5,3	6,5	5,59	5,79	6,99	7,34	7,94
2,7	0	2,1	2,75	2,87	3,13	3	2,6	3,8	2,89	3,09	4,29	4,64	5,24
4,8	2,1	0	0,65	0,77	1,03	1,63	2,83	2,29	0,79	0,99	2,19	2,54	3,14
5,45	2,75	0,65	0	0,12	0,38	0,98	2,18	1,64	0,1	0,34	1,54	1,89	2,49
5,57	2,87	0,77	0,12	0	0,26	0,86	2,06	1,66	0,26	0,46	1,66	2,01	2,61
5,83	3,13	1,03	0,38	0,26	0	0,6	1,8	1,4	0,52	0,72	1,92	2,27	2,87
5,7	3	1,63	0,98	0,86	0,6	0	1,2	0,8	1,12	1,32	2	2,35	2,95
5,3	2,6	2,83	2,18	2,06	1,8	1,2	0	1,2	2,32	2,52	2,4	2,75	3,35
6,5	3,8	2,29	1,64	1,66	1,4	0,8	1,2	0	1,5	1,7	1,2	1,55	2,15
5,59	2,89	0,79	0,14	0,26	0,52	1,12	2,32	1,5	0	0,2	1,4	1,75	2,35
5,79	3,09	0,99	0,34	0,46	0,72	1,32	2,52	1,7	0,2	0	1,2	1,55	2,15
6,99	4,29	2,19	1,54	1,66	1,92	2	2,4	1,2	1,4	1,2	0	0,35	0,95
7,34	4,64	2,54	1,89	2,01	2,27	2,35	2,75	1,55	1,75	1,55	0,35	0	0,6
7,94	5,24	3,14	2,49	2,61	2,87	2,95	3,35	2,15	2,35	2,15	0,95	0,6	0

### Iterasi k=10

$D^{(10)}=$

0	2,7	4,8	5,45	5,57	5,83	5,7	5,3	6,5	5,59	5,79	7,7	∞	∞
2,7	0	2,1	2,75	2,87	3,13	3	2,6	3,8	2,89	3,09	5	∞	∞
4,8	2,1	0	0,65	0,77	1,03	1,63	2,83	2,29	0,79	0,99	3,49	∞	∞
5,45	2,75	0,65	0	0,12	0,38	0,98	2,18	1,64	0,1	0,34	2,84	∞	∞
5,57	2,87	0,77	0,12	0	0,26	0,86	2,06	1,66	0,26	0,46	2,86	∞	∞
5,83	3,13	1,03	0,38	0,26	0	0,6	1,8	1,4	0,52	0,72	2,6	∞	∞
5,7	3	1,63	0,98	0,86	0,6	0	1,2	0,8	1,12	1,32	2	∞	∞
5,3	2,6	2,83	2,18	2,06	1,8	1,2	0	1,2	2,32	2,52	2,4	∞	∞
6,5	3,8	2,29	1,64	1,66	1,4	0,8	1,2	0	1,5	1,7	1,2	∞	∞
5,59	2,89	0,79	0,14	0,26	0,52	1,12	2,32	1,5	0	0,2	2,7	∞	∞
5,79	3,09	0,99	0,34	0,46	0,72	1,32	2,52	1,7	0,2	0	1,2	∞	∞
7,7	5	3,49	2,84	2,86	2,6	2	2,4	1,2	2,7	1,2	0	0,35	∞
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,35	0	0,6
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6	0

### Iterasi k=12

$D^{(12)}=$

0	2,7	4,8	5,45	5,57	5,83	5,7	5,3	6,5	5,59	5,79	6,99	7,34	∞
2,7	0	2,1	2,75	2,87	3,13	3	2,6	3,8	2,89	3,09	4,29	4,64	∞
4,8	2,1	0	0,65	0,77	1,03	1,63	2,83	2,29	0,79	0,99	2,19	2,54	∞
5,45	2,75	0,65	0	0,12	0,38	0,98	2,18	1,64	0,1	0,34	1,54	1,89	∞
5,57	2,87	0,77	0,12	0	0,26	0,86	2,06	1,66	0,26	0,46	1,66	2,01	∞
5,83	3,13	1,03	0,38	0,26	0	0,6	1,8	1,4	0,52	0,72	1,92	2,27	∞
5,7	3	1,63	0,98	0,86	0,6	0	1,2	0,8	1,12	1,32	2	2,35	∞
5,3	2,6	2,83	2,18	2,06	1,8	1,2	0	1,2	2,32	2,52	2,4	2,75	∞
6,5	3,8	2,29	1,64	1,66	1,4	0,8	1,2	0	1,5	1,7	1,2	1,55	∞
5,59	2,89	0,79	0,14	0,26	0,52	1,12	2,32	1,5	0	0,2	1,4	1,75	∞
5,79	3,09	0,99	0,34	0,46	0,72	1,32	2,52	1,7	0,2	0	1,2	1,55	∞
6,99	4,29	2,19	1,54	1,66	1,92	2	2,4	1,2	1,4	1,2	0	0,35	∞
7,34	4,64	2,54	1,89	2,01	2,27	2,35	2,75	1,55	1,75	1,55	0,35	0	0,6
∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0,6	0

### Iterasi k=14

$D^{(14)}=$

0	2,7	4,8	5,45	5,57	5,83	5,7	5,3	6,5	5,59	5,79	6,99	7,34	7,94
2,7	0	2,1	2,75	2,87	3,13	3	2,6	3,8	2,89	3,09	4,29	4,64	5,24
4,8	2,1	0	0,65	0,77	1,03	1,63	2,83	2,29	0,79	0,99	2,19	2,54	3,14
5,45	2,75	0,65	0	0,12	0,38	0,98	2,18	1,64	0,1	0,34	1,54	1,89	2,49
5,57	2,87	0,77	0,12	0	0,26	0,86	2,06	1,66	0,26	0,46	1,66	2,01	2,61
5,83	3,13	1,03	0,38	0,26	0	0,6	1,8	1,4	0,52	0,72	1,92	2,27	2,87
5,7	3	1,63	0,98	0,86	0,6	0	1,2	0,8	1,12	1,32	2	2,35	2,95
5,3	2,6	2,83	2,18	2,06	1,8	1,2	0	1,2	2,32	2,52	2,4	2,75	3,35
6,5	3,8	2,29	1,64	1,66	1,4	0,8	1,2	0	1,5	1,7	1,2	1,55	2,15
5,59	2,89	0,79	0,14	0,26	0,52	1,12	2,32	1,5	0	0,2	1,4	1,75	2,35
5,79	3,09	0,99	0,34	0,46	0,72	1,32	2,52	1,7	0,2	0	1,2	1,55	2,15
6,99	4,29	2,19	1,54	1,66	1,92	2	2,4	1,2	1,4	1,2	0	0,35	0,95
7,34	4,64	2,54	1,89	2,01	2,27	2,35	2,75	1,55	1,75	1,55	0,35	0	0,6
7,94	5,24	3,14	2,49	2,61	2,87	2,95	3,35	2,15	2,35	2,15	0,95	0,6	0

Sesuai dengan perhitungan algoritma *Floyd Warshall* maka diperoleh hasil matriks rute terpendek yaitu dengan melihat bobot matriks  $d_{1,1}$  sampai dengan  $d_{1,14}$ . Dengan urutan terkecil yaitu 0 – 2,7 – 4,8 – 5,3 – 5,45 – 5,57 – 5,59 – 5,7 – 5,79 – 5,83 – 6,5 – 6,99 – 7,34 – 7,94. Bila diurutkan ke dalam bentuk matriks menjadi  $V_1 - V_2 - V_3 - V_8 - V_4 - V_5 - V_{10} - V_7 - V_{11} - V_6 - V_9 - V_{12} - V_{13} - V_{14}$ . Dengan total jarak 13,89 km.





$$w_{ij} = \begin{cases} 0 & \text{jika } i = j \\ \text{bobot edge}(i,j) & \text{jika } i \neq j \text{ dan } (i,j) \in E \\ \infty & \text{jika } i \neq j \text{ dan } (i,j) \notin (E) \end{cases}$$

Matriks berukuran  $14 \times 14$  tersebut kemudian akan diseleksi untuk setiap selnya. Seleksi dilakukan dengan melihat nilai dari  $d_{ij}$  dan  $d_{ik} + d_{kj}$ .  $d_{ij}$  adalah jarak sebenarnya. Jika  $d_{ik} + d_{kj} < d_{ij}$  dengan (k adalah iterasi 1 sampai n) maka  $d_{ij}$  diganti dengan nilai  $d_{ik} + d_{kj}$ . Jika  $d_{ik} + d_{kj} \geq d_{ij}$  maka  $d_{ij}$  bernilai tetap. Iterasi dilakukan sebanyak jumlah *verteks*nya yaitu 14. Dari proses iterasi tersebut didapatkan:

1. Iterasi k=1 didapatkan matriks  $D^{(1)}$  yang memiliki sel sama seperti  $D^{(0)}$  (matriks tidak berubah).
2. Iterasi k=2 didapatkan matriks  $D^{(2)}$  dengan pergantian sel matriks yaitu pada sel  $d_{1,3}, d_{1,6}, d_{1,7}, d_{1,8}, d_{3,1}, d_{3,6}, d_{3,7}, d_{3,8}, d_{6,1}, d_{6,3}, d_{6,8}, d_{7,1}, d_{7,3}, d_{8,1}, d_{8,3}, d_{8,6}$  (perubahan pada sel matriks dapat dilihat pada matriks yang selnya dicetak tebal)
3. Iterasi k=3 didapatkan matriks  $D^{(3)}$  dengan pergantian sel matriks yaitu pada sel  $d_{1,4}, d_{2,4}, d_{4,1}, d_{4,2}, d_{4,6}, d_{4,7}, d_{4,8}, d_{6,4}, d_{7,4}$
4. Iterasi k=4 didapatkan matriks  $D^{(4)}$  dengan pergantian sel matriks yaitu pada sel  $d_{1,5}, d_{1,10}, d_{2,5}, d_{2,10}, d_{3,5}, d_{3,10}, d_{5,1}, d_{5,2}, d_{5,3}, d_{5,7}, d_{5,8}, d_{5,10}, d_{6,10}, d_{7,5}, d_{7,10}, d_{8,5}, d_{8,10}, d_{10,1}, d_{10,2}, d_{10,3}, d_{10,5}, d_{10,6}, d_{10,7}, d_{10,8}$
5. Iterasi k=5 didapatkan matriks  $D^{(5)}$  dengan pergantian sel matriks yaitu pada sel  $d_{1,6}, d_{2,6}, d_{3,6}, d_{4,6}, d_{6,1}, d_{6,2}, d_{6,3}, d_{6,4}, d_{6,8}, d_{6,10}, d_{8,6}, d_{10,6}$
6. Iterasi k=6 didapatkan matriks  $D^{(6)}$  dengan pergantian sel matriks yaitu pada sel  $d_{1,9}, d_{2,9}, d_{3,7}, d_{3,9}, d_{4,7}, d_{4,9}, d_{5,7}, d_{5,9}, d_{7,3}, d_{7,4}, d_{7,5}, d_{7,10}, d_{9,1}, d_{9,2}, d_{9,3}, d_{9,4}, d_{9,5}, d_{10,7}$
7. Iterasi k=7 didapatkan matriks  $D^{(7)}$  dengan pergantian sel matriks yaitu pada sel  $d_{1,9}, d_{2,9}, d_{3,8}, d_{4,8}, d_{5,8}, d_{6,8}, d_{8,3}, d_{8,4}, d_{8,5}, d_{8,6}, d_{8,10}, d_{9,1}, d_{9,2}, d_{10,8}$
8. Iterasi k=8 didapatkan matriks  $D^{(8)}$  yang memiliki sel sama seperti  $D^{(7)}$  (matriks tidak berubah).
9. Iterasi k=9 didapatkan matriks  $D^{(9)}$  dengan pergantian sel matriks yaitu pada sel  $d_{1,12}, d_{2,12}, d_{3,12}, d_{4,12}, d_{5,12}, d_{6,12}, d_{7,12}, d_{8,12}, d_{10,12}, d_{12,1}, d_{12,2}, d_{12,3}, d_{12,4}, d_{12,5}, d_{12,6}, d_{12,7}, d_{12,8}, d_{12,10}$
10. Iterasi k=10 didapatkan matriks  $D^{(10)}$  dengan pergantian sel matriks yaitu pada sel  $d_{1,11}, d_{2,11}, d_{3,9}, d_{3,11}, d_{3,12}, d_{4,9}, d_{4,11}, d_{4,12}, d_{5,11}, d_{6,11}, d_{7,11}, d_{8,11}, d_{9,3}, d_{9,4}, d_{9,11}, d_{11,1}, d_{11,2}, d_{11,3}, d_{11,4}, d_{11,5}, d_{11,6}, d_{11,7}, d_{11,8}, d_{11,9}, d_{12,3}, d_{12,4}$
11. Iterasi k=11 didapatkan matriks  $D^{(11)}$  dengan pergantian sel matriks yaitu pada sel  $d_{1,12}, d_{2,12}, d_{3,12}, d_{4,12}, d_{5,12}, d_{6,12}, d_{10,12}, d_{12,1}, d_{12,2}, d_{12,3}, d_{12,4}, d_{12,5}, d_{12,6}, d_{12,10}$

12. Iterasi  $k=12$  didapatkan matriks  $D^{(12)}$  dengan pergantian sel matriks yaitu pada sel  
 $d_{1.13}, d_{2.13}, d_{3.13}, d_{4.13}, d_{5.13}, d_{6.13}, d_{7.13}, d_{8.13}, d_{9.13}, d_{10.13}, d_{11.13}, d_{13.1}, d_{13.2}, d_{13.3}, d_{13.4}, d_{13.5},$   
 $d_{13.6}, d_{13.7}, d_{13.8}, d_{13.9}, d_{13.10}, d_{13.11}$
13. Iterasi  $k=13$  didapatkan matriks  $D^{(13)}$  dengan pergantian sel matriks yaitu pada sel  
 $d_{1.14}, d_{2.14}, d_{3.14}, d_{4.14}, d_{5.14}, d_{6.14}, d_{7.14}, d_{8.14}, d_{9.14}, d_{10.14}, d_{11.14}, d_{12.14}, d_{14.1},$   
 $d_{14.2}, d_{14.3}, d_{14.4}, d_{14.5}, d_{14.6}, d_{14.7}, d_{14.8}, d_{14.9}, d_{14.10}, d_{14.11}, d_{14.12}$
14. Iterasi  $k=14$  didapatkan matriks  $D^{(14)}$  yang memiliki sel sama dengan  $D^{(13)}$  (matriks tidak berubah)

Setelah selesai melakukan 14 iterasi maka didapatkan bobot dari perhitungan *Floyd Warshall* dari  $d_{1.1}$  hingga  $d_{1.14}$  dengan urutan mulai dari yang terkecil sampai ke yang terbesar. Dengan urutan  $V_1 - V_2 - V_3 - V_8 - V_4 - V_5 - V_{10} - V_7 - V_{11} - V_6 - V_9 - V_{12} - V_{13} - V_{14}$ . dengan titik lokasinya yaitu PT Rama Manggala Gas – PT Mayangkara LPG – Jl. Bengawan Solo, Pakunden, Sukorejo – Jl. Mayang Tengah RT 02/RW 05 Sukorejo, Sukorejo – Jl. Mayang Tengah RT 03/RW 05 Sukorejo, Sukorejo – Jl. Mawar , Sukorejo, Sukorejo – Jl. Kaliporong , Pakunden, Sukorejo – Toko Buwana – Jl. Bakung , Sukorejo , Sukorejo – PT Hariputra Santosa Jaya SPBU no 54.66103 - Jl. Jati GG 1 No 27 RT 02 RW 12 Sukorejo, Sukorejo – Jl Jati No 65 RT 04 RW 12 Sukorejo, Sukorejo – Jl Jati RT 02 RW 14 Sukorejo , Sukorejo. Dengan total jarak yaitu 13,89 km dan waktu tempuh 48 menit.

### 3.2.2. Perhitungan Greedy

Perhitungan Greedy dilakukan setelah mengetahui peta wilayah pendistribusian kemudian digambarkan graf untuk merepresentasikan jarak dan lokasi agar lebih mudah dalam proses perhitungan. Langkah untuk menyelesaikan algoritma *Greedy* yaitu:

1. Menentukan simpul pertama sebagai titik awal . Titik awal dari perhitungan ini adalah PT Rama Manggala Gas sebagai  $V_1$
2. Kemudian dipilih *edge* yang terhubung dengan  $V_1$  yang memiliki jarak terpendek. Yaitu PT Mayangkara LPG.
3. Kemudian terbentuk *edge* dari  $V_1$  ke  $V_2$
4. Selanjutnya dipilih *verteks* yang berdekatan dengan  $(V_1, V_2)$  yang memiliki bobot terkecil
5. Setelah terpilih , lanjutkan langkah 4 sampai semua *verteks* terlewati.

Dari langkah-langkah tersebut didapatkan rute  $V_1 - V_2 - V_3 - V_4 - V_5 - V_6 - V_7 - V_9 - V_8 - V_{12} - V_{13} - V_{14} - V_{11} - V_{10}$ . Jika direpresentasikan ke titik lokasinya yaitu PT Rama Manggala Gas- PT Mayangkara LPG- Jl Galunggung RT 03/02, Sukorejo,

Sukorejo-Jl Mayang Tengah RT 02/05, Sukorejo, Sukorejo- Jl. Mayang Tengah RT 03/RW 02, Sukorejo, Sukorejo- Jl. Bakung, Sukorejo, Sukorejo- Jl. Kaliporong, Pakunden, Sukorejo- PT Hariputra Santosa Jaya SPBU no 54.66103-Jl. Bengawan Solo, Pakunden, Sukorejo- Jl. Jati GG 1 No 27 RT 02/12 Sukorejo, Sukorejo- Jl Jati No 65 RT 04/12 Sukorejo, Sukorejo- Jl. Jati RT 02/14 Sukorejo, Sukorejo- Toko Buwana- Jl. Mawar, Sukorejo, Sukorejo. Dengan total jarak yaitu 13,13 km dan waktu tempuh 45 menit.

#### IV. KESIMPULAN

Diketahui bahwa hasil perhitungan dengan algoritma *Floyd Warshall* dan algoritma *Greedy* didapatkan algoritma *Floyd Warshall* dengan jarak 14,57 km dan algoritma *Greedy* dengan jarak 13,13 km. Sehingga metode yang lebih optimum untuk digunakan dalam pemecahan permasalahan rute terpendek pendistribusian gas elpiji pada PT Petro Jaya Gas adalah menggunakan algoritma *Greedy* karena memiliki jarak yang lebih pendek yaitu dengan melewati PT Rama Manggala Gas- PT Mayangkara LPG- Jl Galunggung RT 03/02, Sukorejo, Sukorejo-Jl Mayang Tengah RT 02/05, Sukorejo, Sukorejo- Jl. Mayang Tengah RT 03/RW 02, Sukorejo, Sukorejo- Jl. Bakung, Sukorejo, Sukorejo- Jl. Kaliporong, Pakunden, Sukorejo- PT Hariputra Santosa Jaya SPBU no 54.66103-Jl. Bengawan Solo, Pakunden, Sukorejo- Jl. Jati GG 1 No 27 RT 02/12 Sukorejo, Sukorejo- Jl Jati No 65 RT 04/12 Sukorejo, Sukorejo- Jl. Jati RT 02/14 Sukorejo, Sukorejo- Toko Buwana- Jl. Mawar, Sukorejo, Sukorejo.

#### DAFTAR PUSTAKA

- [1]. Akhir, T., & Luckitasari, A.. *Optimisasi Rute Kendaraan Untuk Menurunkan Biaya Distribusi Gas Lpg 3 Kg Di Kota Bandung Optimisasi Rute Kendaraan Untuk Menurunkan Biaya Distribusi Gas Lpg 3 Kg Di*.2019
- [2]. Amri, M., Rahman, A., & Yuniarti, R. (n.d.). *PENYELESAIAN VEHICLE ROUTING PROBLEM DENGAN MENGGUNAKAN ( Studi Kasus : MTP Nganjuk Distributor PT . Coca Cola ) NEAREST NEIGHBOR METHOD TO SOLVE VEHICLE ROUTING PROBLEM*.2014.
- [3]. Anggoro, A. A. *Pencarian Titik Lokasi Dengan Pemanfaatan Algoritma Floyd-Warshall Sebagai Perhitungan Jarak Terdekat Di Institut Teknologi Bandung . Bandung*:2015. STMIK LPKIA.
- [4]. Darmawan, R., & Ghaniy, R. Analisa dan Penerapan Algoritma Floyd Warshal. *Ilmiah Teknologi*, 8(November),2018. 67–78.  
<https://media.neliti.com/media/publications/288943-analisa-dan-penerapan-algoritma-floyd-wa-907349ac.pdf>
- [5]. Jakaria, & R. *Implementasi Algoritma Greedy pada Metode Transportasi dengan Menggunakan VAM dalam Pendistribusian Produk*.2017.

- [6]. Mukti, M. & M. R. Menentukan Rute Terpendek Dengan Menggunakan Algoritma Floyd-Warshall Dalam Pendistribusian Barang Pada Pt. Rapy Ray Putratama. *KARISMATIKA: Kumpulan Artikel Ilmiah, Informatika, Statistik, Matematika Dan Aplikasi*, 4(1).2018. <https://doi.org/10.24114/jmk.v4i1.11857>
- [7]. Purnomo, A. C., Yuliana, M., Mt, S. T., Prasetyaningrum, I., & Mt, S. S. (n.d.). Implementasi Algoritma Greedy Pada Layanan Taksi Wisata Berbasis Web. *Institut Teknologi Sepuluh Nopember, Kampus ITS*. 2011.
- [8]. Subakdo, W. A., & Nugroho, Y. A. In-Bound dan Out-Bound Logistic pada Distribusi LPG 3KG di Indonesia. *Jurnal Universitas Muhammadiyah Jakarta, November*. 2016.1–10.