OPTIMASI JALUR DISTRIBUSI GAS LPG DARI AGEN PT. CAS KE PANGKALAN DENGAN PENDEKATAN ALGORITMA DJIKSTRA

Jazuli[1], Dwi Nurul Izzhati[2], dan Nelli Rokhmah[3]

1. Staf Pengajar Program Studi Teknik Industri, Universitas Dian Nuswantoro Semarang
2. Staf Pengajar Program Studi Teknik Industri, Universitas Dian Nuswantoro Semarang
3. Mahasiswa Program Studi Teknik Industri, Universitas Dian Nuswantoro Semarang

Kontak Person:
Jl. Nakula I No. 5-11 Semarang 50131
Telp: 024-3555628, E-mail: jazuli@dsn.dinus.ac.id

Abstrak

Penelitian ini membahas tentang optimasi jalur transportiasi atau distribusi LPG dari Agen ke Pangkalan untuk mendapatkan rute tercepat dan terpendek. Pendekatan yang digunakan untuk optimasi rute transportasi distribusi ini menggunakan Algoritma Djikstra. Ada tiga kelompok distribusi yang dilakukan untuk menjangkau seluruh pangkalan yang ada. Dari hasil iterasi Algoritma Djikstra didapatkan kelompok A jarak sebesar 10,4 km 36 menit, kelompok B jarak sebesar 63,5 km, waktu 146 menit, kelompok C jarak 20,1 km, waktu 60 menit, atau mampu memangkas jarak sebesar 9,1 km dan waktu 32 menit.

Kata Kunci : optimasi, transportasi, distribusi, algoritma djikstra

PENDAHULUAN

Jaringan distribusi dan transportasi memungkinkan produk berpindah dari penjual ke konsumen yang sering kali dibatasi oleh jarak. Distribusi dan transportasi yang baik merupakan hal penting agar produk dapat dikirim sampai ke konsumen secara tepat waktu, tepat pada tempat yang telah dilakukan dan produk dalam kondisi baik (Muhammad dan Rahmi, 2017). Aktivitas distribusi suatu produk dilakukan dengan menyusun jadwal dan juga menetukan rute pengiriman. Penentuan rute merupakan keputusan pemilihan jalur terbaik sebagai upaya pelayanan konsumen. Keputusan penentuan jadwal serta rute pengiriman menjadi sesuatu yang penting dalam rangka meminimumkan biaya pengiriman, meminimumkan waktu atau jarak tempuh (Pujawan dan Mahendrawati, 2010). Ada berbagai faktor yang dapat digunakan dalam menentukan rute. Hal ini berkaitan dengan parameter aksesibilitas suatu lokasi tujuan dari lokasi asal. Faktor-faktor yang dapat digunakan untuk menentukan rute terbaik adalah jarak terpendek, waktu tercepat, dan biaya termurah (Muslim, 2005). Saat ini strategi pengiriman yang dilakukan PT. CAS adalah dengan cara membagi beberapa pangkalan yang ia punya menjadi 3 kelompok pangkalan, namun dalam 3 kelompok tersebut masih acak-acakan atau tidak sesuai lokasi dengan jarak terdekat. Berikut ini daftar nama-nama pemilik pangkalan beserta alamatnya dan juga jarak gudang ke masing-masing pangkalan. Jadi perusahaan mengeluarkan biaya distribusi per hari dengan total Rp. 430.000,-. Dengan memilih rute terbaik tentunya dapat menghemat biaya distribusi yang dikeluarkan oleh perusahaan. Pemilihan rute terbaik akan membuat aktivitas distribusi produk lebih efektif dan juga efisien. Rute terbaik adalah dengan rute dengan jarak terpendek, yang tentumya akan mempengaruhi biaya transportasi yang terjadi. Jarak tempuh kendaraan yang lebih pendek berarti biaya transportasi yang lebih rendah (Sarjono, 2014). Untuk itu diperlukan sebuah sistem untuk membantu melakukan pencarian lintasan terpendek yang dapat merepresentasikan data (Munir, 2003). Data tersebut dapat disimpan, diolah dan disajikan dalam bentuk yang lebih sederhana sehingga memudahkan dalam menentukan lintasan terpendek. Algoritma Djikstra yang juga merupakan algoritma Greedy bisa menentukan lintasan terpendek.

Pada penelitian ini akan dibahas tentang penentuan rute distribusi gas LPG dari PT. CAS kepada pangkalan yang ada dalam naungannya dengan pendekatan algoritma Djikstra untuk mendapatkan biaya transportasi distribusi yang lebih kecil dibandingkan dengan kondisi sebelumnya.

LANDASAN TEORI

Travelling Salesman Problem (TSP)

TSP merupakan salah satu permasalahan optimasi klasik yang sulit untuk dipecahkan secara konvensional. Penyelesaian eksak terhadap persoalan ini akan melibatkan algoritma yang mengharuskan mencari kemungkinan semua solusi yang ada. Sehingga akan terjadi ledakan kombinasi dan membuat kompleksitas waktu dari eksekusi algoritma sangat tinggi. Masalah TSP melibatkan seorang sales yang harus melakukan kunjungan ke sejumlah kota dalam menjajakan produknya. Rangkaian kota-kota yang dikunjungi harus membentuk suatu jalur sedemikian sehingga kota-kota tersebut hanya boleh dilewati tepat satu kali dan kemudian kembali lagi ke kota awal. Penyelesaian terhadap permasalahan TSP ini adalah untuk memperoleh jalur terpendek. Penyelesaian eksak terhadap masalah TSP mengharuskan untuk melakukan perhitungan terhadap semua kemungkinan rute yang dapat diperoleh, kemudian memilih salah satu rute yang terpendek (Puspitorini, 2008)

Algoritma Djikstra

Algoritma Dijkstra ditemukan oleh Edsger W. Dijkstra yang merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi dan bersifat sederhana. Algoritma ini menyelesaikan masalah mencari sebuah lintasan terpendek (sebuah lintasan yang mempunyai panjang minimum) dari vertex a ke vertex z dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif, namun jika terjadi demikian, maka penyelesaian yang diberikan adalah infiniti.

Secara umum, sebelum melakukan iterasi, algoritma sudah mengidentifikasikan jarak terdekat dari i-1 vertex terdekatnya. Selama seluruh edge berbobot tertentu yang (positif), maka vertex terdekat berikutnya dari node asal dapat ditemukan selama vertex berdekatan dengan vertex T_i. Kumpulan vertex yang berdekatan dengan vertex di Ti dapat dikatakan sebagai “fringe vertices”. Vertex inilah yang merupakan kandidat dari Algoritma Dijkstra untuk memilih vertex berikutnya dari node asal (Saputra, 2011).

Algoritma Dijkstra mempunyai sifat sederhana (*straightforward*). Sesuai dengan arti greedy yang secara harfiah berarti tamak atau rakus, namun tidak dalam konteks negatif, Algortima Greedyini hanya memikirkan solusi terbaik yang akan diambil pada setiap langkah tanpa memikirkan konsekuensi kedepan.

Elemen-Elemen Penyusun Greedy dalam Algoritma Dijkstra:

1. Himpunan Kandidat

Himpunan ini berisi elemen-elemen yang memiliki peluang untuk membentuk solusi. Pada persoalan lintasan terpendek dalam graf, himpunan kandidat ini adalah himpunan simpul pada graf tersebut.

2. Himpunan Solusi

Himpunan ini berisi solusi dari permasalahan yang diselesaikan dan elemennya terdiri dari elemen dalam himpunan kandidat namun tidak semuanya atau dengan kata lain himpunan solusi ini adalah bagian dari himpunan kandidat.

3. Fungsi Seleksi

Fungsi Seleksi adalah fungsi yang akan memilih setiap kandidat yang memungkinkan untuk menghasilkan solusi optimal pada setiap langkahnya.

Fungsi Kelayakan

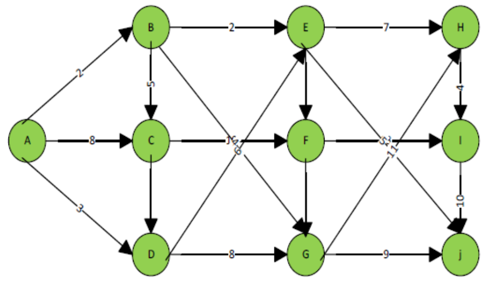
Fungsi kelayakan akan memeriksa apakah suatu kandidat yang telah terpilih (terseleksi) melanggar constraint atau tidak. Apabila kandidat melanggar constraint maka kandidat tidak akan dimasukkan ke dalam himpunan solusi.

4. Fungsi Objektif

Fungsi Objektif akan memaksimalkan atau meminimalkan nilai solusi. Tujuannya adalah memilih satu saja solusi terbaik dari masing-masing anggota himpunan solusi.

Jika menggunakan Algoritma Dijkstra untuk menentukan jalur terpendek dari suatu graph maka pasti akan menemukan jalur yang terbaik. Karena pada waktu penentuan jalur yang akan terpilih, akan dianalisis bobot dari node yang belum terpilih. Lalu dipilih node dengan bobot yang terkecil, jika ternyata ada bobot yang lebih kecil jika melalui node tertentu maka bobot dapat berubah. Algoritma ini akan berhenti jika semua node sudah terpilih dan dengan Algoritma Dijkstra ini dapat menemukan jarak terpendek dari node yang dipilih.

Persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi. Graf yang di gunakan dalam mencari lintasan terpendek dalam graph berbobot. Bobot pada sisi graph dapat menyatakan jarak antar kota, waktu, biaya dan sebagainya. Dalam hal ini bobot harus bernilai positif seperti Gambar 1 berikut:



Gambar 1. Algoritma Djikstra

METODE PENELITIAN

langkah per langkah penelitian pencarian jalur terpendek secara rinci dimulai dari node awal sampai node tujuan dengan nilai jarak terkecil.

Algoritma Djikstra untuk mencari *path* terpendek adalah sebagai berikut :

1. Inisialisasi : L = {} ; V = {v_2, v_3, ..., v_n}
2. Untuk i =2, ..., n, lakukan D(i) = W(1,i)
3. Selama v_n€ L (vn belum merupakan titik permanen), lakukan :
   a. Pilih titik v_k € V – L (titik tidak permanen) dengan D(k) terkecil.
   L = L U {v_k} (jadikan $v_k$menjadi titik permanen)
   b. Untuk setiap v_j€ V – L lakukan :
   Jika D(k) + W(k,j) < D(j) maka ganti D(j) dengan D(k) + W (k,j)

Menurut algoritma diatas, jalur terpendek dari titik v_1 ke vn adalah melalui titik-titik dalam L secara berurutan, dan jumlah bobot *path* terkecilnya adalah D(n).

Keterangan :

V(G) = {v_1, v_2, ..., v_n}.

L = Himpunan titik-titik € v(G) yang sudah terpilih (titik permanen) dalam jalur *path* terpendek.

D (j) = Jumlah bobot *path* terkecil dari v_1 ke v_j.

W(i,j) = Bobot garis dari titik v_i ke titik v_j.

W*(1,j) = Jumlah bobot *path* terkecil dari v_1 ke v_j.

Mengolah data menggunakan algoritma DjikstraMembuat matriks *m x n*

Inialisasi dengan $v = \{v_1, v_2, ..., v_n\}$ menggunakan iterasi dari 0 sampai n. Dimana v merupakan node atau masing-masing pangkalan.

Contoh matriks :

$$W = \begin{array}{c|ccccccc|} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \\ v_1 & 100 & 200 & 300 & ... & ... & ... & ... \\ v_2 & ... & ... & ... & ... & ... & ... & ... \\ v_3 & ... & ... & ... & ... & ... & ... & ... \\ v_4 & ... & ... & ... & ... & ... & ... & ... \\ v_5 & ... & ... & ... & ... & ... & ... & ... \\ v_6 & ... & ... & ... & ... & ... & ... & ... \\ v_7 & ... & ... & ... & ... & ... & ... & ... \end{array}$$

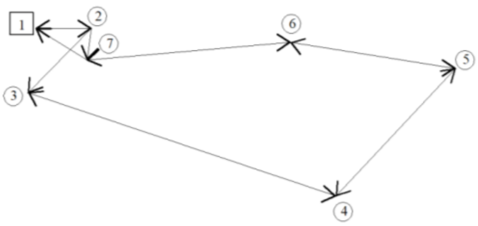
HASIL PENELITIAN DAN PEMBAHASAN

Algoritma yang ditemukan oleh Djikstra merupakan algoritma untuk mencari jalur terpendek untuk transportasi/distribusi antara 2 titik. Misalkan G adalah graf berlabel (berarah atau tidak berarah) dengan titik-titik V(G) = $\{v_1, v_2, ..., v_n\}$ dan *path* terpendek yang dicari adalah dari v_1 ke v_n. Algoritmna Djikstra dimulai dari titik v_1. Dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang sudah dipilih disebut titik permanen, dan titik-titik tersebut tidak dilibatkan lagi dalam iterasi berikutnya.

Langkah-langkah untuk menyelesaikan permasalahan mencari rute terpendek menggunakan algoritma djikstra akan dijelaskan pada lampiran 2.

Berikut ini penyelesaian algoritma djikstra untuk masing-masing kelompok :

a. Kelompok A



Gambar 2. Pola Rute Kelompok A

Keterangan :

1	Ariloka (Gudang)
2	Ariloka
3	Kenconowungu
4	Penaton
5	Kelengan
6	Abimanyu
7	Wiroto

Matriks hubung W untuk menyatakan graf pada Gambar 2 adalah sebagai berikut :

$$W = \begin{array}{c|ccccccc|} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \\ v_1 & \infty & 0{,}6 & 0{,}7 & \infty & \infty & \infty & \infty \\ v_2 & \infty & \infty & 1{,}2 & \infty & \infty & \infty & 0{,}4 \\ v_3 & \infty & \infty & \infty & 3{,}5 & \infty & \infty & \infty \\ v_4 & \infty & \infty & 2{,}8 & \infty & 3{,}4 & \infty & \infty \\ v_5 & \infty & \infty & \infty & 20 & \infty & 4{,}1 & \infty \\ v_6 & \infty & \infty & \infty & \infty & \infty & 1{,}8 & \infty \\ v_7 & 0{,}65 & 0{,}4 & \infty & \infty & \infty & 2{,}2 & \infty \end{array}$$

Proses Iterasi Penyelesaian Masalah Kelompok A

I = 0

Mula-mula L = { } dan V = $\{v_2, v_3, ..., v_7\}$.

D(2) = W(1,2) = 0,6 D(3) = W(1,3) = 0,7

D(4) = W(1,4) = ∞ D(5) = W(1,5) = ∞

D(6) = W(1,6) = ∞ D(7) = W(1,7) = 0,65

Pada iterasi pertama, D(1) = 0 dan diambil sebagai titik permanen pertama, L = $\{v_1\}$. Iterasi selengkapanya ada pada tabel 4.8.

I = 1

Harga D(j) yang mungkin berubah adalah pada titik-titik yang dapat dicapai secara langsung dari titik permanen terakhir (dalam iterasi ini adalah dari titik v_1 dengan nilai D(1) = 0). Titik yang dapat dicapai langsung dari $v_2$dan v_3.

D(2) adalah harga minimum dari nilai D(2) sebelumnya (=∞) dibandingkan dengan D (titik permanen sebelumnya) + jarak dari titik permanen sebelumnya ke titik v_2 secara langsung = min (∞, D(1) + W (1,2)) = min (∞, 0 + 0,6) = 0,6.

Secara analog, D(3) adalah harga minimum dari nilai D(3) sebelumnya (=∞) dibandingkan dengan D (titik

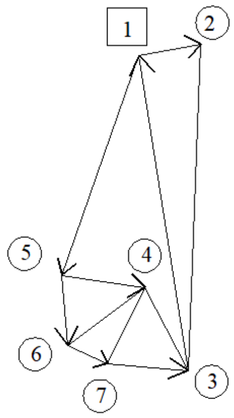
permanen sebelumnya) + jarak dari titik permanen sebelumnya ke titik v_3 secara langsung = min (∞, D(1) + W (1,3)) = min (∞, 0 + 1,2) = 1,2.
Jadi diperoleh nilai baru D(2) = 0,6 dan D(3) = 1,2. Nilai D(j) yang lain tetap karena tidak dapat dicapai secara langsung dari$v_{1.}$ Nilai D(j) terkecil pada iterasi ini adalah D(2) = 0,6. Maka titikv_2 dijadikan titik permanen untuk iterasi berikutnya.
I = 2
Pada iterasi ini titik permanen yang didapat sebelumnya adalah v_2 dengan D(2) = 0,6. Titik yang dapat dicapai secara langsung dari titik permanen v_2 adalah v_3(dengan D(3) = 0,7) dan v_7 (dengan D(7) = 0,65).
Maka
D(3) = min (∞, D(2) + W (1,3)) = min (∞, 0,6 + 1,2) = 1,8
D(7) = min (0,65, D(2) + W (1,7)) = min (0,65, 0,6 + 0,4) = 1
Pada iterasi nilai D(j) yang paling minimum adalah D(7)= 1. Maka titil v_7 kita jadikan titik permanen pada iterasi berikutnya.
I = 3
Pada iterasi sebelumnya, titik permanen adalah v_7 dengan D(7) = 1. Titik yang dapat dicapai langsung dari v_7 hanya v_6.
Maka :
D(6) = min (∞, 1 + W(7,6)) = min (∞, 1+2,2) = 3,2
Pada iterasi ini harga D(j) minimum adalah D(6) = 3,2.
I = 4
Pada iterasi sebelumnya, titik permanen adalah v_6 dengan D(6) = 3,2. Titik yang dapat dicapai langsung dari v_6 hanya v_5.
Maka :
D(5) = min (∞, 3,2 + W(6,5)) = min (∞, 3,2+1,8) = 5
Pada iterasi ini harga D(j) minimum adalah D(5) = 5.
I = 5
Pada iterasi sebelumnya, titik permanen adalah v_5 dengan D(5) = 5. Titik yang dapat dicapai langsung dari v_5 hanya v_4.
Maka :
D(4) = min (∞, 5 + W(5,4)) = min (∞, 5+2) = 7
Pada iterasi ini harga D(j) minimum adalah D(4) = 7.
I = 6
Satu-satunya titik yang belum permanen adalah $v_{3.}$
Maka :
D(3) = min (1,2, 7 + W(4,3)) = min (1,2, 7+2,8) = 9,8
Pada iterasi ini harga D(j) minimum adalah D(3) = 9,8.
I = 7
Disini iterasi dihentikan karena satu-satunya titik yang tersisa adalah v_3 (titik permanen sebelumnya), sehingga otomatis akan menyambung ke titik semula yaitu v_1 dengan nilai minimum :
D(1) = min (0, 9,8 + W(3,1)) = min (0, 9,8+0,7) =10,5
Jadi, jarak minimum dari titik v_1 kembali ke titik v_1 adalah 10,5. Perhatikan bagaimana iterasi ini dikerjakan dalam Tabel 1 dibawah ini.

Tabel 1 Penyelesaian iterasi Kelompok

I	D(a)	D(b)	D(c)	D(d)	D(e)	D(f)	D(z)	L
1	-	= min (∞, 0 + 0,6) = 0,6	= min (∞, 0 + 1,2) = 1,2	∞	∞	∞	∞	{v1, v2}
2	-	-	1,2	∞	∞	∞	= min (0,65, 0,6 + 0,4) = 1	{v1, v2, v7}
3	-	-	-	∞	∞	= min (∞, 1+2,2) = 3,2	-	{v1, v2, v7. v6}
4	-	-	-	∞	= min (∞, 3,2+1,8) = 5	-	-	{v1, v2, v7. v6, v5}
5	-	-	-	= min (∞, 5+2) = 7	-	-	-	{v1, v2, v7. v6, v5, v4}
6	-	-	= min (1,2, 7+2,8) = 9,8	-	-	-	-	{v1, v2, v7. v6, v5, v4, v3}
7	= min (0, 9,8+0,7) =10,5	-	-	-	-	-	-	{v1, v2, v7. v6, v5, v4, v3, v1}

Dari penyelesaian diatas didapatkan jarak terdekat dari Ariloka (Gudang) - Ariloka – Wiroto – Abimanyu – Kelengan – Penaton – Kenconowungu – Ariloka (Gudang) dengan jarak yang ditempuh 10,4 km dan waktu yang ditempuh 36 menit.

b. Kelompok B



Gambar 3. Pola Rute Kelompok B

Keterangan :

1	Ariloka (Gudang)
2	Purwosari
3	Karangmalang
4	Jatibarang
5	Wonolopo
6	Graha Pesona
7	Tambangan

Matriks hubung W untuk menyatakan graf pada Gambar 3 adalah sebagai berikut :

$$
W = \begin{array}{c|ccccccc|}
 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \\
v_1 & \infty & 4 & 18 & \infty & \infty & \infty & \infty \\
v_2 & 5,5 & \infty & 21 & 17 & \infty & \infty & \infty \\
v_3 & 18 & \infty & \infty & 6 & \infty & \infty & 5,2 \\
v_4 & \infty & \infty & 6 & \infty & 6 & \infty & \infty \\
v_5 & 18 & \infty & \infty & 7 & \infty & 4,5 & \infty \\
v_6 & 21 & \infty & \infty & \infty & 4,5 & \infty & 2,8 \\
v_7 & \infty & \infty & 5,2 & \infty & \infty & 2,6 & \infty
\end{array}
$$

Proses Iterasi Penyelesaian Masalah Kelompok B mengikuti alur langkah seperti iterasi pada Kelompok A yang meghasilkan Tabel XX sebagai penyelesaian masalahnya.

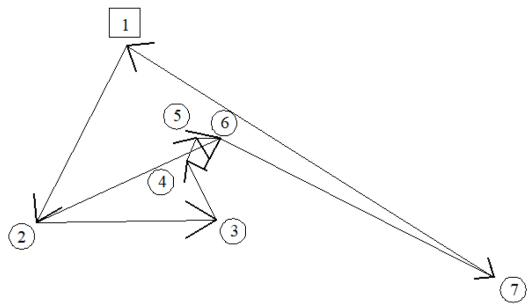
Jadi, jarak minimum dari titik v_1 kembali ke titik v_1 adalah 55,5. Perhatikan bagaimana iterasi ini dikerjakan dalam Tabel 2. dibawah ini.

Tabel 2. Penyelesaian iterasi Kelompok B

I	D(a)	D(b)	D(c)	D(d)	D(e)	D(f)	D(z)	L
1	-	= min (∞, 0 + 4) = 4	∞	∞	= min (∞, 0 + 18) = 18	∞	∞	{v1, v2}
2	-	-	∞	= min (∞, 4 + 17) = 21	18	∞	∞	{v1, v2, v5}
3	-	-		21	-	= min (∞, 18+4,5) = 22,5	∞	{v1, v2, v5, v4}
4	-	-	= min (∞, 21+6) = 27	-	-	22,5	∞	{v1, v2, v5, v4, v6}
5	-	-	-	-	-	-	= min (∞, 27+2,8) = 29,3	{v1, v2, v5, v4, v6, v7}
6	-	-	= min (∞, 29,3+5,2) = 34,5	-	-	-	-	{v1, v2, v5, v4, v6, v7, v3}
7	= min (0, 30,5+21) = 51,5	-	-	-	-	-	-	{v1, v2, v5, v4, v6, v7, v3, v1}

Dari penyelesaian tabel diatas didapatkan jarak terpendek dari Ariloka (Gudang) – Purwosari – Wonolopo – Jaibarang - Tambangan – Graha Pesona – Karangmalang – Ariloka (Gudang) dengan jarak 63,5 km dan waktu yang ditempuh 146 menit.

c. Kelompok C



Gambar 4 Pola Rute Kelompok C

Keterangan :

1	Ariloka (Gudang)
2	Gisikdrono
3	Lemahgempal
4	Bulustalan
5	Jayengan
6	Jayengan
7	Wonodri

Matriks hubung W untuk menyatakan graf pada Gambar 4 adalah sebagai berikut :

$$
W = \begin{array}{c|ccccccc|}
 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \\
v_1 & \infty & 3{,}3 & \infty & \infty & \infty & \infty & 6{,}7 \\
v_2 & 4{,}2 & \infty & 5{,}1 & \infty & \infty & 6 & \infty \\
v_3 & 3{,}4 & 2{,}9 & \infty & 1{,}3 & \infty & \infty & 4{,}4 \\
v_4 & \infty & \infty & 1 & \infty & 0{,}4 & \infty & \infty \\
v_5 & \infty & \infty & \infty & 0{,}4 & \infty & 0{,}4 & \infty \\
v_6 & 4{,}1 & \infty & \infty & \infty & 0{,}4 & - & 5{,}1 \\
v_7 & 6{,}7 & \infty & 5 & \infty & \infty & 6 & \infty
\end{array}
$$

Proses Iterasi Penyelesaian Masalah Kelompok C mengikuti urutan seperti langkah Pada Kelompok A dan B yang menghasilkan solusi seperti ditunjukkan pada Tabel XX. Jadi, jarak minimum dari titik v_1 kembali ke titik v_1 adalah 14,5. Perhatikan bagaimana iterasi ini dikerjakan dalam Tabel 3 dibawah ini.

Tabel 3. Penyelesaian Iterasi Kelompok C
Sumber : Olah Data, 2018

I	D(a)	D(b)	D(c)	D(d)	D(e)	D(f)	D(z)	L
1	-	= min (∞, 0 + 3,3) = 3,3.					= min (∞, 0 + 6,7) = 6,7	{v1, v2}
2			= min (∞, 3,3 + 5,1) = 8,4			= min (∞, 3,3 + 6) = 9,3	6,7	{v1, v2, v7}
3			8,4			9,3		{v1, v2, v7, v6}
4			-		= min (∞, 9,3 + 0,4) = 9,7			{v1, v2, v7, v6, v5}
5				= min (∞, 9,7+0,4) = 10,1				{v1, v2, v7, v6, v5, v4}
6			= min (∞, 10,1+1) = 11,1					{v1, v2, v7, v6, v5, v4, v3}
7	= min (0, 11,1+3,4) = 14,5							{v1, v2, v7, v6, v5, v4, v3, v1}

Dari hasil penyelesaian tabel diatas didapatkan jarak terpendek dari Ariloka (Gudang) – Gisikdrono – Wonodri – Jayengan – Jayengan – Bulustalan – Lemahgempal – Ariloka (Gudang) dengan jarak 20,1 km dan waktu yang ditempuh 60 menit.

KESIMPULAN

Algoritma Djikstra yang digunakan pada penelitian ini mendapatkan nilai pada kelompok A jarak sebesar 10,4 km 36 menit, kelompok B jarak sebesar 63,5 km, waktu 146 menit, kelompok C jarak 20,1 km, waktu 60 menit,. Dari perbandingan antara kondisi riil dengan Algoritma Djikstra didapatkan bahwa hasil rute yang paling optimal pada algoritma djikstra yang mampu menurunkan nilai lebih banyak dari kondisi saat ini. Dengan nilai, selisih jarak sebesar 9,1 km, waktu 32 menit

DAFTAR PUSTAKA

Muhammad, B., dan Rahmi, 2017. “Penentuan Distribusi Sirup untuk Meminimalkan Biaya Transportasi”. *Jurnal Teknik Industri Universitas Malikulassaleh.*

Muslim, 2005. “Aplikasi Penentuan Rute Terbaik Berbasis Sistem Informasi Geografis”. *Jurnal Ilmiah Teknologi Informasi DINAMIK*, Vol X.

Pujawan, I. N., dan Mahendrawathi E.R., 2010., *Supply Chain Management. Edisi 2*. Surabaya: Guna Widya.

Sarjono, 2014. “Determination of Best Route to Minimize Transportation Costs Using NearestNeighbor Procedure”. *Applied Mathematical Sciences*, 8(62), 3063–3074.

Munir. R.**,** 2003.**,** *Deskripsi Sistem. Edisi ke-2*. Bandung: Informatika.

Puspitorini, S., 2008., “Penyelesaian Masalah Travelling Salesman Problem Dengan Jaringan Saraf Self Organizing”, *Media Informatika* Vol.6 No.1, 39-55

Saputra. R., ”Sistem Informasi Pencarian Obyek Kota Medan Dengan Algoritma Dijkstra.” *Jurnal Matematika*, No. 1, Vol.14, Hal 19-24.