

## ANALISIS ALGORITMA *FLOYD WARSHALL* UNTUK MENENTUKAN LINTASAN TERPENDEK PENGANGKUTAN SAMPAH (Studi Kasus: Pengangkutan Sampah di Kabupaten Kubu Raya)

Vega Setiawan, Mariatul Kiftiah, Woro Budiartini Partwi

### INTISARI

*Sampah adalah sesuatu yang tidak digunakan, tidak dipakai, tidak disenangi atau sesuatu yang dibuang yang berasal dari kegiatan manusia dan tidak terjadi dengan sendirinya. Kubu Raya merupakan salah satu kabupaten dengan intensitas sampah yang cukup tinggi. Jumlah penduduk yang semakin meningkat tentu menjadi salah satu penyebab semakin banyaknya jumlah sampah yang ada di kabupaten tersebut. Salah satu Algoritma yang digunakan untuk menentukan lintasan terpendek untuk kasus pengangkutan sampah di Kabupaten Kubu Raya adalah Algoritma Floyd Warshall. Algoritma Floyd Warshall membandingkan semua kemungkinan lintasan pada graf untuk setiap sisi dari semua simpul. Tujuan dari penelitian ini adalah untuk menerapkan Algoritma Floyd Warshall dalam penentuan lintasan terpendek pengangkutan sampah dari TPS ke TPA di Kabupaten Kubu Raya. Data yang digunakan pada penelitian ini berupa data sekunder tentang lintasan pengangkutan sampah dari 10 TPS ke TPA yang ada di Kabupaten Kubu Raya. Simpul merepresentasikan TPS yang ada di Kabupaten Kubu Raya, sedangkan sisi merepresentasikan lintasan yang menghubungkan antar TPS dan TPS ke TPA. Sedangkan bobot merepresentasikan jarak (km) antar 10 TPS dan jarak antara TPS dan TPA. Hasil dari penelitian ini adalah diperoleh lintasan terpendek pengangkutan sampah yaitu  $38,7 \text{ km}$  dengan lintasan  $v_1 - v_2 - v_3 - v_7 - v_8 - v_9 - v_6 - v_5 - v_4 - v_{10} - v_{11}$ .*

**Kata kunci:** *Floyd Warshall, TPS, TPA.*

### PENDAHULUAN

Menurut *World Health Organization* (WHO), sampah adalah sesuatu yang tidak digunakan, tidak dipakai, tidak disenangi atau sesuatu yang dibuang yang berasal dari kegiatan manusia dan tidak terjadi dengan sendirinya [1]. Sampah menyebabkan persoalan yang tak kunjung teratasi, salah satu contohnya adalah permasalahan sampah yang ada di Kabupaten Kubu Raya.

Sebagai salah satu kabupaten terluas di Kalimantan Barat, Kubu Raya menjadi salah satu kabupaten dengan intensitas sampah yang cukup tinggi. Pada tahun 2015, Kubu Raya tercatat sebagai kabupaten dengan jumlah penduduk terbesar kedua setelah Kota Pontianak, yaitu 545.409 jiwa dan tidak dapat dipungkiri jumlah tersebut terus bertambah setiap tahunnya [2]. Jumlah penduduk yang semakin meningkat tentu menjadi salah satu penyebab semakin banyaknya jumlah sampah yang ada di kabupaten tersebut. Tercatat pada tahun 2017 rata-rata volume sampah perTPS dalam satu hari adalah 10 meter kubik dan daya angkut maksimum satu *dump truck* (truk sampah) adalah 3,2 meter kubik [3]. Bertambahnya jumlah sampah pasti menimbulkan banyak dampak negatif, seperti bau tidak sedap dari sampah tersebut, menimbulkan penyakit, berdampak pada terjadinya banjir, hingga dapat menyebabkan kemacetan yang terjadi apabila hingga pagi sampah masih menumpuk di Tempat Pembuangan sampah Sementara (TPS) dan belum diangkut ke Tempat Pembuangan Akhir (TPA).

Keterlambatan dalam pengangkutan sampah dari TPS ke TPA berpotensi menimbulkan kemacetan, selain itu pihak pengangkut sampah menjadi tidak optimal dalam mengatur penggunaan armadanya, dan pada akhirnya bermuara pada kredibilitas pengangkut sampah tersebut. Pada masalah ini diperlukan suatu cara untuk menangani permasalahan tersebut agar pengangkutan sampah dari TPS ke TPA yang ada di Kubu Raya dapat lebih optimal. Pengangkutan sampah dari

TPS ke TPA lebih optimal bila pengangkutan tersebut tidak memerlukan banyak waktu, dan waktu yang diperlukan untuk pengangkutan semakin cepat dengan jalur pengangkutan yang pendek. Penentuan jalur terpendek dapat diselesaikan secara sistematis. Ilmu Matematika yang dapat menyelesaikan permasalahan penentuan lintasan terpendek adalah graf. Graf biasa digunakan untuk penentuan lintasan terpendek. Dalam penelitian ini, penentuan lintasan terpendek digunakan untuk menyelesaikan masalah pengangkutan sampah dari TPA ke TPS, menggunakan Algoritma *Floyd Warshall* [4].

Perhitungan jarak pengangkutan sampah dari TPS ke TPS yang ada di Kubu Raya masih menggunakan perhitungan manual, sehingga memungkinkan terjadinya keterlambatan pengangkutan sampah dari TPA ke TPS yang ada di Kubu Raya. Oleh karena itu, dalam penelitian ini dibahas solusi matematis untuk dapat menyelesaikan permasalahan tersebut. Hasilnya diharapkan dapat menjadi masukan untuk pihak Dinas Cipta Karya Tata Ruang dan Kebersihan dalam menyelesaikan masalah pengangkutan sampah dari TPS ke TPA di Kubu Raya.

Penelitian ini bertujuan untuk mengkaji langkah-langkah penentuan lintasan terpendek menggunakan Algoritma Floyd Warshall dan menerapkan Algoritma Floyd Warshall untuk menentukan lintasan terpendek dalam pengangkutan sampah dari 10 TPS ke TPA yang ada di Kabupaten Kubu Raya. Pada penelitian ini satu *dump truck* maksimal dapat mengangkut sampah di tiga TPS, yang akan dilanjutkan oleh *dump truck* berikutnya mulai dari TPS ke empat pada lintasan terpendek yang terbentuk. Begitu selanjutnya hingga sampah yang ada di 10 TPS diangkut ke TPA.

Langkah pertama penentuan jalur terpendek menggunakan Algoritma *Floyd Warshall* adalah mengumpulkan data jumlah TPS, jarak antar 10 TPS yang ada di Kabupaten Kubu Raya. Langkah kedua adalah menginterpretasikan data tersebut ke dalam graf, dengan menjadikan TPS sebagai simpul, lintasan yang menghubungkan antar TPS sebagai sisi dan panjang jalan antar simpul sebagai bobot. Setelah itu, langkah selanjutnya adalah graf berbobot yang sudah terbentuk diinterpretasikan ke dalam matriks bobot berukuran  $n \times n$  (dengan  $n$  adalah jumlah simpul). Kemudian menentukan matriks pendahulu yang merupakan matriks awal dari representasi permasalahan yang diambil. Setelah itu dilakukan iterasi sebanyak  $n$  simpul untuk menentukan bobot lintasan terpendek. Penentuan bobot lintasan terpendek adalah nilai dari setiap elemen yang diperoleh dari matriks bobot. Sedangkan lintasan terpendek dari simpul awal ke simpul tujuan diperoleh dari matriks pendahulu.

## ALGORITMA FLOYD WARSHALL

*Algoritma Floyd Warshall* berfungsi untuk menyelesaikan masalah lintasan terpendek. Algoritma ini membandingkan semua kemungkinan lintasan pada graf untuk setiap sisi dari semua simpul. Selain itu, algoritma ini juga menerapkan penggunaan program dinamis yang lebih menjamin keberhasilan dalam menemukan solusi optimum pada kasus penentuan lintasan terpendek untuk semua pasangan simpul (*single pair shortest path*) [5]. Algoritma *Floyd Warshall* dapat digunakan untuk mencari lintasan terpendek antara semua pasangan simpul dalam graf sederhana yang terhubung [6].

Cara kerja dari Algoritma *Floyd Warshall* adalah dengan membandingkan semua lintasan yang mungkin terjadi dalam graf untuk setiap pasang simpul dan melakukan pengujian dari setiap kombinasi simpul yang diperoleh [7]. Kelebihan dari Algoritma *Floyd Warshall* antara lain [8]:

1. Algoritma *Floyd Warshall* dapat digunakan untuk mencari jarak terpendek (*shortest path*) dari setiap pasangan *edge*.
2. Algoritma *Floyd Warshall* menggunakan matriks  $n \times n$  sebagai masukan, dimana  $n$  merupakan jumlah *edge*.
3. Algoritma *Floyd Warshall* dapat mentolerir *negative edge*.

Algoritma *Floyd Warshall* menerapkan penggunaan program dinamis yang lebih menjamin

keberhasilan dalam menemukan solusi optimum pada kasus penentuan lintasan terpendek untuk semua pasangan simpul [5]. Algoritma *Floyd Warshall* menggunakan matriks sebagai representasi dari sebuah graf. Jika suatu graf terdiri dari  $n$  buah sisi (*edge*) maka matriks yang akan dibentuk dalam proses perhitungan adalah  $n \times n$ . Matriks ini merepresentasikan bobot dari seluruh sisi yang ada pada graf dengan  $d_{ij}$  adalah entri dari matriks tersebut. Bobot dari simpul  $v_i$  yang merupakan simpul awal ke simpul  $v_j$  yang merupakan simpul tujuan yang mempunyai tiga kemungkinan nilai, yaitu [9]:

1.  $d_{ij} = 0$  jika  $i = j$  (dari simpul  $v_i$  ke simpul  $v_i$  itu sendiri)
2.  $d_{ij} =$  bobot sisi jika  $i \neq j$  dan simpul  $v_i$  terhubung dengan simpul  $v_j$
3.  $d_{ij} = \infty$  jika  $i \neq j$  dan simpul  $v_i$  tidak terhubung dengan simpul  $v_j$

Algoritma *Floyd Warshall* memeriksa setiap lintasan dan membandingkan lintasan tengah untuk mendapatkan lintasan terpendek. Lintasan tengah merupakan lintasan yang dilalui antara dua simpul yang merupakan simpul antara. Perhitungan yang digunakan Algoritma *Floyd Warshall* untuk mencari lintasan terpendek adalah dengan menggunakan metode rekursif yang memiliki kelebihan yaitu mempunyai jenis *all pairs* dimana penentuan lintasan terpendek dengan menentukan semua pasangan simpul dapat ditentukan dengan cepat jika diterapkan dalam suatu sistem, kinerjanya stabil, dan keputusan yang akan diambil saling terkait [10]. Algoritma *Floyd Warshall* dapat menghitung bobot terkecil dari sisi yang memiliki bobot negatif. Bobot negatif dalam perhitungan Algoritma *Floyd Warshall* dapat ditemukan pada proses perhitungan biaya [9].

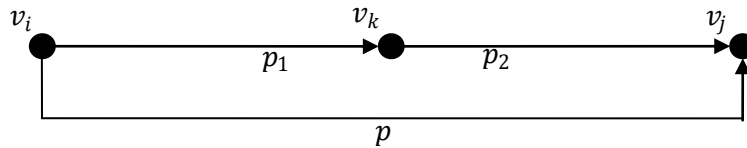
Pengerjaan Algoritma *Floyd Warshall* dimulai dengan menentukan matriks bobot yang berukuran  $n \times n$  yaitu  $L^{(0)} = (d_{ij}^{(0)})$  atau  $L^{(0)} = W$ . Selanjutnya menentukan matriks pendahulu dengan  $\Pi^{(0)} = (\Pi_{ij}^{(0)})$  dimana  $\Pi_{ij}^{(0)} = j, \forall v_j \in V$  dari matriks  $L^{(0)} = W$ . Selanjutnya menentukan  $L^{(k)} = (d_{ij}^{(k)})$  untuk setiap  $k = 1, 2, \dots, n$  dengan persamaan sebagai berikut:

$$d_{ij}^{(k)} = \begin{cases} d_{ij}, & \text{jika } k = 0 \\ \min \{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\}, & \text{jika } k > 0, \forall i, j = 1, 2, \dots, n \end{cases} \quad (1)$$

Jika  $d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$  maka  $\Pi_{ij}^{(k)} = \Pi_{ik}^{(k-1)}$ . Jika tidak demikian, maka  $\Pi_{ij}^{(k)} = (\Pi_{ij}^{(k)})$ . Lintasan yang dilalui oleh lintasan dari simpul  $v_i$  ke simpul  $v_j, \forall v_i, v_j \in V$  diberikan oleh matriks  $\Pi^{(k)} = (\Pi_{ij}^{(k)})$ . Bobot lintasan terpendek dari simpul  $v_i$  ke simpul  $v_j, \forall v_i, v_j \in V$  adalah nilai dari setiap elemen pada matriks  $L^{(k)} = (d_{ij}^{(k)})$ .  $L^{(k)}$  adalah matriks bobot pada iterasi ke- $k$ , dengan  $k = (0, 1, 2, \dots, n)$ .  $d_{ij}^{(k)}$  adalah jarak antar simpul  $i$  ke simpul  $j$  pada iterasi ke  $k$ . Dimana  $\Pi^{(k)}$  adalah matriks pendahulu pada iterasi ke- $k$ , dengan  $k = (0, 1, 2, \dots, n)$ ,  $\Pi_{ij}^{(k)}$  adalah lintasan simpul  $i$  ke simpul  $j$  pada iterasi ke- $k$ , dan  $W$  adalah bobot jarak.

#### **PENENTUAN LINTASAN TERPENDEK SETIAP PASANGAN SIMPUL PADA ALGORITMA FLOYD WARSHALL**

Misalkan terdapat suatu graf  $G$  dengan  $V = \{v_1, v_2, \dots, v_n\}$ . Misalkan pula simpul  $V^* = \{v_1, v_2, \dots, v_k\}$  untuk  $V^*$  sebarang subset dari  $V$ . Andaikan setiap lintasan dari simpul  $v_i$  ke simpul  $v_j$  mempunyai simpul antara yang berada pada  $V^*$  dan  $p$  adalah lintasan yang memiliki bobot minimum diantara lintasan-lintasan tersebut.



Gambar 1 Lintasan terpendek dari simpul  $v_i$  ke simpul  $v_j$  dengan simpul  $v_k$  sebagai simpul antara

Lintasan  $p$  dapat dijabarkan  $v_i \xrightarrow{p_1} v_k \xrightarrow{p_2} v_j$  menjadi seperti pada Gambar 1 dengan  $v_k$  adalah simpul antara pada lintasan  $p$ . Lintasan  $p_1$  merupakan lintasan terpendek dari simpul  $v_i$  ke simpul  $v_k$  dengan semua simpul antara yang berada dalam himpunan  $\{v_1, v_2, \dots, v_{k-1}\}$ . Karena  $v_k$  bukan merupakan simpul antara pada lintasan  $p_1$ , maka  $p_1$  adalah lintasan terpendek dari simpul  $v_i$  ke simpul  $v_k$ . Demikian pula pada lintasan  $p_2$  yang merupakan lintasan terpendek dari simpul  $v_k$  ke simpul  $v_j$ .

Misalkan diberikan graf  $G = (V, E)$  dengan  $V = \{v_1, v_2, \dots, v_n\}$  dan  $E = \{e_1, e_2, \dots, e_n\}$ . Didefinisikan matriks  $\Pi^{(k)} = (\Pi_{ij}^{(k)})$  merupakan matriks berukuran  $n \times n$  yang setiap elemennya adalah simpul pada lintasan dari simpul  $v_i$  ke simpul  $v_j$  dengan semua simpul yang berada dalam himpunan  $\{v_1, v_2, \dots, v_k\}$ . Secara umum penentuan lintasan terpendek setiap pasangan simpul pada Algoritma Floyd Warshall ditulis sebagai berikut:

1. Jika  $k = 0$  maka  $\Pi_{ij}^{(0)} = j, \forall v_j \in V$
2. Jika  $k \geq 1 (k = 1, 2, \dots, n)$  maka

$$\Pi_{ij}^{(k)} = \begin{cases} \Pi_{ij}^{(k-1)}, & \text{jika } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\ \Pi_{ik}^{(k-1)}, & \text{jika } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \end{cases} \quad (1)$$

Simpul antara pada lintasan dari simpul  $v_i$  ke simpul  $v_j, \forall v_i, v_j \in V$  diberikan oleh matriks  $\Pi^{(k)} = (\Pi_{ij}^{(k)})$ . Misalkan  $v_i$  merupakan simpul awal dan  $v_j$  sebagai simpul akhir, jika pada matriks  $\Pi^{(k)} = (\Pi_{ij}^{(k)})$ ,  $\Pi_{ij}^{(k)} = v_j$  maka tidak terdapat simpul antara  $v_i$  ke  $v_j$ . Akan tetapi jika  $\Pi_{ij}^{(k)} \neq v_j$  maka terdapat  $v_x$  yang merupakan simpul antara pada lintasan  $v_i$  ke  $v_j$  sehingga lintasan terpendek dari simpul  $v_i$  ke  $v_j$  adalah  $v_i - v_x - v_j$ .

## PENERAPAN ALGORITMA FLOYD WARSHALL UNTUK MENENTUKAN LINTASAN TERPENDEK PENGANGKUTAN SAMPAH

### 1. Data

Data yang digunakan pada penelitian ini berupa data sekunder tentang lintasan pengangkutan sampah dari 10 TPS ke TPA yang ada di Kabupaten Kubu Raya. Simpul merepresentasikan TPS yang ada di Kabupaten Kubu Raya, sedangkan sisi merepresentasikan lintasan yang menghubungkan antar TPS, dan TPS ke TPA. Sedangkan bobot merepresentasikan jarak (km) antar 10 TPS dan jarak antara TPS dan TPA.

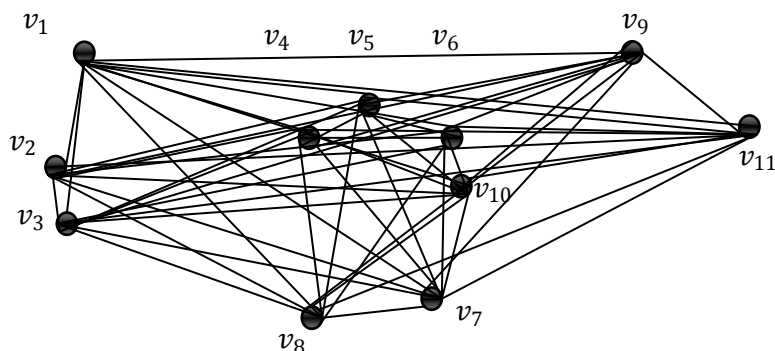
### 2. Penentuan Lintasan Terpendek menggunakan Algoritma Floyd Warshall

Langkah pertama penentuan jalur terpendek menggunakan Algoritma Floyd Warshall adalah mengumpulkan data jumlah TPS, jarak antar 10 TPS yang ada di Kabupaten Kubu Raya. TPA tersebut terletak di Kecamatan Rasau Jaya Kabupaten Kubu Raya. Data tersebut disajikan pada Tabel 1.

Tabel 1 Data Lokasi TPS dan Jarak antara TPS dan TPA

Simpul	Lokasi TPS
$v_1$	Di depan Terminal Antar Lintas Batas Negara
$v_2$	Di bawah Tol Kapuas dua
$v_3$	Di bawah Tol Kapuas dua
$v_4$	Di depan New Kalbar Processor
$v_5$	Di depan New Kalbar Processor
$v_6$	Di depan New Kalbar Processor
$v_7$	Di depan Pondok Indah Lestari
$v_8$	Di depan Pondok Indah Lestari
$v_9$	Di depan Panti Jompo
$v_{10}$	Di desa Kiwi
$v_{11}$	TPA Rasau Jaya

Lintasan terpendek yang dilalui untuk semua TPS dan TPA ditunjukkan pada Gambar 2.



Gambar 2 Graf Lokasi TPS dan TPA

Selanjutnya menentukan jarak antar simpul yang diperoleh dari *Google Maps* dan survei lapangan, seperti pada Tabel 1

Tabel 1 Jarak antar Simpul (km)

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	$v_7$	$v_8$	$v_9$	$v_{10}$	$v_{11}$
$v_1$	0	5	5.1	10	10.1	10.3	7.6	7.7	10	22	23.2
$v_2$	5	0	0.1	4.9	5.1	5.2	2.8	2.9	5.1	17	18.2
$v_3$	5.1	0.1	0	4.8	5	5.1	2.7	3	5	16.9	18.1
$v_4$	10	4.9	4.8	0	0.2	0.3	7.8	7.9	2	11.4	23.1
$v_5$	10.1	5.1	5	0.2	0	0.1	7.6	7.7	1.8	11.2	22.9
$v_6$	10.3	5.2	5.1	0.3	0.1	0	7.5	7.6	1.7	11.1	22.8
$v_7$	7.6	2.8	2.7	7.8	7.6	7.5	0	0.1	6.9	15	16.2
$v_8$	7.7	2.9	3	7.9	7.7	7.6	0.1	0	6.8	14.9	16.1
$v_9$	10	5.1	5	2	1.8	1.7	6.9	6.8	0	11	12.4
$v_{10}$	22	17	16.9	11.4	11.2	11.1	15	14.9	11	0	10.6
$v_{11}$	23.2	18.2	18.1	23.1	22.9	22.8	16.2	16.1	12.4	10.6	0

Setelah itu, langkah selanjutnya adalah graf berbobot yang sudah terbentuk direpresentasikan ke dalam matriks bobot berukuran  $n \times n$  (dengan  $n$  adalah jumlah simpul).

$$L^{(0)} = \begin{bmatrix} 0 & 5 & 5.1 & 10 & 10.1 & 10.3 & 7.6 & 7.7 & 10 & 22 & 23.2 \\ 5 & 0 & 0.1 & 4.9 & 5.1 & 5.2 & 2.8 & 2.9 & 5.1 & 17 & 18.2 \\ 5.1 & 0.1 & 0 & 4.8 & 5 & 5.1 & 2.7 & 3 & 5 & 16.9 & 18.1 \\ 10 & 4.9 & 4.8 & 0 & 0.2 & 0.3 & 7.8 & 7.9 & 2 & 11.4 & 23.1 \\ 10.1 & 5.1 & 5 & 0.2 & 0 & 0.1 & 7.6 & 7.7 & 1.8 & 11.2 & 22.9 \\ 10.3 & 5.2 & 5.1 & 0.3 & 0.1 & 0 & 7.5 & 7.6 & 1.7 & 11.1 & 22.81 \\ 7.6 & 2.8 & 2.7 & 7.8 & 7.6 & 7.5 & 0 & 0.1 & 6.9 & 15 & 16.2 \\ 7.7 & 2.9 & 3 & 7.9 & 7.7 & 7.6 & 0.1 & 0 & 6.8 & 14.9 & 16.1 \\ 10 & 5.1 & 5 & 2 & 1.8 & 1.7 & 6.9 & 6.8 & 0 & 11 & 12.4 \\ 22 & 17 & 16.9 & 11.4 & 11.2 & 11.1 & 15 & 14.9 & 11 & 0 & 10.6 \\ 23.2 & 18.2 & 18.1 & 23.1 & 22.9 & 22.8 & 16.2 & 16.1 & 12.4 & 10.6 & 0 \end{bmatrix}$$

Kemudian menentukan matriks pendahulu yang merupakan matriks awal dari representasi permasalahan yang diambil.

$$\Pi^{(0)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{bmatrix}$$

Setelah itu dilakukan iterasi sebanyak 11 simpul untuk menentukan bobot lintasan terpendek. Pada iterasi 1 dengan menggunakan persamaan 1 dan persamaan 2, diperoleh matriks bobot  $L^{(1)}$  dan matriks pendahulu  $\Pi^{(1)}$  sebagai berikut:

$$L^{(1)} = \begin{bmatrix} 0 & 5 & 5.1 & 10 & 10.1 & 10.3 & 7.6 & 7.7 & 10 & 22 & 23.2 \\ 5 & 0 & 0.1 & 4.9 & 5.1 & 5.2 & 2.8 & 2.9 & 5.1 & 17 & 18.2 \\ 5.1 & 0.1 & 0 & 4.8 & 5 & 5.1 & 2.7 & 3 & 5 & 16.9 & 18.1 \\ 10 & 4.9 & 4.8 & 0 & 0.2 & 0.3 & 7.8 & 7.9 & 2 & 11.4 & 23.1 \\ 10.1 & 5.1 & 5 & 0.2 & 0 & 0.1 & 7.6 & 7.7 & 1.8 & 11.2 & 22.9 \\ 10.3 & 5.2 & 5.1 & 0.3 & 0.1 & 0 & 7.5 & 7.6 & 1.7 & 11.1 & 22.81 \\ 7.6 & 2.8 & 2.7 & 7.8 & 7.6 & 7.5 & 0 & 0.1 & 6.9 & 15 & 16.2 \\ 7.7 & 2.9 & 3 & 7.9 & 7.7 & 7.6 & 0.1 & 0 & 6.8 & 14.9 & 16.1 \\ 10 & 5.1 & 5 & 2 & 1.8 & 1.7 & 6.9 & 6.8 & 0 & 11 & 12.4 \\ 22 & 17 & 16.9 & 11.4 & 11.2 & 11.1 & 15 & 14.9 & 11 & 0 & 10.6 \\ 23.2 & 18.2 & 18.1 & 23.1 & 22.9 & 22.8 & 16.2 & 16.1 & 12.4 & 10.6 & 0 \end{bmatrix} \quad \Pi^{(1)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{bmatrix}$$

Bobot dari semua pasangan lintasan terpendek direpresentasikan oleh  $L^{(1)}$  dan lintasannya oleh  $\Pi^{(1)}$ . Perhitungan yang sama dilakukan pada iterasi 2, dengan menggunakan persamaan 1 dan persamaan 2, diperoleh matriks bobot  $L^{(2)}$  dan matriks pendahulu  $\Pi^{(2)}$  sebagai berikut:







$$L^{(9)} = \begin{bmatrix} 0 & 5 & 5.1 & 9.9 & 10.1 & 10.2 & 7.6 & 7.7 & 10 & 21 & 22.4 \\ 5 & 0 & 0.1 & 4.9 & 5.1 & 5.2 & 2.8 & 2.9 & 5.1 & 16.1 & 17.5 \\ 5.1 & 0.1 & 0 & 4.8 & 5 & 5.1 & 2.7 & 2.8 & 5 & 16 & 17.4 \\ 9.9 & 4.9 & 4.8 & 0 & 0.2 & 0.3 & 7.5 & 7.6 & 2 & 11.4 & 14.4 \\ 10.1 & 5.1 & 5 & 0.2 & 0 & 0.1 & 7.6 & 7.7 & 1.8 & 11.2 & 14.2 \\ 10.2 & 5.2 & 5.1 & 0.3 & 0.1 & 0 & 7.5 & 7.6 & 1.7 & 11.1 & 14.1 \\ 7.6 & 2.8 & 2.7 & 7.5 & 7.6 & 7.5 & 0 & 0.1 & 6.9 & 15 & 16.2 \\ 7.7 & 2.9 & 2.8 & 7.6 & 7.7 & 7.6 & 0.1 & 0 & 6.8 & 14.9 & 16.1 \\ 10 & 5.1 & 5 & 2 & 1.8 & 1.7 & 6.9 & 6.8 & 0 & 11 & 12.4 \\ 21 & 16.1 & 16 & 11.4 & 11.2 & 11.1 & 15 & 14.9 & 11 & 0 & 10.6 \\ 22.4 & 17.5 & 17.4 & 14.4 & 14.2 & 14.1 & 16.2 & 16.1 & 12.4 & 10.6 & 0 \end{bmatrix} \quad \Pi^{(9)} = \begin{bmatrix} 1 & 2 & 3 & 2 & 5 & 2 & 7 & 8 & 9 & 9 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 9 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 7 & 9 & 9 & 9 \\ 2 & 2 & 3 & 4 & 5 & 6 & 3 & 3 & 9 & 10 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 \\ 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 \\ 1 & 2 & 3 & 3 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 7 & 7 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 9 & 9 & 9 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 9 & 9 & 9 & 9 & 9 & 9 & 7 & 8 & 9 & 10 & 11 \end{bmatrix}$$

Bobot dari semua pasangan lintasan terpendek diberikan oleh  $L^{(9)}$  dan lintasannya direpresentasikan oleh  $\Pi^{(9)}$ . Perhitungan yang sama dilakukan pada iterasi 10, dengan menggunakan persamaan 1 dan persamaan 2, diperoleh matriks bobot  $L^{(10)}$  dan matriks pendahulu  $\Pi^{(10)}$  sebagai berikut:

$$L^{(10)} = \begin{bmatrix} 0 & 5 & 5.1 & 9.9 & 10.1 & 10.2 & 7.6 & 7.7 & 10 & 21 & 22.4 \\ 5 & 0 & 0.1 & 4.9 & 5.1 & 5.2 & 2.8 & 2.9 & 5.1 & 16.1 & 17.5 \\ 5.1 & 0.1 & 0 & 4.8 & 5 & 5.1 & 2.7 & 2.8 & 5 & 16 & 17.4 \\ 9.9 & 4.9 & 4.8 & 0 & 0.2 & 0.3 & 7.5 & 7.6 & 2 & 11.4 & 14.4 \\ 10.1 & 5.1 & 5 & 0.2 & 0 & 0.1 & 7.6 & 7.7 & 1.8 & 11.2 & 14.2 \\ 10.2 & 5.2 & 5.1 & 0.3 & 0.1 & 0 & 7.5 & 7.6 & 1.7 & 11.1 & 14.1 \\ 7.6 & 2.8 & 2.7 & 7.5 & 7.6 & 7.5 & 0 & 0.1 & 6.9 & 15 & 16.2 \\ 7.7 & 2.9 & 2.8 & 7.6 & 7.7 & 7.6 & 0.1 & 0 & 6.8 & 14.9 & 16.1 \\ 10 & 5.1 & 5 & 2 & 1.8 & 1.7 & 6.9 & 6.8 & 0 & 11 & 12.4 \\ 21 & 16.1 & 16 & 11.4 & 11.2 & 11.1 & 15 & 14.9 & 11 & 0 & 10.6 \\ 22.4 & 17.5 & 17.4 & 14.4 & 14.2 & 14.1 & 16.2 & 16.1 & 12.4 & 10.6 & 0 \end{bmatrix} \quad \Pi^{(10)} = \begin{bmatrix} 1 & 2 & 3 & 2 & 5 & 2 & 7 & 8 & 9 & 9 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 9 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 7 & 9 & 9 & 9 \\ 2 & 2 & 3 & 4 & 5 & 6 & 7 & 3 & 9 & 10 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 \\ 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 \\ 1 & 2 & 3 & 3 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 7 & 7 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 9 & 9 & 9 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 9 & 9 & 9 & 9 & 9 & 9 & 7 & 8 & 9 & 10 & 11 \end{bmatrix}$$

Bobot dari semua pasangan lintasan terpendek diberikan oleh  $L^{(10)}$  dan lintasannya direpresentasikan oleh  $\Pi^{(10)}$ . Perhitungan yang sama dilakukan pada iterasi 11, dengan menggunakan persamaan 1 dan persamaan 2, diperoleh matriks bobot  $L^{(11)}$  dan matriks pendahulu  $\Pi^{(11)}$  sebagai berikut:

$$L^{(11)} = \begin{bmatrix} 0 & 5 & 5.1 & 9.9 & 10.1 & 10.2 & 7.6 & 7.7 & 10 & 21 & 22.4 \\ 5 & 0 & 0.1 & 4.9 & 5.1 & 5.2 & 2.8 & 2.9 & 5.1 & 16.1 & 17.5 \\ 5.1 & 0.1 & 0 & 4.8 & 5 & 5.1 & 2.7 & 2.8 & 5 & 16 & 17.4 \\ 9.9 & 4.9 & 4.8 & 0 & 0.2 & 0.3 & 7.5 & 7.6 & 2 & 11.4 & 14.4 \\ 10.1 & 5.1 & 5 & 0.2 & 0 & 0.1 & 7.6 & 7.7 & 1.8 & 11.2 & 14.2 \\ 10.2 & 5.2 & 5.1 & 0.3 & 0.1 & 0 & 7.5 & 7.6 & 1.7 & 11.1 & 14.1 \\ 7.6 & 2.8 & 2.7 & 7.5 & 7.6 & 7.5 & 0 & 0.1 & 6.9 & 15 & 16.2 \\ 7.7 & 2.9 & 2.8 & 7.6 & 7.7 & 7.6 & 0.1 & 0 & 6.8 & 14.9 & 16.1 \\ 10 & 5.1 & 5 & 2 & 1.8 & 1.7 & 6.9 & 6.8 & 0 & 11 & 12.4 \\ 21 & 16.1 & 16 & 11.4 & 11.2 & 11.1 & 15 & 14.9 & 11 & 0 & 10.6 \\ 22.4 & 17.5 & 17.4 & 14.4 & 14.2 & 14.1 & 16.2 & 16.1 & 12.4 & 10.6 & 0 \end{bmatrix} \quad \Pi^{(11)} = \begin{bmatrix} 1 & 2 & 3 & 2 & 5 & 2 & 7 & 8 & 9 & 9 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 9 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 7 & 9 & 9 & 9 \\ 2 & 2 & 3 & 4 & 5 & 6 & 3 & 3 & 9 & 10 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 \\ 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 \\ 1 & 2 & 3 & 3 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 7 & 7 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 9 & 9 & 9 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 9 & 9 & 9 & 9 & 9 & 9 & 7 & 8 & 9 & 10 & 11 \end{bmatrix}$$

Berdasarkan matriks  $L^{(11)}$  dan  $\Pi^{(11)}$  diperoleh lintasan terpendek untuk pengangkutan sampah dengan simpul awal  $v_1$  ke simpul akhir  $v_{11}$  adalah  $v_1 - v_2 - v_3 - v_7 - v_8 - v_9 - v_6 - v_5 - v_4 - v_{10} - v_{11}$ . Artinya jika pengangkut sampah ingin mengangkut sampah dari simpul  $v_1$  (TPS di depan Terminal Antar Lintas Batas Negara) ke TPA maka pengangkut sampah dapat melewati TPS yang berada di bawah tol kapuas dua (sebanyak dua TPS), lalu dilanjutkan untuk mengangkut sampah di TPS yang berada depan Pondok Indah Lestari (sebanyak dua TPS) dan dilanjutkan ke TPS yang berada di depan Panti Jompo. Selanjutnya pihak pengangkut dapat melanjutkan pengangkutan sampah di TPS yang ada di depan New Kalbar Processor (sebanyak tiga TPS) dan dilanjutkan ke TPS yang ada di Desa Kiwi, setelah itu sampah dapat diangkat ke TPA yang berada di Desa Rasau Jaya.

## PENUTUP

Penentuan lintasan terpendek pada kasus pengangkutan sampah dari 10 TPS ke TPA di Kabupaten Kubu Raya dapat ditentukan dengan menggunakan Algoritma *Floyd Warshall*. Lintasan terpendek dengan simpul awal,  $v_1$  ke simpul akhir  $v_{11}$  adalah  $v_1 - v_2 - v_3 - v_7 - v_8 - v_9 - v_6 - v_5 - v_4 - v_{10} - v_{11}$ . Artinya jika pengangkut sampah ingin mengangkut sampah dari simpul  $v_1$  (TPS di depan Terminal Antar Lintas Batas Negara) ke TPA maka pengangkut sampah dapat melewati TPS yang berada di bawah tol kapuas dua (sebanyak dua TPS), lalu dilanjutkan untuk mengangkut sampah di TPS yang berada depan Pondok Indah Lestari (sebanyak dua TPS) dan dilanjutkan ke TPS yang berada di depan Panti Jompo. Selanjutnya pihak pengangkut dapat melanjutkan pengangkutan sampah di TPS yang ada di depan New Kalbar Processor (sebanyak tiga TPS) dan dilanjutkan ke TPS yang ada di Desa Kiwi, setelah itu sampah dapat diangkut ke TPA yang berada di Desa Rasau Jaya.

## DAFTAR PUSTAKA

- [1]. Chandra B. *Pengantar Kesehatan Lingkungan*. Jakarta. Penerbit buku kedokteran EGC, 2007.
- [2]. Badan Pusat Statistik Provinsi Kalimantan Barat [Internet]. 2016 [updated 2017; cited 2017 Maret 20]. Available from: <https://kalbar.bps.go.id/>
- [3]. Dinas Cipta Karya Tata Ruang dan Kebersihan, 2017
- [4]. Munir R. *Matematika Diskrit*. Ed ke 3. Bandung: Informatika; 2010
- [5]. Ardiansyah I. Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Menggunakan Algoritma Floyd di Lokasi Wisata Purbalingga. *JUITA ISSN*, 2012; 2(2), 2086-9398
- [6]. Rosen K H. *Discrete Mathematics and Its Applications*. 7<sup>th</sup> Edition. Toronto: Mc Graw Hill; 2011.
- [7]. Siang J J. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Jogjakarta: Andi; 2009.
- [8]. Jonemaro E M A. Rancang Bangun Aplikasi Web Pencarian Rute Terpendek Antar Gedung di Kampus Menggunakan Algoritma Floyd-warshall. *Journal Basic Science And Technology*, 2012; 1(2), 30-34.
- [9]. Purwananto Y. Implementasi dan Analisis Algoritma Pencarian Rute Terpendek di Kota Surabaya. *Jurnal Penelitian dan Pengembangan Telekomunikasi*, 2005; 10 (2):94-101
- [10]. Iftadi I. Perancangan Peta Evakuasi Menggunakan Algoritma Floyd Warshall untuk Penentuan Lintasan Terpendek. *Performa*. 2011; 10(2): 95 – 104.

VEGA SETIAWAN	: vegasetiawan00@gmail.com
MARIATUL KIFTIAH	: kiftiah_mariatul@ymail.com
WORO BUDIARTINI PARTIWI	: woro.partiwi@ymail.com