Volume 2 No 1 (2023) 25-38
http://jurnal.fmipa.unmul.ac.id/index.php/Basis

# Penerapan Algoritma Dijkstra dan Algoritma Greedy Pada Optimasi Jalur Evakuasi Banjir

Muliya Wiladi[1], Wasono[2*], Asmaidi[3]

[1,2,3] *Universitas Mulawarman*

Dikirim: Oktober 2022; Diterima: November 2022; Dipublikasi: Maret 2023

Alamat Email Korespondesi: wasono@fmipa.unmul.ac.id

**Abstrak**
Penyelamatan dan evakuasi merupakan hal yang harus dilakukan dengan segera setelah terjadinya sebuah bencana agar dapat mengurangi jumlah korban dan mengurangi dampak buruk yang ditimbulkan. Salah satu bencana yang sering terjadi di Kota Samarinda yaitu bencana banjir. Pada penelitian ini mengkaji bagaimana pengoptimasian jalur evakuasi warga yang terdampak banjir di Kelurahan Sempaja Timur, Kota Samarinda menggunakan algoritma Dijkstra dan algoritma Greedy. Algoritma Dijkstra merupakan algoritma yang dapat digunakan dalam menentukan jalur terpendek. Algoritma Greedy adalah algoritma yang membuat pilihan terbaik (nilai tertinggi) pada setiap langkahnya. Berdasarkan hasil penelitian, didapatkan 11 lintasan optimal yang dapat digunakan pada proses evakuasi di Jl. Terong dan Jl. Terong Pipit, Kelurahan Sempaja Timur, Kota Samarinda. Sebelas lintasan optimal yang didapatkan merupakan urutan evakuasi banjir yang dapat digunakan agar seluruh warga dapat dievakuasi dan proses evakuasi berjalan optimal.

**Kata Kunci:**
*Algoritma Dijkstra, Algoritma Greedy, Masalah Knapsack, Masalah Lintasan Terpendek, Optimasi*

## PENDAHULUAN

Optimasi merupakan suatu proses yang memiliki tujuan mencari nilai minimum atau maksimum suatu fungsi dengan memperhatikan kendala-kendala yang ada. Masalah optimasi adalah masalah besaran tertentu atau disebut juga fungsi objektif *(objective)* yang dimaksimalkan atau diminimumkan dengan bergantung pada sejumlah berhingga variabel masukan (input variables). Variabel-variabel ini dapat saling bergantung melalui satu atau lebih kendala *(constrains)* atau juga tidak saling bergantung [7]. Optimasi dalam hal waktu dan beberapa penghematan di bidang lain dapat dilakukan dengan menentukan jalur terpendak. Pekerjaan dapat lebih efektif, cepat dan juga menghemat biaya dengan melewati jalur terpendek. Jalur terpendek merupakan jumlah nilai dari keseluruhan bentuk lintasan yang bernilai minimal [1].

Salah satu masalah optimasi kombinatorial yang juga banyak dipelajari adalah masalah knapsack 0-1. Masalah ini mempunyai tujuan untuk memaksimumkan nilai total dari seluruh objek yang dapat dimuat ke dalam knapsack (suatu wadah atau tas) dengan kendalanya yaitu memastikan total bobot objek kurang dari atau sama dengan kapasitas maksimal *knapsack*. Setiap objek tidak dapat dimasukkan secara tidak utuh atau dimasukkan lebih dari satu kali ke dalam *knapsack* [3]. Terdapat banyak cara yang dapat dilakukan untuk menyelesaikan masalah *knapsack*, salah satunya adalah Algoritma Greedy. Pemecahan masalah optimasi dengan Algoritma Greedy tidak selalu mendapatkan solusi yang optimum. Prinsip utama dari algoritma ini adalah memilih sebanyak mungkin objek yang dapat diperoleh sekarang [6].

Algoritma Dijkstra merupakan algoritma yang dapat digunakan dalam menentukan jalur terpendek dari titik sumber menuju titik tujuan berdasarkan nilai pada setiap sisi. Nilai tersebut dapat berupa waktu, jarak, biaya maupun yang lainnya. Algoritma Dijkstra bekerja dengan cara melewati setiap titik yang terdapat pada graf dengan dimulai dari titik sumber. Algoritma ini memilih titik-titik terdekat dan menghitung total nilai semua sisi yang dilewati dengan dilakukan secara berulang [5].

Masalah *knapsack* dan masalah lintasan terpendek juga dapat dikerjakan secara bersamaan, seperti penelitian yang dilakukan oleh [10]. Masalah tersebut menjadi masalah kompleks yang menggabungkan dua permasalahan yaitu *knapsack* dan lintasan terpendek pada graf berbobot. Penelitian tersebut menggunakan Algoritma Dijkstra untuk mencari lintasan terpendek dan algoritma Aproksimasi Greedy Dantzig untuk memilih objek pada lintasan terpilih. Penelitian yang dilakukan oleh [10] mendapati hasil yang belum optimal pada kasus dengan pembatasan bobot objek.

Banjir merupakan masalah lingkungan yang biasa terjadi di sejumlah kawasan di Samarinda, Kalimantan Timur, salah satunya adalah Kelurahan Sempaja Timur. Pada tanggal 9 Juni 2019, Kelurahan Sempaja Timur merupakan daerah yang paling parah dilanda banjir. Jumlah penduduk yang terkena bencana di Kelurahan Sempaja Timur, Kecamatan Samarinda Utara adalah sebanyak 2.327 jiwa. Penyebab banjir di Kota Samarinda dikarenakan tingginya intensitas hujan. Intensitas hujan yang tinggi menyebabkan meluapnya Sungai Mahakam dan Sungai Karang Mumus sehingga ketinggian banjir semakin naik. Akibat ketinggian banjir yang naik, tim SAR gabungan harus mengevakuasi sejumlah warga untuk mengungsi [2]. Menurut Peraturan Walikota Samarinda Nomor 6 Tahun 2014 [11], Tanggap darurat bencana adalah serangkaian kegiatan yang dilakukan dengan segera pada saat kejadian bencana untuk menangani dampak buruk yang ditimbulkan, yang meliputi kegiatan penyelamatan dan evakuasi korban, harta benda, pemenuhan kebutuhan dasar, perlindungan, pengurusan pengungsi, penyelamatan, serta pemulihan prasarana dan sarana.

Berdasarkan uraian di atas, penelitian ini akan mengkaji tentang optimasi jalur terpendek yang diselesaikan menggunakan penggabungan Algoritma Dijkstra dan Algoritma Greedy dengan studi kasus di Kelurahan Sempaja Timur, Kota Samarinda.

## TINJAUAN PUSTAKA

### 1. Optimasi

Optimasi merupakan suatu proses maksimasi atau minimasi dengan mencari kondisi yang optimum. Keadaan optimum yang memberikan nilai maksimum (maksimasi) biasanya berkaiatan dengan masalah keuntungan. Keadaan optimum yang memberikan nilai minimum (minimasi) biasanya berkaitan dengan masalah pengeluaran. Hal-hal penting dalam pembelajaran optimasi yaitu meliputi fungsi objektif, variabel keputusan (*decision variables*), kendala (*constraints*), dan pernyataan matematika yang menyatakan daerah nilai minimum atau maksimum akan dicari [9].

### 2. Graf

Graf $G$ dinotasikan sebagai $G = (V, E)$ dengan $V$ merupakan himpunan tidak kosong dari titik-titik (*vertices* atau *node*) dan $E$ merupakan himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang titik. $V$ dinyatakan dengan himpunan tak kosong, sedangkan $E$ boleh kosong, oleh karena itu sebuah graf dimungkinkan untuk tidak mempunyai sisi tetapi harus memiliki minimal satu titik. Graf yang tidak terdapat sisi dan memiliki satu buah titik dinamakan graf trivial [4].

## 3. Masalah Lintasan Terpendek

Salah satu masalah optimasi di dalam graf adalah masalah pencarian lintasan terpendek. Masalah lintasan terpendek ini menggunakan graf berbobot yang setiap sisinya memiliki satuan nilai atau bobot. Nilai atau bobot tersebut dapat menyatakan jarak antar kota, waktu pengiriman, ongkos pembangunan, dan sebagainya. Asumsi yang digunakan pada masalah lintasan terpendek adalah semua bobot bernilai positif [4].

## 4. Masalah *Knapsack*

Masalah *knapsack* 0-1 adalah salah satu masalah optimasi kombinatorial yang banyak dipelajari. Masalah ini mempunyai tujuan untuk memaksimumkan nilai total dari seluruh objek yang dapat dimuat ke dalam *knapsack* dengan kendalanya yaitu memastikan total bobot objek kurang dari atau sama dengan kapasitas maksimal *knapsack.* Setiap objek tidak dapat dimasukkan secara tidak utuh atau dimasukkan lebih dari satu kali ke dalam *knapsack*. Masalah *knapsack* 0-1 dapat dirumuskan sebagai berikut [3]:

Fungsi tujuan:

$$\sum_{i=1}^{m} q_i y_{i,} \qquad (1)$$

Keterangan:

$m$ : Jumlah titik
$i$ : Indeks objek
$q_i$ : Nilai objek ke-$i$
$y_i$ : Menunjukan objek dipilih atau tidak dengan $y_i$ bernilai 0 atau 1

Fungsi kendala:

$$\sum_{i=1}^{m} w_i y_i \leq W, \qquad (2)$$

Keterangan:

$m$ : Jumlah titik
$i$ : Indeks objek
$w_i$ : Bobot objek ke-$i$
$y_i$ : Menunjukan objek dipilih atau tidak dengan $y_i$ bernilai 0 atau 1
$W$ : Kapasitas *knapsack*

## 5. Algoritma

Algoritma merupakan langkah-langkah berurutan untuk melakukan komputasi atau menyelesaikan sebuah masalah. Algoritma merupakan istilah yang diambil dari nama seorang ahli matematika yang hidup sekitar abad ke-9. Awalnya, kata *algorism* merupakan aturan saat melakukan aritmatika menggunakan notasi desimal, namun pada abad ke-18 istilah *algorism* berubah menjadi kata *algorithm* atau dalam bahasa Indonesia disebut algoritma. Seiring meningkatnya minat dalam mesin komputasi, konsep algoritma diberikan arti yang lebih umum, tidak hanya prosedur dalam melakukan aritmatika tetapi juga memuat semua prosedur pasti untuk menyelesaikan masalah [8].

### 5.1 Algoritma Dijkstra

Algoritma lintasan terpendek yang terkenal salah satunya adalah Algoritma Dijkstra (sesuai dengan nama penemunya, Edsger W. Dijkstra). Algoritma Dijkstra diterapkan untuk mencari lintasan terpendek pada graf berarah. Namun, algoritma ini juga benar untuk graf tak-berarah [4].

Secara singkat Algoritma Dijkstra dijelaskan sebagai berikut [5]:

1. Inisialisasi titik ($v$).
2. Inisialisasi jarak antar titik ($e$).
3. Menentukan titik awal dan titik tujuan.
4. Memberi label permanen 0 (nol) pada titik awal dan label sementara ∞ (tak hingga) pada titik lainnya.
5. Memberi label sementara, yaitu $min\,\{label\ lama\ v_i, (label\ lama\ v_i + e_i)\}$ untuk setiap titik $v_t$ yang belum mendapat label permanen.
6. Mencari nilai minimum dari semua titik yang masih berlabel sementara.
7. Merubah titik minimum yang berlabel sementara menjadi titik dengan label permanen, jika lebih dari satu titik maka dipilih sembarang titik.
8. Mengulangi langkah 5 sampai 7 hingga semua titik mendapat label permanen.
9. Menyimpan hasil perhitungan.
10. Menampilkan hasil pencarian.

### 5.2 Algoritma Greedy

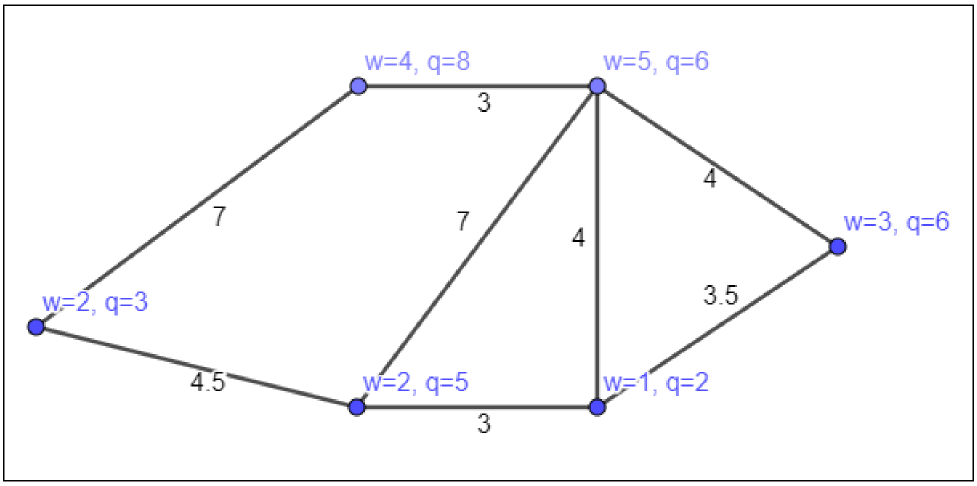
Solusi optimasi biasanya merupakan pendekatan yang paling sederhana. Salah satu pendekatan ini adalah pendakatan yang memilih pilihan terbaik pada setiap langkah, daripada mempertimbangkan semua urutan langkah yang mungkin mengarah pada solusi optimal. Algoritma yang membuat pilihan terbaik (nilai tertinggi) pada setiap langkahnya disebut Algoritma Greedy. Solusi yang didapat dari Algoritma Greedy perlu ditentukan apakah solusi tersebut merupakan solusi optimal [8].

Terdapat tiga jenis Algoritma Greedy yang dapat digunakan pada penyelesaian masalah *knapsack*, yaitu [6]:

1. Greedy *By Weight*
   Pada setiap langkah, objek yang dipilih adalah objek yang mempunyai berat teringan. Langkah diambil dengan tujuan memaksimumkan keuntungan dengan memasukkan sebanyak mungkin objek ke dalam knapsack. Pertama yang dilakukan adalah mengurutkan secara menaik objek-objek berdasarkan beratnya. Kemudian dipilih satu demi satu objek yang dapat ditampung oleh knapsack hingga knapsack penuh atau tidak dapat lagi memasukkan objek lainnya ke dalam knapsack.
2. Greedy *By Profit*
   Pada setiap langkah, objek yang dipilih adalah objek dengan keuntungan terbesar. Langkah diambil dengan tujuan memaksimumkan keuntungan dengan memasukkan objek yang paling menguntungkan. Pertama yang dilakukan adalah mengurutkan secara menaik objek-objek berdasarkan profit-nya. Kemudian dipilih satu demi satu objek yang dapat ditampung oleh knapsack hingga knapsack penuh atau tidak dapat lagi memasukkan objek lainnya ke dalam knapsack.
3. Greedy *By Density*
   Pada setiap langkah, objek yang dipilih adalah objek yang mempunyai nilai perbandingan terbesar antara profit dan berat. Langkah diambil dengan tujuan memaksimumkan keuntungan dengan memasukkan objek yang mempunyai density terbesar. Pertama yang dilakukan adalah mengurutkan secara menaik objek-objek berdasarkan density-nya. Kemudian dipilih satu demi satu objek yang dapat ditampung oleh knapsack hingga knapsack penuh atau tidak dapat lagi memasukkan objek lainnya ke dalam knapsack.

## 6. *Knacpsack* dan Lintasan Terpendek

Graf berbobot dinotasikan sebagai $G^* = (V,\ E)$ dengan $V$ merupakan himpunan tidak kosong dari titik-titik (*vertices* atau *node*) dan $E$ merupakan himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang titik. Setiap titik pada graf tersebut diberikan bobot $(w_i)$ dan nilai $(q_i)$. $w_i$ dan $q_i$ pada titik melambangkan bobot atau berat dan nilai atau profit dari sebuah objek. Pada graf tersebut juga diberikan sebuah batas maksimum bobot *Knapsack* yang dilambangkan dengan $m$ [10].




Gambar 1. Graf berbobot $G^*$

Graf $G^*$ yang sudah dijelaskan sebelumnya dicari nilai maksimum objek-objek pada graf tersebut dari lintasan terpendek antara dua titik yang ditentukan (titik awal dan titik tujuan) dengan sebuah batas bobot *Knapsack*. Langkah pertama yaitu menentukan lintasan terpendek antara dua titik yang telah ditentukan. Langkah kedua yaitu mempertimbangkan bobot dan nilai objek pada titik pada hasil sub-graf dari lintasan terpendek terpilih [10]. Penelitian yang dilakukan oleh Voloch [10] menentukan lintasan terpendek antara dua titik dengan menggunakan Algoritma Dijkstra dan memilih objek atau titik dari lintasan terpendek terpilih menggunakan algoritma Aproksimasi Greedy Dantzig.

## HASIL DAN PEMBAHASAN

## 1. Pembuatan Graf Berbobot

Graf berbobot yang dibuat menggunakan data rute dan data penduduk terdampak banjir pada tanggal 12 Juni 2019 di Kelurahan Sempaja Timur. Berikut tabel data nama jalan dan jumlah penduduk terdampak banjir di Kelurahan Sempaja Timur yang digunakan pada penelitian ini [2]:

Tabel 1. Data Nama Jalan dan Jumlah Penduduk

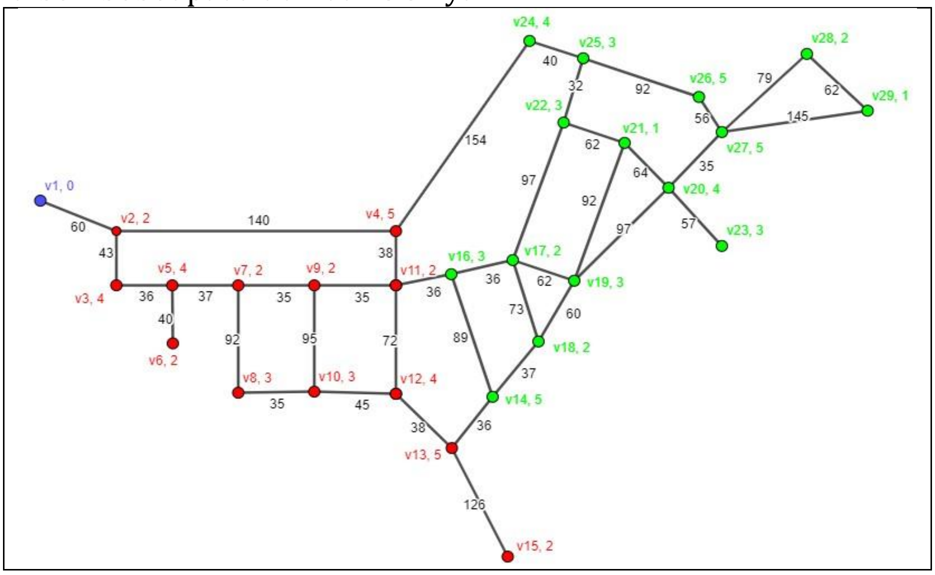
| Nama Jalan | Jumlah Penduduk |
|---|---|
| Jl. Terong 1 s/d 6 | 435 |
| Jl. Terong Pipit 1 s/d 12 | 434 |

Sumber: Data Kementerian Pekerjaan Umum dan Perumahan Rakyat tahun 2019

Data nama jalan pada Tabel 1 dibuat graf dengan mengikuti rute dari *Google Maps.* Titik pada graf yang dibuat menunjukkan persimpangan jalan dan lokasi titik penjemputan warga. Berikut gambaran lokasi jalan pada Tabel 1 dari *Google Maps*:


Gambar 2. Rute Jl. Terong 1 s/d 6 dan Jl. Terong Pipit 1 s/d 12 pada *Google Maps*

Graf yang telah dibuat, diberi bobot pada setiap sisi dan titiknya. Bobot pada sisi menunjukkan jarak jalan antar persimpangan dalam satuan meter dan nilai pada titik menunjukkan jumlah warga. Sebagai contoh, label $v_{24}$, 4 pada titik memiliki arti yaitu titik $v_{24}$ terdapat jumlah warga sebanyak 4 orang dan label 154 pada sisi memiliki arti yaitu terdapat jarak yang menghubungkan 2 titik dengan besaran 154 meter. Jumlah warga yang digunakan adalah 10% dari 435 warga terdampak pada Jl. Terong yaitu 43 warga dan 10% dari 434 warga terdampak pada Jl. Terong Pipit yaitu 43 warga. Titik berwarna biru adalah titik awal yang merupakan persimpangan pertama sebelum memasuki Jl. Terong. Titik berwarna merah merupakan titik yang berada di Jl. Terong. Titik berwarna hijau merupakan titik yang berada di Jl. Terong Pipit. Empat puluh tiga warga pada Jl. Terong dan Jl. Terong Pipit dibagi secara random sebanyak jumlah titik pada jalan tersebut yaitu 13 titik pada Jl. Terong dan 15 titik pada Jl. Terong Pipit. Hasil random yang didapatkan digunakan sebagai bobot titik pada Jl. Terong dan Jl. Terong Pipit. Berikut graf yang telah diberi bobot pada titik dan sisinya:




Gambar 3. Graf berbobot G

## 2. Pembuatan Matriks Ketetanggaan

Graf berbobot pada Gambar 4.2 direpresentasikan dalam bentuk matriks ketetanggaan berbobot dengan notasi $A = a_{ij;i=1,2,3,\dots,29;j=1,2,3,\dots,29}$. Berikut tabel titik terhubung dan bobot sisi yang menjadi nilai-nilai elemen pada matriks A:

Tabel 2. Titik Terhubung dan Bobot Sisi

| No | Titik Terhubung | Bobot Sisi |
|---|---|---|
| 1. | $v_1v_2$ | 60 |
| 2. | $v_2v_3$ | 43 |
| 3. | $v_2v_4$ | 140 |
| 4. | $v_3v_5$ | 36 |
| 5. | $v_4v_{24}$ | 154 |
| 6. | $v_4v_{11}$ | 38 |
| 7. | $v_5v_6$ | 40 |
| 8. | $v_5v_7$ | 37 |
| 9. | $v_7v_8$ | 92 |
| 10. | $v_7v_9$ | 35 |
| 11. | $v_8v_{10}$ | 35 |
| 12. | $v_9v_{10}$ | 95 |
| 13. | $v_9v_{11}$ | 35 |
| 14. | $v_{10}v_{12}$ | 45 |
| 15. | $v_{11}v_{12}$ | 72 |
| 16. | $v_{11}v_{16}$ | 36 |
| 17. | $v_{12}v_{13}$ | 38 |
| 18. | $v_{13}v_{14}$ | 36 |
| 19. | $v_{13}v_{15}$ | 126 |
| 20. | $v_{14}v_{16}$ | 89 |
| 21. | $v_{14}v_{18}$ | 37 |
| 22. | $v_{16}v_{17}$ | 36 |
| 23. | $v_{17}v_{18}$ | 73 |
| 24. | $v_{17}v_{22}$ | 97 |
| 25. | $v_{17}v_{19}$ | 62 |
| 26. | $v_{18}v_{19}$ | 60 |
| 28. | $v_{19}v_{21}$ | 92 |
| 29. | $v_{20}v_{21}$ | 64 |
| 30. | $v_{20}v_{23}$ | 57 |
| 31. | $v_{20}v_{27}$ | 35 |
| 32. | $v_{21}v_{22}$ | 62 |
| 33. | $v_{22}v_{25}$ | 32 |
| 34. | $v_{24}v_{25}$ | 40 |
| 35. | $v_{25}v_{26}$ | 92 |
| 36. | $v_{26}v_{27}$ | 56 |
| 37. | $v_{27}v_{28}$ | 79 |
| 38. | $v_{28}v_{29}$ | 62 |
| 39. | $v_{29}v_{27}$ | 145 |

Pada matriks A, $a_{ij}$ bernilai 1 jika $v_i$ terhubung atau bertetangga dengan $v_j$, sedangkan $a_{ij}$ bernilai 0 saat $v_i$ tidak terhubung dengan $v_j$ atau ketika $v_i$ sama dengan $v_j$. Berdasarkan Tabel 2, matriks A direpresentasikan sebagai berikut.

$$
\begin{bmatrix}
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0
\end{bmatrix}
$$

## 3. Penentuan Lintasan Terpendek Menggunakan Algoritma Dijkstra dan Algoritma Greedy

Penelitian ini menentukan lintasan optimal untuk mengevakuasi warga yang ada di Jl. Terong dan Jl. Terong Pipit, Kelurahan Sempaja Timur. Lintasan terpilih adalah lintasan optimal yang memiliki nilai densitas terbesar dengan jumlah total warga yang dievakuasi tidak lebih dari total maksimal kapasitas perahu. Total maksimal warga yang dapat dievakuasi menggunakan perahu yaitu 10 orang.

Algoritma Dijkstra yang digunakan pada penelitian ini diberikan langkah yang berbeda dari langkah umum Algoritma Dijkstra. Perbedaannya terdapat pada nilai yang dibandingkan pada setiap langkah. Langkah umum Algoritma Dijkstra mencari lintasan dengan jarak tempuh terpendek sedangkan pada penelitian ini mencari lintasan dengan densitas terbesar. Setelah mendapatkan lintasan dengan nilai densitas terbesar dan memenuhi batas maksimal perahu, nilai bobot pada titik diubah menjadi 0 (nol) yang menandakan warga pada titik tersebut sudah dievakuasi dan perhitungan diulangi hingga semua bobot pada titik telah bernilai 0 (nol) atau sudah tidak ada warga yang harus dievakuasi. Penggabungan Algoritma Dijkstra dan Algoritma Greedy untuk penentuan rute optimal:

a. Inisialisasi titik $(v)$.
b. Inisialisasi jarak antar titik $(e)$.
c. Inisialisasi bobot pada titik $(w)$.
d. Menentukan titik awal.
e. Memberi label 1 pada titik awal dan label 0 pada titik lainnya.
f. Memberi nilai $d(v_i) = -\infty$ pada semua titik $v_i$ .
g. Memberi nilai $d(V) = \frac{s(w)}{s(e)}$, dengan $s(w) = w_i$ dan $s(e) = 0$.
h. Memberi nilai $d(v_i) = baru$, yaitu $maks\left\{d(vi)\ lama, (s(w) + w_i)/\left(s(e) + e_{i,j}\right)\right\}$, dengan syarat $s(w) + w_i \leq C$ untuk setiap titik $v_i$.
i. Mencari nilai $d(v_i)$ maksimum selain -∞ dari semua titik yang berlabel 0. Jika

terdapat nilai yang sama, diambil $d(v_i)$ dengan nilai $s(w) + w_i$ terbesar.

j. Merubah nilai $d(v_i)$ terpilih menjadi $d(V)$ dan memberi label 1 pada $v_i$ terpilih.

k. Merubah nilai $w_i$ dari setiap titik yang dilewati pada lintasan terpilih dengan nilai 0 dan mengembalikan nilai $w_i$ yang tidak dilewati pada lintasan terpilih.

l. Mengulangi langkah h sampai k hingga tidak ada lagi $d(v_i)$ yang memenuhi langkah i.

m. Memilih nilai $d(v_i)$ maksimum dari semua titik dan merubah nilai $w_i$ dari setiap titik yang dilewati pada lintasan terpilih secara permanen dengan nilai 0 (nol).

n. Menyimpan lintasan terpilih.

o. Mengembalikan nilai $w_i$ yang tidak terpilih.

p. Mengulangi langkah e sampai o hingga semua $w_i$ bernilai 0 (nol).

Berikut perhitungan penentuan lintasan optimal menggunakan Algoritma Dijkstra dan Algoritma Greedy:

1. Perhitungan 1

Pada penentuan lintasan optimal pertama yang selanjutnya disebut perhitungan 1, bobot di setiap titik graf G belum ada yang bernilai 0 selain titik awal. Titik yang menjadi titik awal pada perhitungan ini adalah titik $v_1$, sehingga titik $v_1$ diberi label 1. Nilai densitas lintasan menuju titik $v_i$ $(d(v_i))$ untuk semua titik diberi nilai -∞, nilai densitas lintasan terpilih $(d(V)) = -\infty$, total bobot lintasan terpilih $(s(w)) = 0$, dan total jarak lintasan terpilih $(s(e)) = 0$.

Iterasi 1:

Pada iterasi 1, titik $v_1$ hanya bertetangga dengan $v_2$.

$$d(v_2)\ baru = maks\left\{d(v_2)\ lama; \frac{(s(w)+w_2)}{(s(e)+e_{2,1})}\right\}$$
$$= maks\left\{-\infty; \frac{(0+2)}{(0+60)}\right\}$$
$$= maks\{-\infty;\ 0,3333\}$$
$$=0,3333$$

Berdasarkan Iterasi 1 diperoleh bahwa nilai maksimum $d(v_i)$ dari semua titik berlabel 0 pada iterasi 1 adalah $d(v_2)$ = 0,3333, maka nilai $d(V)$ menjadi 0,3333, dimana $s(w) = 2$ dan $s(e) = 60$ dan terdapat nilai $d(v_i)$ dengan label 0 yang bernilai selain -∞, maka dilanjut dengan iterasi 2. Lintasan terpilih untuk iterasi 2 yaitu $(v_1, v_2)$. Nilai $w_1$ dan $w_2$ dirubah menjadi 0 dan $v_2$ mendapat label 1.

Iterasi 2:

Pada iterasi 2, titik yang bertetangga dengan $v_2$ adalah $v_1$, $v_3$ dan $v_4$.

$$d(v_1)\ baru = maks\left\{d(v_1)\ lama; \frac{(s(w)+w_1)}{(s(e)+e_{1,2})}\right\}$$
$$= maks\left\{-\infty; \frac{(2+0)}{(60+60)}\right\}$$
$$= maks\{-\infty;\ 0,01667\}$$
$$= 0,01667$$

$$d(v_3)\ baru = maks\left\{d(v_3)\ lama; \frac{(s(w)+w_3)}{(s(e)+e_{3,2})}\right\}$$
$$= maks\left\{-\infty; \frac{(2+4)}{(60+43)}\right\}$$
$$= maks\{-\infty; 0,05825\}$$
$$= 0,05825$$

$$d(v_4)\ baru = maks\left\{d(v_4)\ lama; \frac{(s(w)+w_4)}{(s(e)+e_{4,2})}\right\}$$

$= maks\left\{-\infty; \frac{(2+5)}{(60+140)}\right\}$
$= maks\{-\infty; 0{,}035\}$
$= 0{,}035$

Berdasarkan iterasi 2 diperoleh bahwa nilai maksimum $d(v_i)$ dari semua titik berlabel 0 pada iterasi 2 adalah $d(v_3)$ = 0,0582, maka nilai $d(V)$ menjadi 0,0582, dimana $s(w) = 6$ dan $s(w) = 103$ dan terdapat nilai $d(v_i)$ dengan label 0 yang bernilai selain -∞, maka dilanjut dengan iterasi 3. Lintasan terpilih untuk iterasi 3 yaitu ($v_1$,$v_2$,$v_3$). Nilai $w_1$, $w_2$ dan $w_3$ dirubah menjadi 0 dan $v_3$ mendapat label 1.

Iterasi 3:

Pada iterasi 3, titik yang bertetangga dengan $v_3$ adalah $v_2$ dan $v_5$

$d(v_2)\ baru = maks\left\{d(v_2)\ lama; \frac{(s(w)+w_2)}{(s(e)+e_{2,3})}\right\}$
$= maks\left\{0{,}03333; \frac{(6+0)}{(103+43)}\right\}$
$= maks\{0{,}03333; 0{,}0411\}$
$=0{,}0411$

$d(v_5)\ baru = maks\left\{d(v_5)\ lama; \frac{(s(w)+w_5)}{(s(e)+e_{5,3})}\right\}$
$= maks\left\{-\infty; \frac{(6+4)}{(103+36)}\right\}$
$= maks\{-\infty; 0{,}07194\}$
$= 0{,}07194$

Berdasarkan iterasi 3 diperoleh bahwa nilai maksimum $d(v_i)$ dari semua titik berlabel 0 pada iterasi 3 adalah $d(v_3)$ = 0,07194, maka nilai $d(V)$ menjadi 0,07194, dimana $s(w)$ = 10 dan $s(w)$ = 139 dan terdapat nilai $d(v_i)$ dengan label 0 yang bernilai selain -∞, maka dilanjut dengan iterasi 4. Lintasan terpilih untuk iterasi 3 yaitu ($v_1$,$v_2$,$v_3$,$v_5$). Nilai $w_1$, $w_2$, $w_3$ dan $v_5$ dirubah menjadi 0 dan $v_5$ mendapat label 1.

Iterasi 4:

Pada iterasi 4, titik yang bertetangga dengan $v_5$ adalah $v_3$,$v_6$ dan $v_7$. Karena $(s(w) + w_3) = 10 + 4 \geq W = 10$, $(s(w) + w_6) = 10 + 2 \geq W = 10$ dan $(s(w) + w_7) = 10 + 2 \geq W = 10$, maka nilai $d(v_3)$, $d(v_6)$ dan $d(v_7)$ tidak berubah.

Berdasarkan iterasi 4 diperoleh bahwa nilai maksimum $d(v_i)$ dari semua titik berlabel 0 pada iterasi 4 adalah $d(v_4)$ = 0,035, maka nilai $d(V)$ menjadi 0,035, dimana $s(w) = 7$ dan $s(e) = 200$ dan terdapat nilai $d(v_i)$ dengan label 0 yang bernilai selain -∞, maka dilanjut dengan iterasi 5. Lintasan terpilih untuk iterasi 5 yaitu ($v_1$,$v_2$,$v_4$). Nilai $w_1$, $w_2$ dan $w_4$ dirubah menjadi 0 dan $v_4$ mendapat label 1.

Iterasi 5:

Pada iterasi 5, titik yang bertetangga dengan $v_4$ adalah $v_2$, $v_{11}$ dan $v_{24}$. Karena $(s(w) + w_{24}) = 7 + 4 \geq W = 10$, maka nilai $d(v_{24})$ tidak berubah.

$d(v_2)\ baru = maks\left\{d(v_2)\ lama; \frac{(s(w)+w_2)}{(s(e)+e_{2,4})}\right\}$
$= maks\left\{0{,}0411; \frac{(7+2)}{(200+140)}\right\}$
$= maks\{0{,}0411;\ 0{,}02647\}$
$= 0{,}0411$

$d(v_{11})\ baru = maks\left\{d(v_{11})\ lama; \frac{(s(w)+w_{11})}{(s(e)+e_{11,4})}\right\}$
$= maks\left\{-\infty; \frac{(7+2)}{(200+38)}\right\}$
$= maks\{-\infty; 0{,}03781\}$
$= 0{,}03781$

Berdasarkan Tabel iterasi 5 diperoleh bahwa nilai maksimum $d(v_i)$ dari semua titik berlabel 0 pada iterasi 5 adalah $d(v_{11}) = 0{,}03781$, maka nilai $d(V)$ menjadi 0,03781, dimana $s(w) = 9$ dan $s(e) = 238$ dan terdapat nilai $d(v_i)$ dengan label 0 yang bernilai selain -∞, maka dilanjut dengan iterasi 6. Lintasan terpilih untuk iterasi 6 yaitu ($v_1$,$v_2$,$v_4$,$v_{11}$). Nilai $w_1$, $w_2$, $w_4$ dan $w_{11}$ dirubah menjadi 0 dan $v_{11}$ mendapat label 1.
Iterasi 6:

Pada iterasi 6, titik yang bertetangga dengan $v_{11}$ adalah $v_4$, $v_9$, $v_{12}$ dan $v_{16}$. Karena $(s(w) + w_4) = 9 + 5 \geq W = 10$, $(s(w) + w_9) = 9 + 2 \geq W = 10$, $(s(w) + w_{12}) = 9 + 4 \geq W = 10$ dan $(s(w) + w_{16}) = 9 + 3 \geq W = 10$, maka nilai $d(v_4)$, $d(v_9)$, $d(v_{12})$ dan $d(v_{16})$ tidak berubah.

Pada perhitungan 1, iterasi terakhir yaitu iterasi ke-6. Ketika iterasi ke-6 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_8) = 0{,}07194$, dimana $s(w) = 10$ dan $s(e) = 139$. Lintasan optimal terpilih pada perhitungan pertama yaitu ($v_1$,$v_2$,$v_3$,$v_5$) dengan jarak tempuh lintasan adalah 139 meter dan total warga yang dievakuasi yaitu 10 orang. Nilai $w_1$, $w_2$, $w_3$, $w_5$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_1$,$v_2$,$v_3$,$v_5$ telah dievakuasi.

2. Perhitungan 2

Pada penentuan lintasan optimal pertama yang selanjutnya disebut perhitungan 1, iterasi terakhir yaitu iterasi ke-13. Ketika iterasi ke-13 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_{16}) = 0{,}0365$, dimana $s(w) = 10$ dan $s(e) = 274$. Lintasan optimal terpilih pada perhitungan kedua yaitu ($v_1$, $v_2$, $v_4$, $v_{11}$, $v_{16}$) dengan jarak tempuh 274 meter dan total warga yang dievakuasi yaitu 10. Nilai $w_4$, $w_{11}$, $w_{16}$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_4$, $v_{11}$, $v_{16}$ telah dievakuasi.

3. Perhitungan 3

Pada penentuan lintasan optimal ketiga yang selanjutnya disebut perhitungan 3, iterasi terakhir yaitu iterasi ke-21. Ketika iterasi ke-21 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_8) = 0{,}02933$, dimana $s(w) = 10$ dan $s(e) = 341$. Lintasan optimal terpilih pada perhitungan ketiga yaitu ($v_1$, $v_2$, $v_3$, $v_5$, $v_7$, $v_9$, $v_{10}$, $v_8$) dengan jarak tempuh 341 meter dan total warga yang dievakuasi yaitu 10. Nilai $w_7$, $w_9$, $w_{10}$, $w_8$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_7$, $v_9$, $v_{10}$, $v_8$ telah dievakuasi.

4. Perhitungan 4

Pada penentuan lintasan optimal keempat yang selanjutnya disebut perhitungan 4, iterasi terakhir yaitu iterasi ke-22. Ketika iterasi ke-22 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_{22}) = 0{,}02347$, dimana $s(w) = 10$ dan $s(e) = 426$.

5. Perhitungan 5

Pada penentuan lintasan optimal kelima yang selanjutnya disebut perhitungan 5, iterasi terakhir yaitu iterasi ke-22. Ketika iterasi ke-22 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_{13}) = 0{,}02332$, dimana $s(w) = 9$ dan $s(e) = 386$. Lintasan optimal terpilih pada perhitungan kelima yaitu ($v_1$, $v_2$, $v_3$, $v_5$, $v_7$, $v_8$, $v_{10}$, $v_{12}$, $v_{13}$) dengan jarak tempuh 386 meter dan total warga yang dievakuasi yaitu 9. Nilai $w_{12}$, $w_{13}$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_{12}$, $v_{13}$ telah dievakuasi.

6. Perhitungan 6

Pada penentuan lintasan optimal keenam yang selanjutnya disebut perhitungan 6, iterasi terakhir yaitu iterasi ke-23. Ketika iterasi ke-23 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_{19}) = 0{,}01927$, dimana $s(w) = 10$ dan $s(e) = 519$. Lintasan optimal terpilih pada perhitungan keenam yaitu ($v_1$, $v_2$, $v_3$, $v_5$, $v_7$, $v_8$, $v_{10}$, $v_{12}$, $v_{13}$, $v_{14}$, $v_{18}$, $v_{19}$) dengan jarak tempuh 519 meter dan total warga yang dievakuasi yaitu 10 nilai $w_{14}$, $w_{18}$, $w_{19}$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_{14}$, $v_{18}$, $v_{19}$ telah dievakuasi.

7. Perhitungan 7

Pada penentuan lintasan optimal ketujuh yang selanjutnya disebut perhitungan 7, iterasi terakhir yaitu iterasi ke-25. Ketika iterasi ke-25 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_{23}) = 0.01711$, dimana $s(w) = 9$ dan $s(e) = 526$. Lintasan optimal terpilih pada perhitungan ketujuh yaitu ($v_1$, $v_2$, $v_4$, $v_{11}$, $v_{16}$, $v_{17}$, $v_{19}$, $v_{20}$, $v_{23}$) dengan jarak tempuh 526 meter dan total warga yang dievakuasi yaitu 9. Nilai $w_{17}$, $w_{20}$, $w_{23}$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_{17}$, $v_{20}$, $v_{23}$ telah dievakuasi.

8. Perhitungan 8

Pada penentuan lintasan optimal kedelapan yang selanjutnya disebut perhitungan 8, iterasi terakhir yaitu iterasi ke-29. Ketika iterasi ke-29 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_{29}) = 0{,}01278$, dimana $s(w) = 9$ dan $s(e) = 704$. Lintasan optimal terpilih pada perhitungan kedelapan yaitu ($v_1$, $v_2$, $v_4$, $v_{11}$, $v_{16}$, $v_{17}$, $v_{19}$, $v_{21}$, $v_{20}$, $v_{27}$, $v_{28}$, $v_{29}$) dengan jarak tempuh 704 meter dan total warga yang dievakuasi yaitu 9. Nilai $w_{21}$, $w_{27}$, $w_{28}$, $w_{29}$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_{21}$, $v_{27}$, $v_{28}$, $v_{29}$ telah dievakuasi.

9. Perhitungan 9

Pada penentuan lintasan optimal kesembilan yang selanjutnya disebut perhitungan 9, iterasi terakhir yaitu iterasi ke-29. Ketika iterasi ke-29 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_6) = 0{,}01117$, dimana $s(w) = 2$ dan $s(e) = 179$. Lintasan optimal terpilih pada perhitungan kesembilan yaitu ($v_1$,$v_2$,$v_3$,$v_5$,$v_6$) dengan jarak tempuh 179 meter dan total warga yang dievakuasi yaitu 2. Nilai $w_6$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_6$ telah dievakuasi.

10. Perhitungan 10

Pada penentuan lintasan optimal kesepuluh yang selanjutnya disebut perhitungan 10, iterasi terakhir yaitu iterasi ke-29. Ketika iterasi ke-29 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_{26}) = 0{,}00942$, dimana $s(w) = 5$ dan $s(e) = 531$. Lintasan optimal terpilih pada perhitungan kesepuluh yaitu ($v_1$, $v_2$, $v_4$, $v_{11}$, $v_{16}$, $v_{17}$, $v_{19}$, $v_{22}$, $v_{25}$, $v_{26}$) dengan jarak tempuh 531 meter dan total warga yang dievakuasi yaitu 5. Nilai $w_{26}$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_{26}$ telah dievakuasi.

11. Perhitungan 11

Pada penentuan lintasan optimal kesebelas yang selanjutnya disebut perhitungan 11, iterasi terakhir yaitu iterasi ke-29. Ketika iterasi ke-29 semua nilai $d(v_i)$ dari titik berlabel 0 bernilai -∞, sehingga dipilih nilai maksimum $d(v_i)$ dari semua titik sebagai lintasan paling optimal yaitu $d(v_{15}) = 0{,}00391$, dimana $s(w) = 2$ dan $s(e) = 512$. Lintasan optimal terpilih pada perhitungan kesebelas yaitu ($v_1$, $v_2$, $v_3$, $v_5$, $v_7$, $v_8$, $v_{10}$, $v_{12}$, $v_{13}$, $v_{15}$) dengan jarak tempuh 512 meter dan total warga yang dievakuasi yaitu 2. Nilai $w_{15}$ dirubah menjadi 0 (nol) secara permanen yang menandakan bahwa warga pada titik $v_{15}$ telah dievakuasi.

Perhitungan lintasan optimal menggunakan Algoritma Dijkstra dan Algoritma Greedy pada graf $G$ yang telah dilakukan mendapatkan hasil yaitu 11 lintasan optimal. Lintasan-lintasan optimal yang diperoleh dari perhitungan 1 hingga perhitungan 11 menunjukkan urutan jalur evakuasi yang dapat digunakan saat evakuasi banjir di Jl. Terong dan Jl. Terong Pipit, Kelurahan Sempaja Timur, Kota Samarinda. Berikut ringkasan lintasan optimal yang diperoleh dari perhitungan 1 hingga perhitungan 11 yang dapat dilihat pada Tabel 3.

Tabel 3. Hasil Perhitungan Lintasan Optimal pada Graf $G$

| No. | Lintasan Optimal | Total Bobot | Jarak Lintasan |
|---|---|---|---|
| 1. | $v_1,v_2,v_3,v_5$ | 10 | 139 |
| 2. | $v_1,v_2,v_4,v_{11},v_{16}$ | 10 | 274 |
| 3. | $v_1,v_2,v_3,v_5,v_7,v_9,v_{10},v_8$ | 10 | 341 |
| 4. | $v_1,v_2,v_4,v_{24},v_{25},v_{22}$ | 10 | 426 |
| 5. | $v_1,v_2,v_3,v_5,v_7,v_8,v_{10},v_{12},v_{13}$ | 9 | 386 |
| 6. | $v_1,v_2,v_3,v_5,v_7,v_8,v_{10},v_{12},v_{13},v_{14},v_{18},v_{19}$ | 10 | 519 |
| 7. | $v_1,v_2,v_4,v_{11},v_{16},v_{17},v_{19},v_{20},v_{23}$ | 9 | 526 |
| 8. | $v_1,v_2,v_4,v_{11},v_{16},v_{17},v_{19},v_{21},v_{20},v_{27},v_{28},v_{29}$ | 9 | 704 |
| 9. | $v_1,v_2,v_3,v_5,v_6$ | 2 | 179 |
| 10. | $v_1,v_2,v_4,v_{11},v_{16},v_{17},v_{19},v_{22},v_{25},v_{26}$ | 5 | 531 |
| 11. | $v_1,v_2,v_3,v_5,v_7,v_8,v_{10},v_{12},v_{13},v_{15}$ | 2 | 512 |

## KESIMPULAN

Berdasarkan hasil dan pembahasan dari penelitian ini dapat disimpulkan bahwa penggabungan Algoritma Dijkstra dan Algoritma Greedy dapat digunakan pada kasus penentuan jalur evakuasi banjir di Jl. Terong dan Jl. Terong Pipit, Kelurahan Sempaja Timur, Kota Samarinda. Pada penelitian ini didapatkan 11 lintasan optimal dengan total bobot tidak lebih dari batas maksimum yaitu 10. Lintasan-lintasan optimal tersebut yaitu ($v_1$, $v_2$, $v_3$, $v_5$), ($v_1$, $v_2$, $v_4$, $v_{11}$, $v_{16}$), ($v_1$, $v_2$, $v_3$, $v_5$, $v_7$, $v_9$, $v_{10}$, $v_8$), ($v_1$, $v_2$, $v_4$, $v_{24}$, $v_{25}$, $v_{22}$), ($v_1$, $v_2$, $v_3$, $v_5$, $v_7$, $v_8$, $v_{10}$, $v_{12}$, $v_{13}$), ($v_1$, $v_2$, $v_3$, $v_5$, $v_7$, $v_8$, $v_{10}$, $v_{12}$, $v_{13}$, $v_{14}$, $v_{18}$, $v_{19}$), ($v_1$, $v_2$, $v_4$, $v_{11}$, $v_{16}$, $v_{17}$, $v_{19}$, $v_{20}$, $v_{23}$), ($v_1$, $v_2$, $v_4$, $v_{11}$, $v_{16}$, $v_{17}$, $v_{19}$, $v_{21}$, $v_{20}$, $v_{27}$, $v_{28}$, $v_{29}$), ($v_1$, $v_2$, $v_3$, $v_5$, $v_6$), ($v_1$, $v_2$, $v_4$, $v_{11}$, $v_{16}$, $v_{17}$, $v_{19}$, $v_{22}$, $v_{25}$, $v_{26}$), dan ($v_1$, $v_2$, $v_3$, $v_5$, $v_7$, $v_8$, $v_{10}$, $v_{12}$, $v_{13}$, $v_{15}$). Lintasan-lintasan optimal tersebut merupakan urutan evakuasi banjir yang dapat digunakan di Jl. Terong dan Jl. Terong Pipit, Kelurahan Sempaja Timur, Kota Samarinda agar seluruh warga dapat dievakuasi dan proses evakuasi berjalan optimal.

## DAFTAR PUSTAKA

[1] Harahap, M. K., dan Khairina N. (2017). Pencarian Jalur Terpendek dengan Algoritma Dijkstra. *Jurnal & Penelitian Teknik Informatika*, 2(2), 18-23.

[2] Kementerian Pekerjaan Umum dan Perumahan Rakyat (2019, Juni). Bencana Banjir Kota Samarinda Provinsi Kalimantan Timur. [Laporan Kondisi Terkini]. Samarinda. KPUPR.

[3] Lin, B., Liu, S., Lin, R., Wu, J., Wang, J., dan Liu, C. (2017). Modeling the 0-1 Knapsack Problem in Cargo Flow Adjustment. *Symmertry*, 9(7), 118.

[4] Munir, R. (2010). *Matematika Diskrit*. Bandung : Informatika.

[5] Puspika, B. N., Rachmat C., A., dan Kurniawan, E. (2012). Implementasi Algoritma Dijkstra Dalam Penentuan Jalur Terpendek Di Yogyakarta Menggunakan GPS Dan Qt Geolocation. *Informatika*, 8(2), 141-149.

[6] Rachmawati, D. dan Candra, A. (2013). Implementasi Algoritma Greedy untuk Menyelesaikan Masalah Knapsack Problem. *Jurnal Saintikom*, 12(3), 185-192.

[7] Rahmawati. (2018). Penerapan Fuzzy Linear Programming Pada Optimasi Pembangunan Rumah Susun (Rusun) Di Kawasan Pondok Cina Provinsi Jawa Barat. *Jurnal Sains Matematika dan Statistika*, 4(1), 79-89.

[8] Rosen, K. H. (2019). *Discrete Mathematics and Its Applications*, Eighth Edition. New York : McGraw-Hill Education.

[9] Sualang, C. S., Kaunang MT., Ir. S. T. G., dan Lumenta ST. MT., A. S. M. (2015). Pembuatan Prototype Aplikasi Pentarifan Rental Mobil Dengan Metode Polinomial Lagrange. *E-Journal Teknik Elektro dan Komputer*. 54-59.

[10] Voloch, N. (2017). A Complex Problem of Knapsack and Shortest Paths on Weighted Graphs. *International Journal of Advanced Computational Engineering and Networking*, 5(1), 31-34.

[11] Walikota Samarinda (2014, Februari). Peraturan Walikota (PERWALI) tentang Penyelenggaraan Penanggulangan, Pegaturan Pendanaan Serta Penetapan Besaran Santunan/Bantuan Korban Bencana. [Peraturan Walikota]. Samarinda. Pemerintah Daerah Kota Samarinda.