PENGEMBANGAN ALGORITMA PRIM UNTUK MENENTUKAN *MINIMUM SPANNING FOREST*

Hari Sumardi[1], Afnaria[2], Suvriadi Panggabean[3]
[1]Universitas Bengkulu, Jl. WR Supratman, Bengkulu
[2]Universitas Islam Sumatera Utara, Jl. Sisingamangaraja, Medan
[3]Universitas Muhammadiyah Sumatera Utara, Jl. Kapt. Mukhtar Basri, Medan
[1]harisumardi@unib.ac.id

Abstrak

Minimum spanning tree (MST) merupakan salah satu permasalahan dalam teori graph. MST dari graph G adalah *spanning tree* dengan total bobot sisi terkecil pada suatu graph berbobot G yang terhubung. Terdapat dua algoritma klasik di dalam MST yakni algoritma Kruskal dan Prim. Kedua algoritma tersebut dapat menghasilkan sebuah MST. *Forest* merupakan graph yang terdiri dari beberapa tree. *Spanning forest* dari graph tak terhubung G merupakan forest yang dibangun dari graph G. *Minimum spanning forest* (MSF) dari graph G merupakan *spanning forest* dengan total bobot sisi terkecil atas semua *spanning forest* pada graph G. Penelitian ini bertujuan untuk menentukan MSF. Metode yang digunakan dalam penelitian ini berupa studi literatur. Pada hasil, penulis menyajikan sebuah algoritma MSF yang dikembangkan dari algoritma Prim.

Kata Kunci: MST; Forest; MSF; Algoritma Prim.

Abstract

Minimum spanning tree (MST) was one of the problems of graph theory. An MST of graph G was a spanning tree that has the smallest total edge-weight of a connected weighted graph G. There were two classic algorithms in MST which were the Kruskal's and the Prim's algorithm. Both of algorithms could construct an MST. Forest was a graph consisting of some trees. Spanning forest of a disconnected graph G was a forest which constructs of graph G. Minimum spanning forest (MSF) of graph G was a spanning forest that has the smallest total edge-weight over all spanning forests. This study aimed to determine the MSF. The method used in this research was literature study. On the results, we presented an MST algorithm which had constructed from the Prim's algorithm.

Keywords: MST; Forest; MSF; Prim's algorithm.

Pendahuluan

Pada tahun 1736, Leonhard Euler memberikan kontribusi yang sangat besar terhadap perkembangan ilmu matematika yakni teori graph. Teori graph memiliki objek yang diskrit, sehingga teori graph menjadi bagian dari matematika diskrit. Seiring perkembangan zaman, konsep teori graph diterapkan dalam beberapa bidang seperti fisika, kimia, biologi, teknik, ekonomi, dan kesehatan.

Graph $G(V, E)$ adalah suatu objek tak kosong yang memuat himpunan titik berhingga yang dinotasikan dengan $V(G)$ dan himpunan sisi yang berupa pasangan tak berurut dari titik-titik berbeda di $V(G)$ yang dinotasikan dengan $E(G)$ (Brualdi, 2009). Sisi yang dimaksud adalah sisi yang menghubungkan suatu titik x dengan titik y pada graph G. Apabila graph $G(V, E)$ memuat sisi berbobot disebut sebagai graph berbobot. Lintasan dari titik u ke titik v dalam suatu graph adalah barisan titik dan sisi yang menghubungkan titik u dan v pada graph G. Suatu graph dikatakan terhubung apabila untuk setiap titik x dan y terdapat lintasan yang menghubungkannya. Jika titik awal dan titik akhir pada lintasan adalah sama maka disebut sebagai lintasan tertutup atau *cycle*. Pada kehidupan sehari-hari, graph dapat diilustrasikan sebagai sebuah peta, dimana titik yang ada adalah persimpangan jalan dan sisinya adalah jalan serta jarak antara dua titik adalah bobot dari sisinya.

Suatu graph terhubung yang *acyclic* disebut dengan *Tree*. Dengan kata lain, *tree* merupakan graph terhubung tanpa memuat *cycle*. Karena *tree* tidak memuat *cycle*, tentunya *tree* dengan n titik memiliki $(n - 1)$ buah sisi. Suatu *tree* dapat dibentuk dari sebuah graph terhubung dan disebut dengan *Spanning Tree*. *Spanning tree* dari suatu graph terhubung sangatlah banyak. Secara khusus, Cayley telah memperlihatkan bahwa *spanning tree* dari graph lengkap K_n adalah n^{n-2} buah (Bóna, 2006).

Minimum Spanning Tree (MST) dari graph berbobot terhubung G adalah *spanning tree* dari G yang mempunyai total bobot sisi terkecil. MST dapat ditentukan dengan menggunakan beberapa algoritma seperti Kruskal dan Prim. Algoritma Kruskal melakukan pencarian MST dengan mengurutkan sisi-sisi dari bobot terkecil hingga terbesar terlebih dahulu. Kemudian memilih sisi terkecil diantara keseluruhan sisi yang ada. Apabila sisi-sisi yang terpilih tidak membentuk *cycle* maka akan diperolehlah MST. Sedangkan algoritma Prim melakukan pencarian dengan menentukan titik awal. Kemudian mencari titik lain yang bertertangga dengan titik awal tersebut dan mempunyai bobot terkecil. Begitu seterusnya sehingga di peroleh MST. Algoritma Kruskal dan Prim akan menghasilkan tepat satu MST. Penelitian tentang MST telah dilakukan oleh (Li & Zhang, 2010), (Bergantios & Vidal-Puga, 2011), (Bollig, 2012), (Hassan, 2012), (Yasin & Afandi, 2014), (Murmu, 2015), (Hardianto, 2015), (Sam & Yuliani, 2016), (Coletti, 2016), (Sril et al., 2019), dan (Marpaung & Arnita, 2020).

Suatu *forest* merupakan graph *acyclic* dengan dengan sedikitnya dua *tree*. *Spanning forest* dari graph G merupakan *forest* yang dibentuk dari suatu graph berbobot G dengan k-komponen. *Minimum spanning forest* (MSF) merupakan *spanning forest* yang mempunyai total bobot sisi terkecil atas semua *spanning forest*. Penelitian tentang MSF telah banyak dilakukan.

Diantaranya: (Kalateh Ahani et al., 2020) menentukan penyelesaian MSF dengan kendala *reliability*. (Pike et al., 2015) menentukan MSF berdasarkan metode klasifikasi. Pada penelitian ini, penulis mengembangkan algoritma Prim untuk menentukan MSF.

Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah studi literatur dengan pengembangan algoritma. Dimana penulis mempelajari sumber-sumber terkait tentang MST dan terdapat dua algoritma klasik yakni Kruskal dan Prim. Kemudian melakukan pengembangan algoritma Prim untuk menentukan MSF. Korte & Vygen (Korte & Vygen, 2018) memberikan algoritma Prim sebagai berikut.

Algoritma Prim

Input : A connected graph G, weights $w: E(G) \to \mathbb{R}$.

Output : Spanning tree T of minimum weight.

1. Choose $v \in V(G)$. Set $T \coloneqq (\{v\}, \emptyset)$.
2. While $V(T) \neq V(G)$ do:

   Choose an edge $e \in \delta_G\big(V(T)\big)$ of minimum weight. Set $T \coloneqq T + e$.

Pada algoritma Prim tersebut, input merupakan graph terhubung G dengan bobot sisi $E(G)$ merupakan bilangan riil. Outputnya adalah *spanning tree* T dengan total bobot sisi terkecil atau MST. Langkah pertama yakni memilih satu titik v dari himpunan $V(G)$ dan tetapkan T yang hanya berisi satu titik terpilih v. Langkah kedua, ketika himpunan titik di T belum merupakan keseluruhan himpunan titik di G, pilihlah tetangga dari v atau $\delta(v)$ dimana memiliki sisi e dengan bobot terkecil dan tetapkan $T \leftarrow T + e$. Lakukan langkah kedua ini berulang-ulang sampai himpunan titik di T sama dengan himpunan titik di G. Sehingga terbentuk *spanning tree* dengan total bobot sisi terkecil.

Hasil dan Pembahasan

Minimum Spanning Forest (MSF)

Misalkan G adalah graph dengan n titik dan k-komponen. *Spanning forest* merupakan keseluruhan subtree dari G dengan k-komponen. *Minimum spanning forest* (MSF) adalah *spanning forest* dengan total bobot sisi terkecil. Andaikan F merupakan himpunan *spanning forest*, MSF dapat dituliskan sebagai $\min(F)$. Algoritma berikut merupakan algoritma MSF yang dikembangkan dari algoritma Prim.

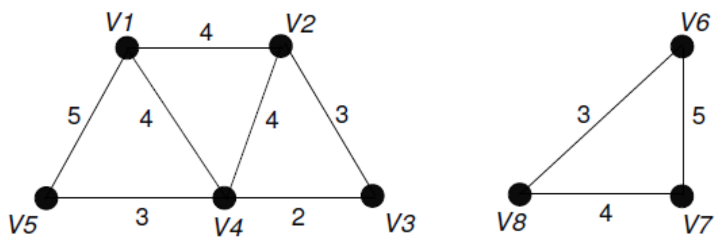
Algoritma MSF

Input : A disconected graph G which consist of k-components,
weights $w: E(G) \to \mathbb{R}$.

Output : Spanning forest F of minimum weight.

1. Set spanning tree $T_1, T_2, \ldots, T_k$ from k-components of G
2. Choose vertex $u \in V(G)$. Set $T_1 \coloneqq (\{u\}, \emptyset)$ and $V(F) = \{u\}$
3. While $V(F) \neq V(G)$ do:
   Choose minimum weight edge $e_i = \{u, v\}, v \in V(G) - V(F)$
   Set $V(F) \coloneqq V(F) + \{v\}$ and $T_1 := T_1 + e_i$
4. If there is no walk from vertex $x \in V(F)$ to $y \in V(G) - V(F)$,
   then do step 5
5. Choose vertex $y \in V(G)$ and $y \notin V(F)$
   Set $T_2 \coloneqq (\{y\}, T_1)$, and $V(F) \coloneqq V(F) + \{y\}$
6. While $V(F) \neq V(G)$ do:
   Choose minimum weight edge $e_j = \{v, y\}, y \in V(G) - V(F)$
   Set $V(F) \coloneqq V(F) + \{y\}$, and $T_2 \coloneqq T_2 + e_j$
7. If $V(F) = V(G)$, then stop iteration; Else repeat step 4.

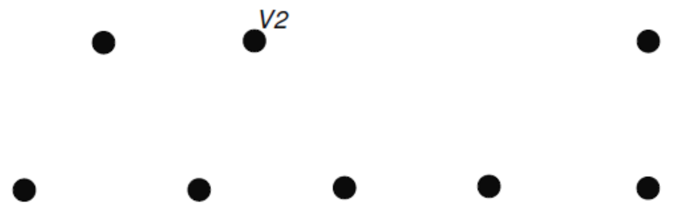
Pada algoritma MSF tersebut, mula-mula ditetapkan k buah spanning tree $T_1, T_2, \ldots, T_k$ yang masing-masing mewakili *subtree* pada setiap komponen. Kemudian carilah MST dari T_1 dengan cara yang sama seperti algoritma Prim. Setelah itu, cari MST untuk T_2, T_3 sampai T_k. Dan gabungkan hasil MST dari $T_1, T_2, \ldots, T_k$ sehingga diperoleh *spanning forest* dengan total bobot sisi terkecil atau *minimum spanning forest*. Berikut ini, disajikan penggunaan algoritma MST pada graph dengan 2-komponen.



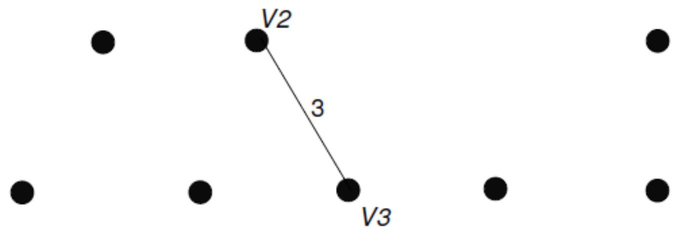
Gambar 1. Graph yang terdiri dari 8 titik dengan 2-komponen

Pada Gambar 1, akan ditentukan MSF dengan menggunakan Algoritma MSF. Langkah pertama, tetapkan T_1 untuk spanning tree pada komponen di sisi kiri dan T_2 untuk spanning tree pada komponen di sisi kanan. Langkah kedua, pilih satu titik di T_1, misalnya v_2 (Gambar 2) sebagai inisial titik di T_1. Langkah ketiga, pilih tetangga dari v_2 atau $\delta(v_2)$ dengan bobot terkecil

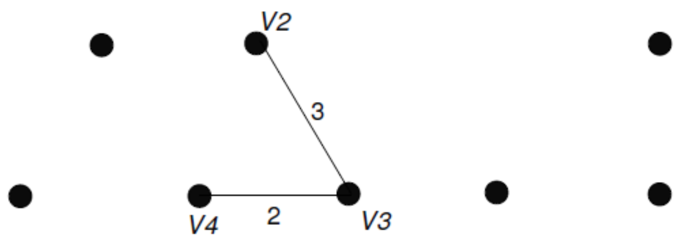
$(\min(3,4,4) = 3)$, sehingga tetangga terpilih adalah v_3 dengan bobot sisi 3 (Gambar 3). Selanjutnya cari tetangga dari v_2 dan v_3 dengan bobot terkecil dan tidak membentuk *cycle* $(\min(4,4,2) = 2$, sehingga tetangga terpilih adalah v_4 dengan bobot sisi 2 (Gambar 4). Lakukan langkah ketiga ini berulang-ulang sehingga diperoleh hasil seperti yang tampak pada Gambar 5, dan 6.



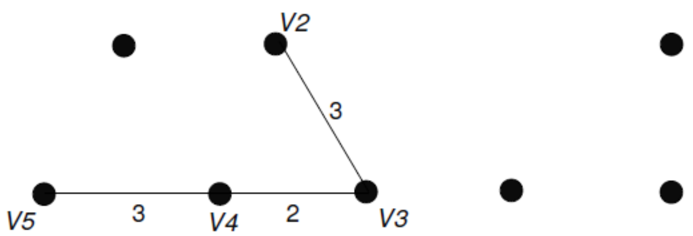
Gambar 2. Inisial titik di T_1



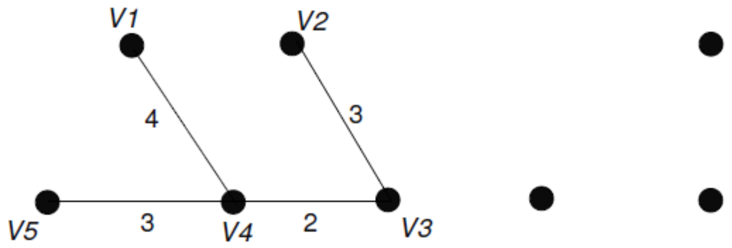
Gambar 3. Iterasi 1



Gambar 4. Iterasi 2

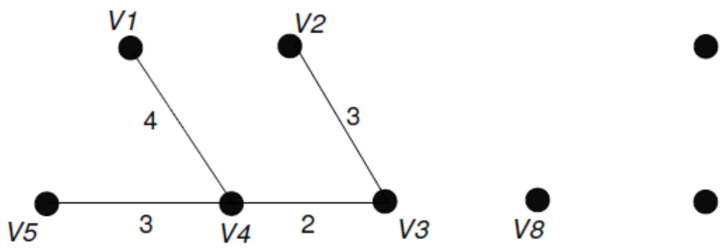


Gambar 5. Iterasi 3

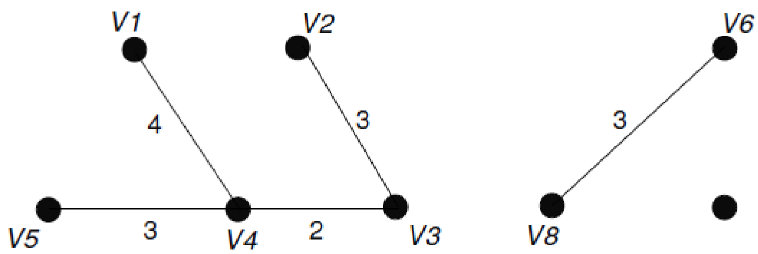


Gambar 6. Iterasi 4

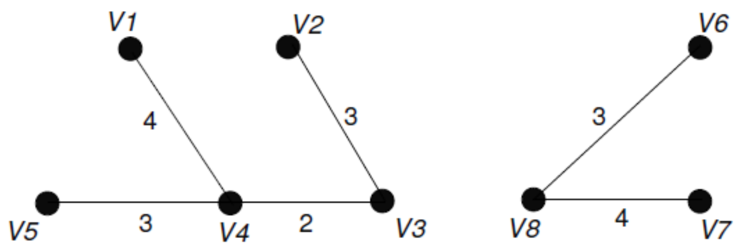
Pada Gambar 6, telah diperoleh MST pada T_1. Perhatikan bahwa v_1, v_2, v_3, v_4, v_5 tidak terhubung dengan v_6, v_7, v_8 (lihat langkah keempat). Oleh sebab itu, bentuklah MST selanjutnya yakni T_2. Langkah kelima, pilih satu titik di T_2 misalnya v_8 (Gambar 7) sebagai inisial titik di T_2. Langkah keenam, pilih tetangga dari v_8 atau $\delta(v_8)$ dengan bobot terkecil $(\min(3,4) = 3)$, sehingga tetangga terpilih adalah v_6 dengan bobot sisi 3 (Gambar 8). Kemudian cari tetangga dari v_6 dan v_8 dengan bobot terkecil dan tidak membentuk *cycle* $(\min(4,5) = 4$, sehingga tetangga terpilih adalah v_7 dengan bobot sisi 4 (Gambar 9) dan telah diperoleh MST pada T_2.



Gambar 7. Inisial titik di T_2

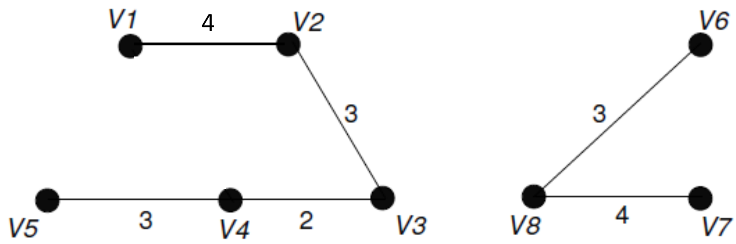


Gambar 8. Iterasi 5



Gambar 9. Iterasi 6

Sekarang telah diperoleh sebuah *spanning forest* seperti tampak pada Gambar 9. Karena *spanning forest* tersebut dihasilkan dari dua buah MST yakni T_1 dan T_2 sehingga disebut sebagai *minimum spanning forest* (MSF) dengan total bobot sisinya adalah $3 + 2 + 3 + 4 + 3 + 4 = 19$. Perlu diketahui bahwa subgraph MSF tidaklah tunggal. Dengan algoritma MSF, diperoleh juga subgraph MSF yang lain dari Gambar 1. Perhatikan Gambar 10 berikut ini.



Gambar 10. Subgraph MSF yang lain

Subgraph pada Gambar 9 dan 10 merupakan MSF dari Gambar 1 dengan total bobot sisi 19.

Simpulan dan Saran

Dari hasil dan pembahasan diperoleh bahwa algoritma MSF yang dikembangkan dari algoritma Prim dapat menentukan *minimum spanning forest* (MSF). MSF yang dihasilkan dari suatu graph tak terhubung tidaklah tunggal. Untuk graph berukuran besar hendaknya dilakukan dengan proses komputasi. Lebih lanjut, pada saat reboisasi khususnya di Provinsi Bengkulu hendaknya menggunakan konsep *minimum spanning forest*. Dimana pohon sebagai titik graph dan jarak antar tanaman merupakan bobot dari graph. Penentuan *minimum spanning forest* dilakukan sebelum proses penanaman agar tidak terjadi persaingan antar pohon dalam memperebutkan unsur hara.

Referensi

Bergantios, G., & Vidal-Puga, J. (2011). The folk solution and Boruvka's algorithm in minimum cost spanning tree problems. *Discrete Applied Mathematics*, *159*(2011), 1279–1283. https://doi.org/10.1016/j.dam.2011.04.017

Bollig, B. (2012). On symbolic OBDD-based algorithms for the minimum spanning tree problem. *Theoretical Computer Science*, 447, 2–12. https://doi.org/10.1016/j.tcs.2011.11.029

Bóna, M. (2006). *A Walk Through Combinatorics, An Introduction to Enumerat-ion and Graph Theory, Second Edition*. World Scientific, Singapore. https://doi.org/10.1142/6177

Brualdi, R. A. (2009). *Introductory Combinatorics (5th Edition)*. China Machine Press, China.

Coletti, P. (2016). Comparing minimum spanning trees of the Italian stock market using returns and volumes. *Physica A: Statistical Mechanics and Its Applications*, *463*(2016), 246–261. https://doi.org/10.1016/j.physa.2016.07.029

Hardianto, H. (2015). Penentuan Penurunan Tegangan Berdasarkan Minimum Spanning Tree Pada Jaringan Listrik Distribusi Primer. *Emitor: Jurnal Teknik Elektro*, *15*(01), 1–10. https://doi.org/10.23917/emitor.v15i1.1758

Hassan, M. R. (2012). An efficient method to solve least-cost minimum spanning tree (LC-MST) problem. *Journal of King Saud University - Computer and Information Sciences*, *24*(2012), 101–105. https://doi.org/10.1016/j.jksuci.2011.12.001

Kalateh Ahani, I., Salari, M., Hosseini, S. M., & Iori, M. (2020). Solution of minimum spanning forest problems with reliability constraints. *Computers and Industrial Engineering*, *142*, 106365. https://doi.org/10.1016/j.cie.2020.106365

Korte, B., & Vygen, J. (2018). *Kombinatorische Optimierung*. Springer, New York. https://doi.org/10.1007/978-3-662-57691-5

Li, W. V., & Zhang, X. (2010). Expected lengths of minimum spanning trees for non-identical edge distributions. *Electronic Journal of Probability*, *15*(5), 110–141. https://doi.org/10.1214/EJP.v15-735

Marpaung, F., & Arnita. (2020). Comparative of prim's and boruvka's algorithm to solve minimum spanning tree problems. *Journal of Physics: Conference Series*, 012043, 1–8. https://doi.org/10.1088/1742-6596/1462/1/012043

Murmu, M. K. (2015). A Distributed Approach to Construct Minimum Spanning Tree in Cognitive Radio Networks. *Procedia Computer Science*, 70, 166–173. https://doi.org/10.1016/j.procs.2015.10.066

Pike, R., Sechopoulos, I., & Fei, B. (2015). A minimum spanning forest based classification method for dedicated breast CT images. *Medical Physics*, *42*(11), 6190–6202. https://doi.org/10.1118/1.4931958

Sam, M., & Yuliani. (2016). Penerapan Algoritma Prim Untuk Membangun Pohon Merentang Minimum (Minimum Spanning Tree) Dalam Pengoptimalan Jaringan Transmisi Nasional Provinsi Sulawesi Selatan. *Jurnal Dinamika*, *07*(1), 50–61.

Sril, Y., Amalia, I. S., & Hasanah, A. (2019). Sril, Y., Amalia, I. S., & Hasanah, A. (2019). "PRIMATHRIC": Aplikasi Algortime Prim untuk Optimasi Penyediaan Akses Energi Listrik di Kabupaten Alor. *Jurnal Matematika Integratif*, *14*(2), 123–134. https://doi.org/10.24198/jmi.v14.n2.19271.123-134

Yasin, M., & Afandi, B. (2014). Simulasi Minimum Spanning Tree Graf Berbobot Menggunakan Algoritma Prim dan Algoritma Kruskal. *Eucazione*, *2*(2), 121–130.