Equator: Journal of Mathematical and Statistical Sciences (EJMSS)
Vol.2, No.1, Mei 2023, E-ISSN : 2964-1969

PENENTUAN POHON PERENTANG MINIMUM KE$-k$ DENGAN MENGGUNAKAN ALGORITMA k $-$MST

Erliana[1], Nilamsari Kusumastuti[2a], Fransiskus Fran[3]
Program Studi Matematika, Universitas Tanjungpura[1,2,3]
E-mail: nilamsari@math.untan.ac.id[a]

Informasi Artikel

Sejarah Artikel:

Diterima. Februari 2023
Disetujui Mei 2023
Dipublikasikan Mei 2023

Kata kunci:

Graf berbobot
Algoritma Prim
Alternatif pohon perentang minimum

ABSTRAK

Pohon perentang minimum (MST) pada graf berbobot G adalah pohon yang memuat semua simpul pada G dan memiliki bobot terkecil. Algoritma yang digunakan untuk menentukan MST adalah algoritma Prim. Secara umum, algoritma Prim hanya menghasilkan satu MST, atau lebih dengan bobot yang sama, sedangkan hasil tersebut mungkin tidak dapat diterapkan. Hal ini terjadi apabila terdapat kendala pada salah satu sisi MST saat diaplikasikan pada permasalahan di kehidupan sehari-hari. Oleh karenanya, diperlukan langkah lanjutan untuk memperoleh alternatif MST berikutnya. Artikel ini membahas penentuan MST ke-1 menggunakan algoritma Prim dan MST ke-k dengan $k = 2,3,\ldots$ menggunakan algoritma k-MST. Penelitian ini diawali dengan menentukan pohon perentang minimum pertama, misalnya T_1. Dari T_1, dicari T_2 dan seterusnya dengan mengganti satu per satu sisi dengan sisi lain pada G yang bukan merupakan sisi pada T_1. Kemudian bobot semua pohon perentang diurutkan berdasarkan bobot terkecil. Dengan mengulangi proses yang sama diperoleh pohon perentang minimum ke-k. Hasil pada graf yang dicontohkan dalam artikel ini, yaitu graf dengan 5 simpul dan 6 sisi, diperoleh 11 pohon perentang dengan bobot terkecil 11 dan bobot terbesar 18. Sedangkan pada contoh graf jaringan pipa dengan kendala satu sisi tidak dapat dilewati, diperoleh MST ke-4 yang memenuhi kendala dengan bobot total 22.

Penulis Korespondensi:

Nilamsari Kusumastuti,
Universitas Tanjungpura, Indonesia
Jalan. Prof. Dr. Hadari Nawawi, Pontianak Tenggara, Kota Pontianak, 78115
Email: nilamsari@math.untan.ac.id

1. PENDAHULUAN

Teori graf merupakan kajian ilmu matematika yang digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek-objek tersebut. Representasi dari graf adalah dengan menyatakan objek sebagai simpul, sedangkan hubungan antar objeknya dinyatakan sebagai sisi [1]. Konsep dari teori graf dapat digunakan untuk menyelesaikan permasalahan di berbagai bidang keilmuan, misalnya graf dapat digunakan untuk memodelkan senyawa karbon, memodelkan rangkaian listrik dan memodelkan jalan raya untuk mengatasi permasalahan lalu lintas.

Graf G adalah pasangan himpunan $(V(G), E(G))$, dengan $V(G)$ merupakan himpunan tidak kosong yang memuat elemen-elemen yang disebut simpul, yang dinotasikan dengan v, dan himpunan $E(G)$ yang elemen-elemennya disebut sisi, sedemikian sehingga setiap sisi $e = (u, v)$ dalam $E(G)$ menghubungkan simpul-simpul u dan v di $V(G)$ [2]. Suatu graf dikatakan graf sederhana jika tidak ada sisi yang menghubungkan simpul yang sama, misalnya $e = (u, u)$, atau tidak ada dua sisi yang menghubungkan simpul-simpul yang sama, misalnya $e_1 = (u, v)$ dan $e_2 = (u, v)$. Lintasan $u - v$ pada G adalah barisan terurut dari simpul-simpul yang diawali dengan simpul u ke simpul tujuan v, dimana semua simpul yang dilalui berlainan [3]. Lebih lanjut, lintasan yang bermula dan berakhir dengan simpul yang sama disebut siklus [2]. Dua buah simpul u dan v pada graf G terhubung jika terdapat lintasan dari u ke v. Lebih lanjut, graf G dikatakan terhubung jika setiap dua simpul berbeda pada graf G terhubung [3]. Graf T dikatakan subgraf dari G jika untuk semua $v \in V(T)$ dan $e \in E(T)$ maka $v \in V(G)$ dan $e \in E(G)$ [4].

Salah satu bahasan dalam teori graf yaitu pohon perentang minimum atau *minimum spanning tree* (*MST*). Pohon (*tree*) $T = ((V(T), E(T))$ merupakan graf sederhana yang terhubung dan tidak memuat siklus, sedangkan yang dimaksud dengan pohon perentang T dari graf G adalah subgraf yang berupa pohon dan memuat semua simpul dari G. Jadi T merupakan pohon perentang dari G, jika $V(T) = V(G)$ dan $E(T) \subseteq E(G)$. Jika setiap sisi dalam T dihubungkan dengan suatu bilangan asli, maka T disebut pohon perentang yang memiliki bobot, dengan bobot totalnya adalah jumlahan dari semua bobot sisi-sisi pada T.

Permasalahan pencarian *MST* adalah menentukan sisi-sisi dari suatu graf yang menghubungkan semua simpul pada graf dengan tidak membentuk siklus dan memiliki bobot minimum [1], [5], [6]. Terdapat beberapa algoritma yang dapat digunakan dalam pencarian *MST*, salah satunya yaitu algoritma Prim. Namun, algoritma ini hanya terbatas untuk memperoleh satu *MST,* atau lebih tetapi dengan bobot yang sama. Padahal, *MST* yang dihasilkan belum tentu dapat diterapkan. Sebagai contoh, pada pemasangan jaringan pipa distribusi air bersih seringkali jalur pipa yang direncanakan mengalami penyesuaian atau perubahan. Salah satu penyebabnya adalah terdapat bangunan yang menghalangi jalur pipa, sehingga perlu adanya alternatif jalur lainnya. Oleh karena itu, algoritma yang hanya menghasilkan satu *MST* berkembang menjadi algoritma yang dapat menghasilkan *MST* dengan bobot minimum berikutnya, atau *MST* ke-k. Algoritma ini disebut dengan algoritma k-*MST*.

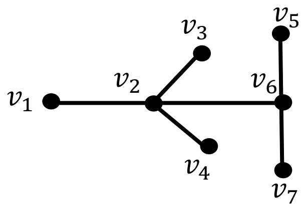
Algoritma k-*MST* dapat digunakan untuk menentukan pohon perentang dengan bobot minimum berikutnya. Konsep algoritma ini diperkenalkan oleh peneliti asal Department of computer and engineering college of engineering Trivandum, India bernama Amal P. M. dan Ajish Kumar K. S. pada tahun 2016 [7]. Penelitian tersebut berhasil menentukan MST ke-k.

Oleh sebab itu, permasalahan yang akan dibahas dalam artikel ini adalah bagaimana menentukan pohon perentang minimum pertama pada suatu graf dengan menggunakan algoritma Prim dan menentukan *MST* ke-k pada suatu graf dengan menggunakan algoritma k-*MST*.

2. METODE PENELITIAN

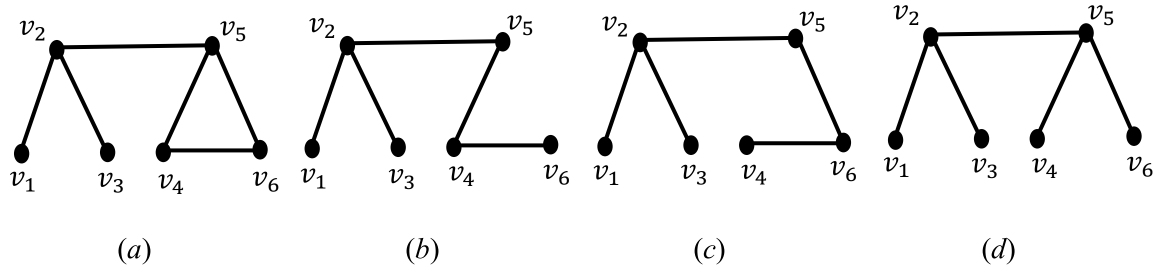
Metode yang digunakan pada artikel ini adalah studi literatur (kajian teori) yang berkaitan dengan teori terkait dengan masalah pohon perentang minimum, algoritma dasar dalam proses pencarian pohon perentang minimum dan algoritma pencarian k-MST. Sebelum melangkah dalam pembahasan, berikut ini dibahas tentang landasan teori yang diperlukan pada pembahasan.

Seperti yang telah disampaikan pada pendahuluan, graf sederhana $T = (V(T), E(T))$ disebut pohon jika terhubung dan tidak memuat siklus. Contoh pohon diperlihatkan pada Gambar 1.



Gambar 1. Pohon T dengan $V(T) = \{v_1, v_2, \dots, v_7\}$

Suatu pohon T merupakan pohon perentang dari G, jika $V(T) = V(G)$ dan $E(T) \subseteq E(G)$. Pohon perentang dari suatu graf tidak tunggal, seperti yang diilustrasikan pada Gambar 2

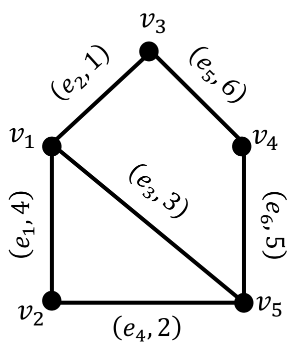


Gambar 2. (*a*) Graf G, (*b*)-(*d*) Pohon-Pohon Perentang graf G

Pohon-pohon perentang dari graf G dapat diperoleh dengan melakukan penghapusan salah satu sisi pada siklus. Banyaknya pohon pembangkit dari graf G dapat dicari menggunakan Teorema Pohon Matriks (*The Matrix Tree Theorem*) [8]. Banyaknya pohon pembangkit dari G dinotasikan dengan $|T_G|$.

Graf berbobot merupakan graf yang pada masing-masing sisinya diberikan sebuah nilai atau bobot [2]. Bobot dari setiap sisi pada suatu graf dinotasikan dengan $w(e)$ dan bobot dari graf G adalah jumlah semua bobot sisinya dan dinotasikan dengan $W(G)$. Masing-masing sisi graf G diberi label $(e_i, w(e_i))$, menyatakan sisi e_i dengan bobot $w(e_i)$.

Contoh 1 Diberikan graf berbobot G dengan: $V(G) = \{v_1, v_2, v_3, v_4, v_5\}$ dan $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6\}$.



Gambar 3. Graf berbobot G dengan $W(G) = 21$

Lebih lanjut, jika $G = (V(G), E(G))$ adalah graf terhubung tak berarah yang mempunyai bobot pada setiap sisinya, maka bobot pohon perentang T dari G dapat diartikan sebagai jumlah bobot semua sisi pada T. Dari semua pohon perentang pada G, dapat ditemukan pohon perentang yang mempunyai bobot minimum, selanjutnya disebut pohon perentang minimum [1]. Berdasarkan uraian tersebut, langkah-langkah yang dilakukan untuk mencari pohon perentang ke-k dari suatu sebuah graf G adalah sebagai berikut:

1. Mencari pohon perentang minimum T_1 menggunakan algoritma Prim. Langkah-langkah algoritma Prim adalah sebagai berikut: diberikan graf G adalah graf berbobot yang terdiri dari n simpul dan graf $T_1 = ((V(T_1), E(T_1))$ adalah pohon perentang minimum yang dicari.
   Langkah 1. Mula-mula $V(T_1) = \emptyset$ dan $E(T_1) = \emptyset$.
   Langkah 2. Pilih sebarang simpul $v \in V(G)$. Tambahkan simpul v ke dalam $V(T_1)$.
   Langkah 3. Untuk $m < n - 1$, lakukan:
      i. Pilih sisi $e \in E(G)$ dengan syarat sisi e bersisian dengan satu simpul pada $V(T_1)$ dan e merupakan sisi dengan bobot terkecil diantara semua sisi yang bersisian dengan simpul-simpul pada $V(T_1)$
      ii. Tambahkan sisi e ke dalam $E(T_1)$ dan simpul v ke dalam $V(T_1)$ dengan syarat penambahan sisi e pada $E(T_1)$ tidak membentuk siklus.
2. Mencari pohon perentang minimum ke-k dari G, T_k $k = 2, \ldots, |T_G|$, menggunakan algoritma k-*MST*. Secara umum, langkah-langkah pada algoritma k-*MST* adalah sebagai berikut: dimisalkan graf G adalah graf awal dengan n simpul, T_k adalah pohon perentang minimum ke-k yang akan dicari dan T_G adalah himpunan pohon perentang minimum yang diperoleh sebelumnya.
   Langkah 1. Mula-mula diberikan pohon perentang minimum T_1 dan bobot total $W(T_1)$ menggunakan algoritma Prim, sehingga $T_G = \{T_1\}$.
   Langkah 2. Menyusun pohon perentang minimum berikutnya melalui proses yang disebut pergantian sisi minimum dengan ketentuan:
      i. Pilih sisi $e_i \in E(G) - E(T_1)$, sedemikian sehingga penambahan sisi e_i akan membentuk sebuah siklus C.
      ii. Hapus sisi $e_i{}' \in E(C)$ dengan bobot terbesar, dimana $w(e_i') \leq w(e_i)$.
      iii. Didapat $e_i(T_1)$ yaitu pohon perentang yang diperoleh dari penambahan sisi e_i ke T_1.
   Langkah 3. Ulangi langkah 2(i), (ii) dan (iii) untuk semua $e_i \in E(G) - E(T_1)$.
   Langkah 4. Urutkan pohon-pohon perentang yang diperoleh dari Langkah 2 berdasarkan bobot terkecil, sedemikian sehingga $T_G = \{T_1, T_2, \ldots, T_j\}$.
   Dimisalkan S adalah himpunan sisi yang anggotanya adalah s_i. Untuk menentukan pohon perentang minimum ke $j + 1$ sampai dengan k, ulangi langkah 2 dengan mengganti T_1 dengan T_2 dan seterusnya dengan syarat pohon tersebut memuat setiap sisi $s_i \in S$.

3. HASIL DAN PEMBAHASAN

Algoritma k-*MST* adalah langkah lanjutan dari algoritma dasar pencarian pohon perentang minimum yang digunakan untuk mendapatkan pohon perentang minimum ke-k. Pada bagian ini, dibahas mengenai bagaimana menentukan *MST* ke-k dengan algoritma k-*MST* dan penerapan algoritma tersebut pada jaringan pipa air bersih.

3.1 Penentuan Pohon Perentang Minimum dengan Algoritma Prim dan *k-MST*.

Tinjau kembali graf G pada Contoh 1 dan Gambar 3. Diketahui G merupakan graf sederhana dengan 5 simpul dan 6 sisi. Pertama-tama akan dicari pohon perentang minimum dengan menggunakan algoritma Prim.

Dimisalkan $T_1 = ((V(T_1), E(T_1))$ adalah pohon perentang minimum yang akan dicari.

Langkah 1. Mula-mula $V(T_1) = \emptyset$ dan $E(T_1) = \emptyset$.

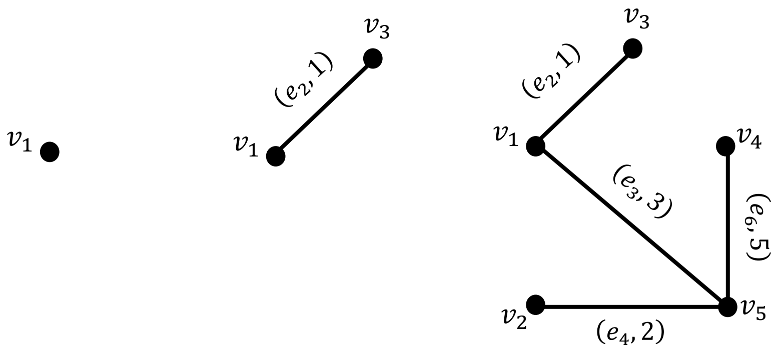
Langkah 2. Pilih simpul v_1 pada graf G sehingga $v_1 \in V(T_1)$ dan seperti yang diperlihatkan pada Gambar 4 (*a*).

Langkah 3. Pilihlah sisi $e \in E(G)$ yang bersisian dengan simpul v_1 yang mempunyai bobot terkecil. Terdapat tiga sisi yang bersisian dengan simpul v_1 yaitu e_1, e_2 dan e_3 dengan bobot berturut-turut adalah 4, 1 dan 3.

i. Pilih sisi $e_2 = (v_1, v_3)$ dengan bobot 1, sedemikian sehingga $e_2 \in E(T_1)$ dan $v_3 \in V(T_1)$, terlihat pada Gambar 4 (*b*).

ii. Tambahkan sisi dan simpul satu persatu pada $E(T_1)$ dan $V(T_1)$ selama penambahan sisi tersebut tidak membentuk siklus sampai dengan $|E(T_1)| = 5 - 1 = 4$.

Didapat pohon perentang minimum T_1 dari graf G dengan $E(T_1) = \{e_2, e_3, e_4, e_6\}$ dan $V(T_1) = \{v_1, v_3, v_5, v_2, v_4\}$ dengan bobot total sebesar 11, seperti yang terlihat pada Gambar 4 (*c*).



(*a*) (*b*) (*c*)

Gambar 4. Langkah-langkah mencari pohon perentang minimum T_1 dengan Algoritma Prim

Selanjutnya akan dicari pohon perentang minimum ke-2 sampai ke-k menggunakan algoritma $k - MST$. Dengan menggunakan Teorema Pohon Matriks (*The Matrix Tree Theorem*) [8], diperoleh bahwa $|T_G| = 11$. Dengan akan dicari pohon perentang minimum sampai $k = 11$ dengan algoritma $k - MST$.

Diberikan graf G pada Gambar 3 dan diketahui pohon pembangkit minimum T_1 dari G yang direpresentasikan pada Gambar 4(c), $T_1 = ((V(T_1), E(T_1))$ dengan $E(T_1) = \{e_2, e_3, e_4, e_6\}$ dan $V(T_1) = \{v_1, v_3, v_5, v_2, v_4\}$ dan $W(T_1) = 11$. Akan dicari $T_2, T_3, \dots, T_{11}$.

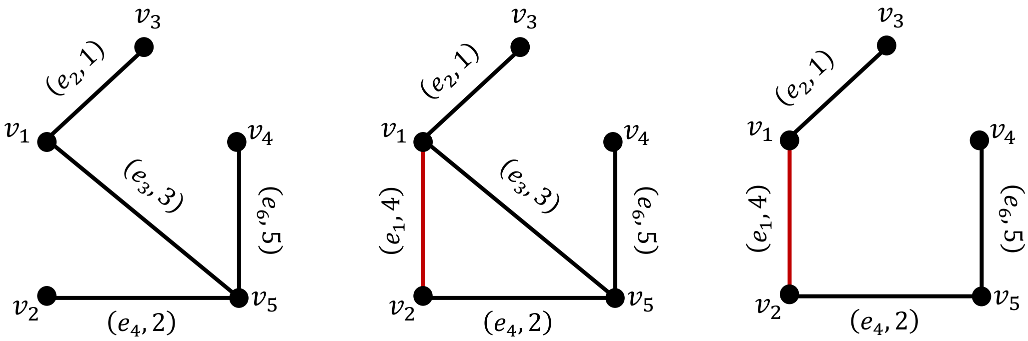
Langkah 1. Mula-mula $T_G = \{T_1\}$.

Langkah 2. Menyusun pohon perentang minimum berikutnya dengan menambahkan satu persatu sisi $e_i \in E(G) - E(T_1)$ ke T_1. Diketahui $E(G) - E(T_1) = \{e_1, e_5\}$

i. Pilih sisi e_1, tambahkan ke T_1 sehingga membentuk siklus v_1, v_2, v_5, v_1 seperti pada Gambar 5 (*b*).

ii. Hapus sisi e_3 dengan $w(e_3) \leq w(e_1)$,

iii. Diperoleh pohon perentang baru $e_1(T_1)$ dengan $W(e_1(T_1)) = 12$, seperti pada Gambar



5 (*c*).

(*a*) (*b*) (*c*)

Gambar 5. Ilustrasi Langkah 2

Equator: JMSS, Vol.2, No.1, Mei 2023: 1 - 9

Langkah 3.

i. Pilih sisi e_5 dan tambahkan pada T_1 sehingga membentuk siklus v_1, v_3, v_4, v_5, v_1.

ii. Hapus sisi e_6 dengan $w(e_6) \leq w(e_5)$ sehingga diperoleh pohon perentang baru $e_2(T_1)$ dengan $W(e_2(T_1)) = 12$.

Langkah 4. Bentuk barisan pohon perentang yang diperoleh berdasarkan bobot terkecil. Karena $W(e_1(T_1)) = W(e_2(T_1))$, maka urutkan berdasarkan bobot terkecil pada sisinya yaitu sisi e_1 dan e_5. Diperoleh $W(e_1(T_1)) \leq W(e_5(T_1))$, sehingga $e_1(T_1) = T_2$ dan $e_5(T_1) = T_3$. Kemudian masukkan ke daftar pohon T_G. Didapat $T_G = \{T_1, T_2, T_3\}$.

Pohon pembangkit berikutnya ditentukan berdasarkan pohon T_2 yang direpresentasikan pada Gambar 5(c), dengan $E(T_2) = \{e_1, e_4, e_4, e_6\}$ dan $V(T_2) = \{v_1, v_3, v_5, v_2, v_4\}$ dan $W(T_2) = 12$. Akan dicari $T_4, T_5, \dots, T_{11}$.

Langkah 1. Mula-mula $T_G = \{T_1, T_2, T_3\}$.

Langkah 2. Menyusun pohon perentang minimum berikutnya dengan menambahkan satu persatu sisi $e_i \in E(G) - E(T_2)$ ke T_2. Diketahui $E(G) - E(T_2) = \{e_3, e_5\}$

i. Pilih sisi e_3 dan tambahkan pada T_2 sehingga membentuk siklus v_1, v_2, v_5.

ii. Hapus sisi e_4 dengan $w(e_4) \leq w(e_3)$ sehingga diperoleh pohon perentang baru $e_3(T_2)$ dengan $W(e_3(T_2)) = 13$.

Langkah 3.

i. Pilih sisi e_5 dan tambahkan pada T_2 sehingga membentuk siklus v_1, v_3, v_4, v_5, v_2.

ii. Hapus sisi e_6 dengan $w(e_6) \leq w(e_5)$ sehingga diperoleh pohon perentang baru $e_5(T_2)$ dengan $W(e_5(T_2)) = 13$.

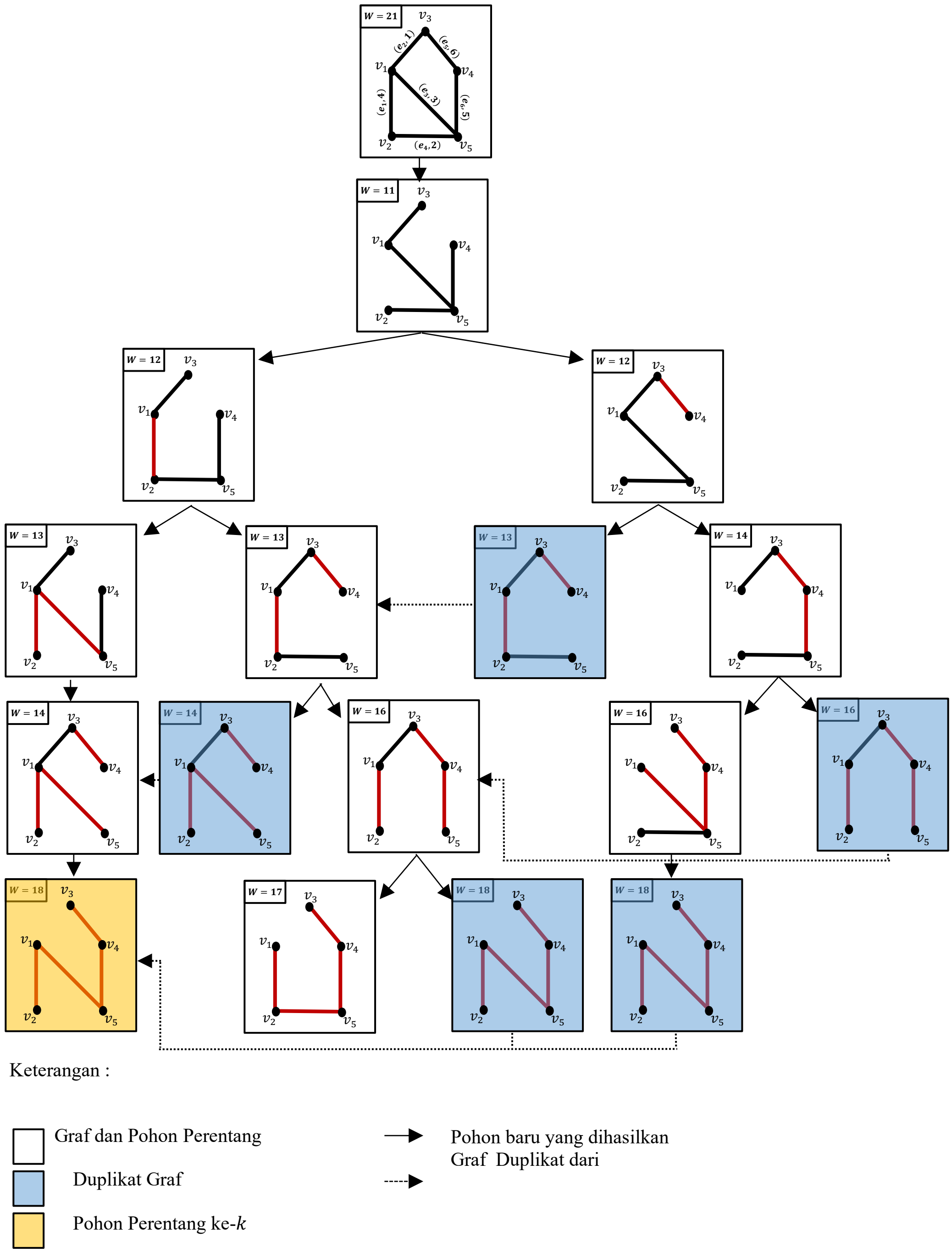
Langkah 4. Bentuk barisan pohon perentang yang diperoleh berdasarkan bobot terkecil. Karena $W(e_3(T_2)) = W(e_5(T_2))$, maka urutkan berdasarkan bobot terkecil pada sisinya yaitu sisi e_3 dan e_5. Diperoleh $W(e_3(T_2)) \leq W(e_5(T_2))$, sehingga $e_3(T_2) = T_4$ dan $e_5(T_2) = T_5$. Kemudian masukkan ke daftar pohon T_G. Didapat $T_G = \{T_1, T_2, T_3, T_4, T_5\}$.

Dengan cara yang sama, diperoleh pohon pembangkit selanjutnya berdasarkan pohon T_3, yaitu pohon perentang $e_6(T_3)$ dengan $W(e_6(T_3)) = 14$. Diperoleh $W(e_3(T_2)) \leq W(e_5(T_2)) < W(e_6(T_3))$, Lebih lanjut, daftar pohon perentang minimum secara lengkap diperlihatkan pada Tabel 1 dan Gambar 6.

Tabel 1 Daftar Pohon Perentang dari graf G

T_j	S	$e_i{}'$	$W(T_j)$
T_1	-	-	11
$T_2 = e_1(T_1)$	-	e_3	12
$T_3 = e_5(T_1)$	-	e_6	12
$T_4 = e_3(T_2)$	$\{e_1\}$	e_4	13
$T_5 = e_5(T_2)$	$\{e_1\}$	e_4	13
$T_6 = e_6(T_3)$	$\{e_5\}$	e_3	14
$T_7 = e_5(T_4)$	$\{e_1, e_3\}$	e_6	14
$T_8 = e_6(T_5)$	$\{e_1, e_5\}$	e_4	16
$T_9 = e_3(T_6)$	$\{e_5, e_6\}$	e_2	16
$T_{10} = e_2(T_8)$	$\{e_1, e_3, e_6\}$	e_2	17
$T_{11} = e_1(T_7)$	$\{e_1, e_3, e_5\}$	e_2	18

Dari proses iterasi berdasarkan algoritma $k - MST$ dalam untuk mencari 11 pohon perentang pada graf G diperoleh pohon perentang minimum ke-11 dengan bobot total 18, $W(T_{11}) = 18$.

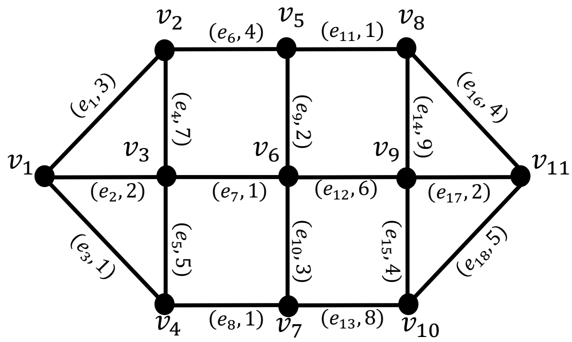


Gambar 6. Daftar Pohon Perentang T pada Graf G

3.2 Penerapan Algoritma k-MST

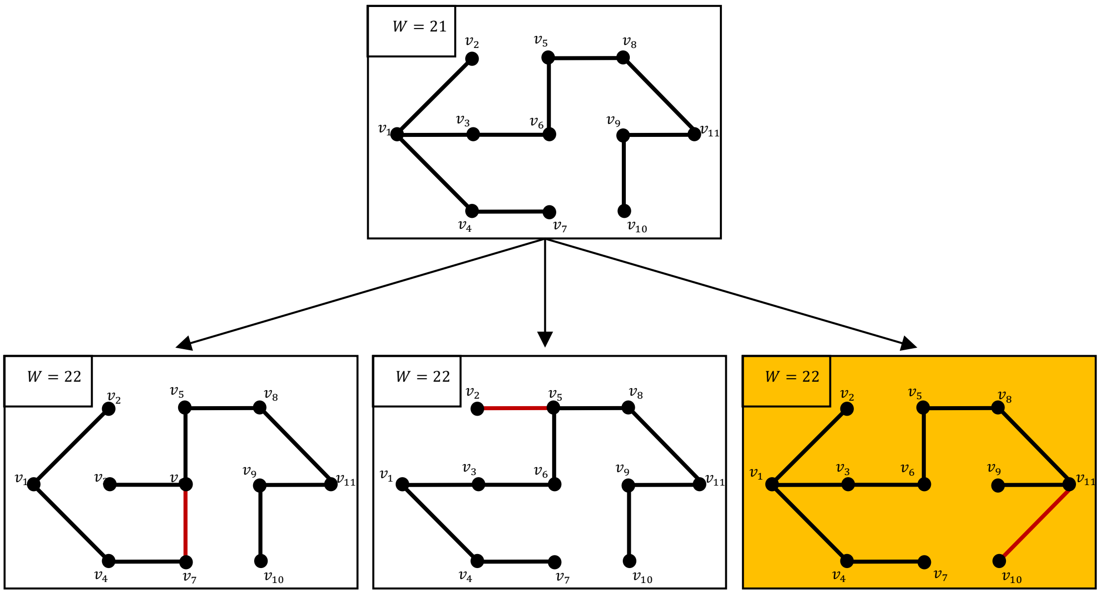
Salah satu contoh permasalahan pohon perentang minimum yang dapat diselesaikan dengan menggunakan algoritma k-MST adalah permasalahan jaringan pipa yang mendistribusikan air bersih dari suatu lokasi ke lokasi yang lain. Permasalahan yang dihadapi yaitu bagaimana meminimumkan panjang jaringan pipa distribusi air bersih sehingga dapat terhubung ke setiap lokasi.

Misalkan terdapat 11 lokasi yang akan dihubungkan oleh jalur pendistribusian air bersih yaitu $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}$ dan v_{11}, dimana v_1 menunjukkan lokasi awal yang digunakan untuk mendistribusikan air bersih ke setiap lokasi. Selanjutnya dilakukan perancangan jalur pipa sehingga membentuk sebuah graf terhubung tak berarah dan berbobot dengan simpul-simpulnya menunjukkan lokasi, sisi-sisinya menunjukkan jalur pipa yang akan menghubungkan setiap lokasi dan bobotnya merupakan panjang pipa yang diperlukan. Graf rancangan jaringan pipa distribusi air bersih dapat dilihat pada Gambar 7.



Gambar 7. Graf Jaringan Pipa Distribusi Air Bersih

Misalkan terdapat kendala pada salah satu sisinya tidak dapat dilewati, yaitu sisi e_{15}, sehingga akan dicari pohon perentang minimum berikutnya yang tidak melalui sisi e_{15}, Dengan pengerjaan yang sama dengan Contoh 10 dalam menentukan pohon perentang minimum ke-k dengan menggunakan algoritma k-MST. Diperoleh pohon perentang minimum ke-4 sudah memenuhi kendala setelah melalui pergantian sisi sebanyak 1 kali dengan bobot totalnya yaitu 22. Dapat dilihat pada Gambar 8.



Gambar 8. Daftar Pohon Perentang Minimum pada Graf Jaringan Pipa Distribusi Air Bersih

4. PENUTUP

Berdasarkan hasil dan pembahasan dapat ditarik kesimpulan bahwa algoritma k-MST dapat digunakan dalam menentukan pohon perentang minimum ke-k pada graf berbobot, dalam hal ini menggunakan algoritma Prim dalam mendapatkan pohon perentang minimum pertama. Pada artikel ini diperoleh bahwa pada graf berbobot dari Gambar 1, menghasilkan pohon perentang minimum pertama dengan bobot totalnya yaitu 11 dan pohon perentang minimum ke-11 setelah melalui pergantian sisi sebanyak 4 kali dengan bobot totalnya yaitu 18. Sedangkan pada graf jaringan pipa distribusi air bersih dari Gambar 8 dengan kendala pada sisi e_{15}, diperoleh pohon perentang minimum pertama dengan bobot totalnya yaitu 21 dan pohon perentang minimum ke-4 setelah melalui pergantian sisi sebanyak 1 kali dengan bobot totalnya yaitu 22.

DAFTAR PUSTAKA

[1] Wamiliana, *Minimum Spanning Tree dan Desain Jaringan*. Bandarlampung: Pusaka Media, 2022.

[2] R. Munir, *Matematika diskrit*, 3 ed. Informatika Bandung, 2010.

[3] Marsudi, *Teori Graf*, 1 ed. Malang: UB Press, 2016.

[4] J. J. Siang, *Riset operasi dalam pendekatan algoritmis*. Yogyakarta: ANDI, 2017.

[5] A. Asmiati, “Graf dan Aplikasinya pada Jarak Terpendek,” 2016.

[6] K. H. Rosen, *Discrete mathematics and its applications*. The McGraw Hill Companies, 2007.

[7] P. Amal dan A. K. KS, “An algorithm for kth minimum spanning tree,” *Electronic Notes in Discrete Mathematics*, vol. 53, hlm. 343–354, 2016.

[8] E. D. Ocansey, “The Matrix-Tree Theorem,” Master Thesis, African Institute for Mathematical Sciences (AIMS), South Africa, 2011.