

## EFISIENSI PENGGUNAAN MATRIKS *IN-DEGREE* UNTUK MENGKONTRUKSI *SPANNING-TREE* PADA GRAF BERARAH

Tito Nurdiyanto<sup>1)</sup>, Ely Susanti<sup>2)</sup>

<sup>1)2)</sup>Program Studi Pendidikan Matematika, FKIP Universitas Sriwijaya, Palembang, Indonesia

[titonurdiyanto\\_tnk@yahoo.co.id](mailto:titonurdiyanto_tnk@yahoo.co.id)<sup>1)</sup>, [ely\\_pasca@yahoo.com](mailto:ely_pasca@yahoo.com)<sup>2)</sup>

### ABSTRACT

Graph has a concept of tree. Tree is a connected graph which does not consist cycle. Concept of tree is an important concept because it can be used to support application of graph in any variation of science. Beside, spanning-tree is a tree in graph— $G$  with every point on graph— $G$ . Every graph can be constructed at least one spanning tree. In the construction process of spanning-tree, the use of matrix in-degree based on the value of the  $k_{qq}$  cofactor of  $K(G)$  is to determine the sum of arborescence in  $G$  rooted at the vertex  $v_q$ . Arborescence in  $G$  is spanning arborescence. Finally, the use of matrix in-degree is more efficient to construct spanning-tree in simple directed graph arborescence.

**Keywords :** *spanning-tree, directed graph, matrix in-degree*

### ABSTRAK

Graf memiliki konsep *tree* (pohon). *Tree* merupakan graf terhubung yang tidak memuat siklus. Konsep *tree* merupakan konsep yang penting karena konsep ini dapat digunakan untuk mendukung penerapan graf dalam berbagai bidang ilmu. Sedangkan *spanning-tree* adalah sebuah pohon pada graf  $G$  yang memuat semua titik di  $G$ . Dari setiap graf dapat dikonstruksi paling sedikit sebuah *spanning-tree*. Pada proses konstruksi tersebut, dengan penggunaan matriks *in-degree* berdasarkan nilai kofaktor  $k_{qq}$  dari  $K(G)$  untuk menentukan banyaknya *arborescence* pada  $G$  dengan titik  $v_q$  sebagai *root*. *Arborescence* pada  $G$  juga merupakan *spanning arborescence*. Pada akhirnya dapat disimpulkan bahwa penggunaan matrik *in-degree* lebih efisien dalam mengkonstruksi *spanning-tree* pada graf berarah sederhana yang *arborescence*.

**Kata kunci:** *spanning-tree, graf berarah, matriks in-degree*

### PENDAHULUAN

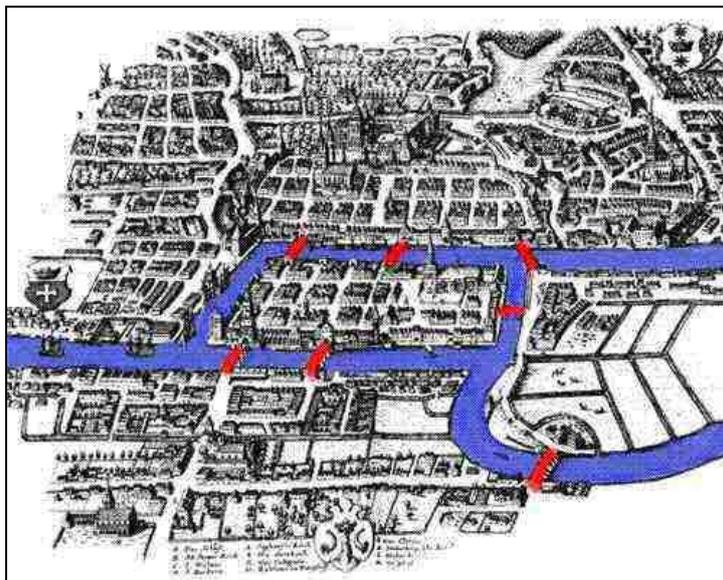
Graf digunakan pertama kali pada tahun 1736. Permasalahan yang dimodelkan dengan menggunakan graf

pada saat itu adalah masalah jembatan Königsberg. Kota Königsberg terletak di sebelah timur Prussia (sekarang bernama Jerman) yang memiliki sungai

Pregel. Sungai ini membagi kota tersebut menjadi empat daratan yang mengalir pulau Kneiphof dan bercabang menjadi dua buah anak sungai. Sekarang, kota tersebut dinamakan Kota Kaliningrad dan menjadi pusat kemajuan bidang ekonomi dan industri utama di Rusia Barat.

Pada abad ke-18, untuk menghubungkan daratan yang dibelah oleh sungai Pregel, dibangun tujuh jembatan secara bersamaan. Permasalahan yang muncul setelah pembangunan itu adalah mungkinkah

seseorang berjalan mengelilingi kota tersebut dengan melintasi ketujuh jembatan masing-masing tepat satu kali dalam perjalanan dengan syarat dimulai dan diakhiri pada tempat yang sama. Seorang ahli matematika dari Swiss, Leonhard Euler, orang yang pertama kali berhasil menemukan solusi untuk memecahkan permasalahan tersebut dengan pembuktian sederhana. Dari penemuan tersebut didapatkan cabang ilmu matematika yang sekarang dikenal dengan nama “Teori Graf”. Permasalahan di atas diilustrasikan pada gambar 1.



**Gambar 1. Peta Kota Königsberg Kuno**

Dalam teori graf, terdapat konsep *tree* (pohon). *Tree* diartikan sebagai graf terhubung yang tidak memuat siklus. Konsep *tree* merupakan konsep yang penting karena konsep ini dapat digunakan untuk mendukung penerapan graf dalam berbagai bidang ilmu.

Dalam konsep *tree* terdapat banyak jenis yang dapat digunakan untuk mencari solusi dari permasalahan

nyata. Salah satunya adalah pohon pembangun (*spanning-tree*). *Spanning-tree* ini telah diterapkan diberbagai bidang dalam kehidupan sehari-hari.

Penelitian Saxena & Urooj (2018), *minimum spanning-tree* pada graf berarah dapat digunakan untuk memecahkan masalah yang berkaitan *travelling salesman*. *Spanning-tree* digunakan untuk mendesain jaringan, analisis *Cluster*, analisis jaringan otak,

dan menyimpan data asam amino dari struktur protein (Lakshmi & Meyyappan, 2017; Kumar & Kumar, 2015). Penelitian Wayangkau (2015) menunjukkan bahwa pemasangan distribusi jaringan pipa air dapat diilustrasikan menggunakan graf. Penelitian Anggraeni (2015) menunjukkan bahwa pohon perentang minimum (*minimum spanning tree*) dapat dicari dengan menggunakan algoritma *sollin*. Penelitian yang telah dilakukan Arifanti (2017) menunjukkan dalam visualisasi efek perubahan fungsi lahan dapat dipermudah dengan menggunakan maksimum *spanning tree* dengan pembobot kolerasi. Seperti penelitian yang dilakukan oleh Aziza, dkk. (2014) membuktikan bahwa pohon matriks dapat digunakan untuk menentukan banyaknya pohon rentangan graf *Wheel W* dan Kipas *F*. Dalam graf, penggunaan algoritma prim untuk membangun pohon merentang minimum dapat menghemat kabel yang dilakukan sepanjang 222 km (Sam & Yuliani, 2016; Hayu, Yuliani, & Sam (2017). Penjelasan di atas menunjukkan banyaknya penerapan graf yang berkaitan dengan *spanning-tree* dalam kehidupan sehari-hari.

*Spanning-tree* pada graf  $G$  adalah sebuah pohon di  $G$  yang memuat semua titik di  $G$ . Pada penelitian Raharjo (2009) menunjukkan bahwa metode pertukaran sisi (*edge exchange*) digunakan untuk menentukan jumlah *spanning-tree* dari graf berarah sederhana  $G$ , serta dapat dicari juga dengan menggunakan matriks *in-degree* ( $K(G)$ ). Dari hasil dan pembahasan pada penelitian sebelumnya, peneliti

menyimpulkan bahwa tahapan yang dilakukan terlihat kurang efisien dalam mengkontruksi *spanning-tree* untuk graf berarah tersebut. Oleh sebab itu, pada penelitian ini, peneliti menekankan dan mendetailkan cara efisien untuk dapat digunakan dalam mengkontruksi *spanning-tree* pada graf berarah yang diharapkan dapat diterapkan dalam berbagai permasalahan yang telah dijelaskan.

Dalam pembahasan ini hanya terbatas pada penentuan jumlah *spanning-tree* pada graf berarah yang sederhana dan *spanning-tree* pada graf berarah adalah *arborescence*. Pembahasan yang disajikan adalah cara efisien untuk dapat digunakan dalam mengkontruksi *spanning-tree* untuk graf berarah sederhana yang *arborescence*.

## TINJAUAN PUSTAKA

Definisi-definisi pada tinjauan pustaka di bawah ini dirujuk dari beberapa referensi, diantaranya Anton (2004), Daswa & Riyadi (2017), Deo (1997), Hadley (1961), Munir (2003), Munir (2012), Raharjo (2009), Rahmawati (2016), dan Siang (2002).

### 1. Graf

Graf  $G$  didefinisikan : pasangan himpunan  $(V, E)$  dengan notasi  $G = (V, E)$ . Dalam hal ini,  $V$  adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) dan  $E$  adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul.

Berdasarkan ada tidaknya gelang atau sisi ganda, graf dibedakan menjadi dua jenis, yaitu graf sederhana (*simple graph*) dan graf tak-sederhana

(*unsimple graph*). Graf yang tidak mengandung gelang maupun sisi-ganda dinamakan graf sederhana. Sedangkan graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana.

Berdasarkan orientasi arah pada sisi, graf dibedakan menjadi dua jenis orientasi arah yaitu graf berarah (*directed graph*) dan graf tak berarah (*undirected graph*). Graf yang sisinya tidak mempunyai orientasi arah dinamakan graf tak berarah. Urutan pasangan titik pada graf tak berarah tidak diperhatikan, jadi sisi  $(u, v)$  sama dengan  $(v, u)$ . Graf tak berarah  $G$  sering disebut dengan graf  $G$  saja. Sedangkan graf yang setiap sisinya merupakan pasangan terurut dari dua titik dinamakan graf berarah. Pada graf berarah, sisi  $(u, v)$  tidak sama dengan  $(v, u)$ . Untuk sisi  $(u, v)$ , titik  $u$  merupakan titik awal dan titik  $v$  merupakan titik akhir.

Pada graf berarah, derajat simpul  $v$  dinyatakan dengan  $d_{in}(v)$  dan  $d_{out}(v)$  yang dalam hal ini:  $d_{in}(v) =$  derajat masuk (*in-degree*) = jumlah busur yang masuk ke simpul  $v$  dan  $d_{out}(v) =$  derajat masuk (*out-degree*)

= jumlah busur yang keluar ke simpul  $v$  dan  $d(v) = d_{in}(v) + d_{out}(v)$ , sehingga berlaku:  $\sum_{v \in V} d_{in}(v) = \sum_{v \in V} d_{out}(v) = |E|$ .

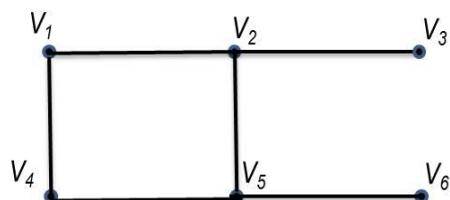
## 2. Tree (Pohon)

Graf terhubung yang tidak memuat sebuah siklus dinamakan *Tree* (pohon).

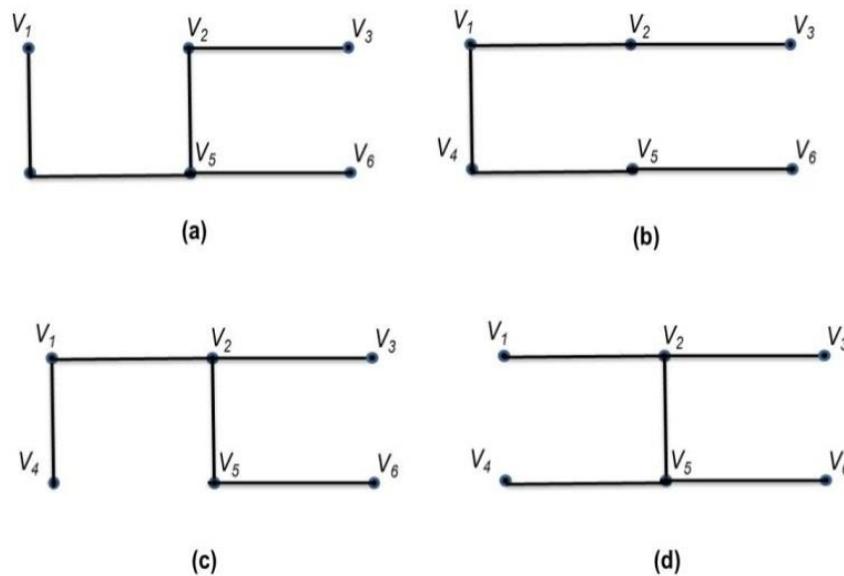
### Teorema 1

Misalkan  $G$  adalah suatu graf dengan  $n$  buah titik dan tepat  $n - 1$  sisi. Bila  $G$  tidak memuat siklus, maka  $G$  adalah pohon. Misal  $G$  sebuah graf. Sebuah pohon di  $G$  yang memuat semua titik  $G$  disebut pohon pembangun (*spanning-tree*) dari  $G$ .

Pada *spanning-tree* terdapat *branch* dan *chord*. *Branch* adalah sebuah sisi yang terdapat dalam sebuah *spanning-tree*. Sedangkan *chord* adalah sebuah sisi yang tidak terdapat dalam sebuah *spanningtree*, tetapi berada dalam graf  $G$ . Contoh graf tak berarah ditunjukkan pada gambar 2, sedangkan *spanning-tree* dari graf  $G$  di atas ditunjukkan pada gambar 3 (a)-(d) berikut.



Gambar 2. Graf Tak Berarah



Gambar 3 (a)-(d) *Spanning-tree* dari graf G

Sebuah *tree* (pohon) pada jenis graf berarah yang tidak memuat suatu siklus berarah disebut pohon berarah. Sedangkan sebuah titik pada pohon berarah yang memiliki derajat masuk 0 disebut *root*.

### 3. Matriks

Matriks didefinisikan sebagai susunan skalar elemen-elemen dalam bentuk baris dan kolom. Matrika  $A$  yang berukuran dari  $m$  baris dan  $n$  kolom ( $m \times n$ ) adalah:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Jika  $A$  dan  $B$  adalah sembarang dua matriks yang memiliki ordo yang sama, maka  $A + B$  matriks yang diperoleh dengan menjumlahkan entri-entri yang bersesuaian dari kedua matriks tersebut. Matriks yang memiliki ordo yang berbeda tidak dapat dijumlahkan.

Jika  $A$  adalah matriks  $m \times r$  dan  $B$  adalah matriks  $r \times n$ , maka hasil kali

$AB$  didefinisikan sebagai matriks  $C$  dengan ordo  $m \times n$  yang entri-entrinya dihitung dari elemen-elemen dari  $A$  dan  $B$  menurut:  $c_{ij} = \sum_{k=1}^r a_{ik}b_{kj}$  dengan  $i = 1, 2, \dots, m; j = 1, 2, \dots, n$

Misalkan  $A$  merupakan suatu matriks kuadrat. Fungsi determinan dinyatakan *det*, dan semua hasil kali elementer bertanda dari  $A$  didefinisikan dengan  $det(A)$ . Jumlah  $det(A)$  dinamakan determinan  $A$ .  $det(A) = |A| = \sum \pm a_{1j_1} a_{2j_2} \dots a_{nj_n}$  dimana  $\Sigma$  menunjukkan bahwa suku-suku tersebut harus dijumlahkan terhadap semua permutasi  $(j_1, j_2, \dots, j_n)$  dan simbol (+) atau (-) dapat dipilih dalam masing-masing suku sesuai dengan apakah permutasi itu genap atau ganjil. Sebuah permutasi dinamakan genap jika jumlah invers seluruhnya adalah sebuah bilangan bulat yang genap (simbol (+)) dan dinamakan ganjil jika jumlah invers seluruhnya adalah sebuah bilangan bulat ganjil (simbol (-)).

**Teorema 2**

Jika  $A$  merupakan matriks segitiga bawah ataupun segitiga atas maka  $\det(A)$  berupa hasil kali entri-entri pada diagonal utama, yaitu:  $\det(A) = a_{11}a_{22} \dots a_{nn}$ .

**Teorema 3**

Misalkan  $A, B$  dan  $C$  adalah matriks-matriks  $n \times n$  yang berbeda hanya satu kolom, misalnya kolom ke- $r$  dari  $C$  dapat diperoleh dengan menjumlahkan entri-entri yang bersesuaian pada kolom ke- $r$  dari  $A$  dan  $B$ . Maka :  $\det(C) = \det(A) + \det(B)$ . Hasil yang sama berlaku untuk baris.

Misalkan matriks  $A = [a_{ij}]$  berordo  $n \times n$ , dan  $A_{ij}$  suatu submatriks dari  $A$  yang berordo  $(n - 1) \times (n - 1)$  di mana baris ke- $i$  dan kolom ke- $j$  dihilangkan. Kofaktor  $a_{ij}$  adalah  $(-1)^{i+j} M_{ij}$ .

**4. Cara Merepresentasikan Graf Berarah pada Matriks**

Terdapat dua cara merepresentasikan graf berarah dengan menggunakan matriks, yaitu sebagai berikut.

$$a_{ij} = \begin{cases} 0, & \text{jika sisi } j \text{ tidak terkait dengan titik } i \\ 1, & \text{jika sisi } j \text{ terkait dengan titik } i \text{ dan arah sisinya keluar dari } i \\ 2, & \text{jika sisi } j \text{ terkait dengan titik } i \text{ dan arah sisinya masuk dari } i \end{cases}$$

Apabila diadakan penghapusan terhadap baris yang bersesuaian dengan sembarang titik pada matriks *exhaustive incidence*, maka terbentuk submatriks  $(n - 1) \times m$ . Titik yang bersesuaian dengan baris yang dihapus tersebut disebut titik acuan (*verteks reference*), sedangkan matriks berukuran  $(n - 1) \times m$  yang terjadi ini disebut matriks keterkaitan (*matriks incidence*) dengan notasi  $A$ .

a) Penyajian Graf dengan Matriks Ketetangaan (*Adjacency Matriks*)

Misal  $G$  adalah sebuah graf berarah dengan  $n$  buah titik. Matriks  $X$  dengan ordo  $(n \times n)$  merupakan matriks ketetangaan dari  $G$  yang didefinisikan:  $X = [x_{ij}]$ , dengan  $x_{ij}$  banyaknya sisi berarah yang menghubungkan dari titik  $v$  menuju titik  $v_j$ .

Bila  $G$  tidak mengandung sisi rangkap. Maka entri-entri pada matriks  $X$  adalah 0 dan 1. Jika graf berarah  $G$  mengandung sisi rangkap, entri-entri pada matriks  $X$  merupakan bilangan bulat non negatif.

b) Penyajian Graf dengan Notasi Matriks Keterkaitan (*Incidence Matriks*)

Matriks *exhaustive incidence*  $Ae = [x_{ij}]$  untuk semua graf yang menyajikan hubungan antara sebuah titik dengan sisi yang menghubungkannya. Untuk graf terhubung berarah elemen matriks *exhaustive incidence*  $Ae = [x_{ij}]$  di mana;

**METODE PENELITIAN**

Metode penelitian yang digunakan pada penelitian ini adalah studi pustaka. Tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut. (1) menemukan permasalahan, (2) merumuskan permasalahan, (3) menggunakan analisis pemecahan masalah, dan (4) menuliskan

kesimpulan dan representasi permasalahan.

**HASIL DAN PEMBAHASAN**

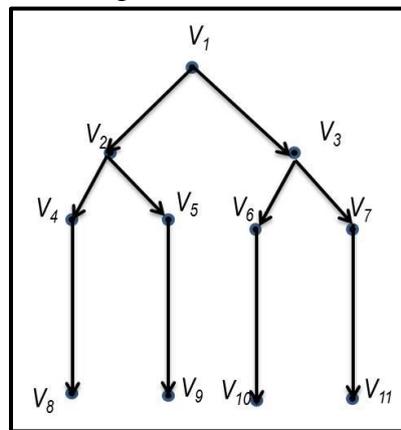
Penelitian ini membahas cara efisien untuk dapat digunakan dalam mengkontruksi *spanning-tree* dari graf berarah sederhana, yaitu *spanning-arborescence*.

Misal  $G = (V, A)$  adalah graf berarah dan  $r \in V$  adalah sebuah titik yang disebut *root*. *Arborescence* dengan

*root*  $r$  adalah sebuah subgraf  $T = (V, F)$  dari  $G$  yang tidak memuat sepasang sisi berlawanan sedemikian hingga kondisi berikut terpenuhi.

- a. Jika arah dari tiap sisi diabaikan, maka  $T$  adalah sebuah *spanning-tree*.
- b. Terdapat sebuah lintasan dari  $r$  ke setiap titik yang lain  $v \in V$ .

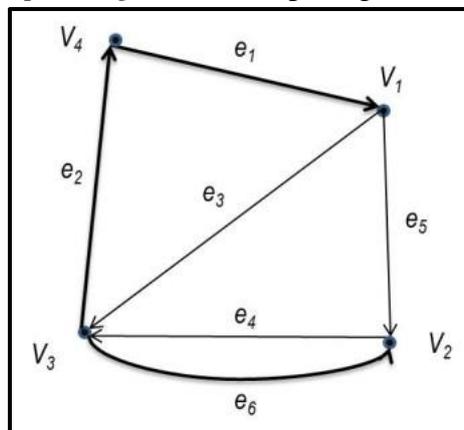
Graf berarah dengan *root*  $v_1$  ditunjukkan pada gambar 4 berikut.



**Gambar 4. Graf Berarah dengan Root  $V_1$**

*Spanning-tree* pada graf terhubung berarah dengan  $n$  buah titik, analog dengan *spanning-tree* pada graf tak berarah yaitu terdiri dari  $n - 1$  buah sisi. *Spanning arborescence* pada graf terhubung berarah adalah *spanning-tree*

yang *arborescence*. Sehingga *spanning-arborescence* merupakan *arborescence*. Subgraf yang dicetak tebal pada graf  $G$  merupakan contoh dari *spanning arborescence*. Digraf  $G$  ditunjukkan pada gambar 5 berikut.



**Gambar 5. Digraf  $G$**

Berikut teorema-teorema yang mengkaji bagaimana menentukan jumlah *spanning-tree* pada graf berarah.

**Teorema 4**

Pada *arborescence* terdapat sebuah lintasan berarah dari *root*  $R$  menuju setiap titik yang lain. Sebaliknya jika  $G$  graf berarah sederhana tanpa siklus,  $G$  disebut *arborescence* jika terdapat sebuah titik  $v$  di  $G$  sedemikian hingga dapat di buat lintasan berarah dari  $v$  ke setiap titik yang lain, maka  $G$  *arborescence*.

Bukti:

- a. Andaikan tidak ada lintasan berarah dari *root*  $R$  ke sebuah titik  $v_i$ , artinya titik  $v_i$  berderajat masuk 0 ( $v_i$  *root*). Hal ini kontradiksi dengan  $G$  *arborescence*. Jadi terdapat lintasan berarah dari *root*  $R$  ke semua titik yang lain di  $G$ .
- b. Jelas  $G$  tidak memuat siklus maka  $G$  adalah pohon. Sehingga tinggal dibuktikan  $G$  memuat tepat satu

$$K(G) = \begin{cases} d_{in}(v_i), & \text{jika } i = j \\ -x_{ij}, & \text{jika } i \neq j \text{ entri dari matriks ketetanggaan, (tanda negatif)} \end{cases}$$

Apabila diadakan penghapusan terhadap sebuah baris dan kolom yang bersesuaian dengan *root*  $v_i$ , maka akan diperoleh submatrik  $K_{ii}$  dengan mengeluarkan baris- $i$  dan kolom- $i$  dari matriks  $K(G)$ .

**Teorema 5**

$G$  graf berarah sederhana dengan  $n$  buah titik dan  $n - 1$  sisi berarah adalah *arborescence* dengan  $v_1$  sebagai *root*, jika dan hanya jika kofaktor  $k$  dari  $K(G)$  adalah 1.

Bukti:

- a. Akan dibuktikan syarat perlu ( $\rightarrow$ )

*root*. Karena terdapat sebuah lintasan berarah dari  $v$  ke titik yang lain, maka derajat masuk  $v$  sama dengan 0, jadi  $v$  *root* dari  $G$ . Andaikan terdapat *root* yang lain di  $G$  sebut  $u$  maka yang terjadi, tidak bisa dibuat lintasan dari  $u$  ke  $v$ . Sehingga kontradiksi, jadi haruslah  $G$  memuat sebuah *root*.

Pada graf berarah diketahui adanya *in-degree* (derajat masuk) dan *out-degree* (derajat keluar), bergantung pada orientasi arah yang diberikan pada tiap sisinya. Akan tetapi pada permasalahan ini, lebih menitik beratkan pada derajat masuk pada graf berarah.

Suatu matriks *in-degree*,  $K(G)$  dari sebuah graf terhubung berarah  $G = (V, E)$  tanpa *loop* dengan  $V = \{v_1, v_2, \dots, v_n\}$  dan  $E = \{e_1, e_2, \dots, e_n\}$  adalah sebuah matriks dengan ukuran  $n \times n$  yang mempunyai sifat:

Misal  $G$  adalah *arborescence* dengan  $n$  buah titik dan  $v_1$  sebagai *root*. Beri label titik-titik di  $G$  dengan  $v_1, v_2, v_3, \dots, v_n$  sehingga titik-titik yang berada di sepanjang lintasan berarah yang dimulai dari  $v_i$ . Karena  $v_1$  berderajat masuk sama dengan nol maka entri-entri pada kolom pertama sama dengan nol. Sedangkan entri yang lain dalam  $K(G)$  adalah sebagai berikut.

$$\begin{aligned} k_{ij} &= 0, i > j \\ k_{ij} &= -x_{ij}, i < j \\ k_{ii} &= 1, i > 1 \end{aligned}$$

Andaikan  $k_{ii} > 1$ , artinya: terdapat paling sedikit 2 sisi berarah yang menuju ke  $v_i$ . Misalkan sisi tersebut berasal dari titik  $v_{i-1}$  dan  $v_{i-2}$ . Sedangkan dari  $v_1$  diketahui terdapat

$$K(G) = \begin{bmatrix} 0 & -x_{12} & -x_{13} & \vdots & -x_{1n} \\ 0 & 1 & -x_{23} & \vdots & -x_{2n} \\ 0 & 0 & 1 & \vdots & -x_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix}$$

Dengan menghapus baris pertama dan kolom pertama dari  $K(G)$  diperoleh submatriks  $K_{11}$  yang merupakan matriks segitiga atas. Berdasarkan teorema 2 diperoleh  $\det K_{11} = 1$  sehingga kofaktor  $k_{11}$  adalah 1.

b. Akan dibuktikan syarat cukup ( $\leftarrow$ )

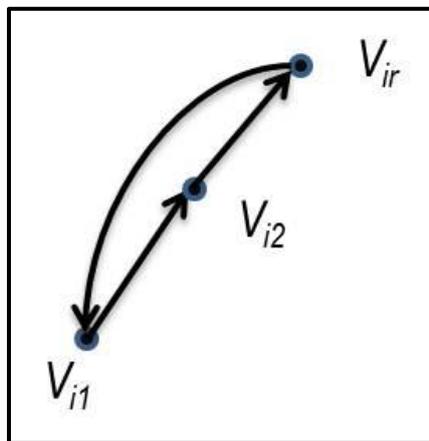
Misal  $G$  adalah graf berarah sederhana dengan  $n$  buah titik dan  $n - 1$  sisi dengan kofaktor  $k_{11}$  dari matriks  $K(G)$  adalah 1, maka  $\det K_{11} = 1$ . Karena  $\det K_{11} \neq 0$ , maka setiap kolom dalam  $K_{11}$  paling sedikit memuat sebuah entri bukan nol. Oleh karena itu,  $d_{ii}(v_i) \geq 1$ , untuk  $i = 2, 3, \dots, n$ .

Tidak mungkin terjadi  $d_{in}(v_i) > 1$ . Andaikan  $k_{ii} > 1$ ,

lintasan berarah dari  $v_1$  ke  $v_{i-1}$  dan  $v_1$  ke  $v_{i-2}$ . Sehingga kontradiksi dengan  $G$  *arborescence*. Maka  $K$  matriks dari *arborescence* dengan *root*  $v_1$  berlaku:

artinya: terdapat paling sedikit 2 sisi berarah yang menuju ke  $v_i$ . Misalkan sisi tersebut berasal dari titik  $v_{i-1}$  dan  $v_{i-2}$ . Sedangkan dari  $v_1$  diketahui terdapat lintasan berarah dari  $v_1$  ke  $v_{i-1}$  dan  $v_1$  ke  $v_{i-2}$ . Sehingga  $G$  memuat siklus dari  $v_1$  ke  $v_i$ . Hal ini kontradiksi dengan  $G$  hanya memuat  $n - 1$  sisi berarah. Sehingga :  $d_{in}(v_i) = 1$  untuk  $i = 2, 3, \dots, n$  dan  $d_{in}(v_1) = 0$ .

Akan dibuktikan  $G$  tidak memuat siklus. Andaikan terdapat siklus di  $G$ , yang melalui titik  $v_{i_1}, v_{i_2}, \dots, v_{i_r}$ . Maka jumlah kolom  $i_1, i_2, \dots, i_r$  dalam  $K_{11}$  adalah nol. Hal ini ditunjukkan pada gambar 6 graf  $G$  berikut.



Gambar 6. Graf  $G$

Menjadi :

$$K(G) = \begin{matrix} & v_{1r} & v_{2r} & \dots & v_{ir} \\ \begin{matrix} v_{1r} \\ v_{2r} \\ \vdots \\ v_{ir} \end{matrix} & \begin{bmatrix} 1 & -1 & \dots & 0 \\ 0 & 1 & \dots & -1 \\ \vdots & \vdots & \dots & \vdots \\ -1 & 0 & \dots & 1 \end{bmatrix} \end{matrix}$$

Sehingga  $r$  kolom pada  $K_{11}$  adalah bergantung linier, Akibatnya  $\det K_{11} = 0$ . Hal ini sebuah kontradiksi dengan  $\det K_{11} = 1$ , sehingga  $G$  tidak memuat siklus.

Karena  $G$  memuat  $n - 1$  sisi dan tidak memuat siklus maka  $G$  adalah pohon. Karena  $d_{in}(v_i) = 0$  dan  $d_{in}(v_i) = 1$  untuk  $i = 2, 3, \dots, n$ , maka  $G$  haruslah *arborescence* dengan root  $v_1$ . **(terbukti)**

**Teorema 6**

Misal  $K(G)$  adalah matriks *in-degree* dari graf berarah sederhana. Maka nilai dari kofaktor  $k_{qq}$  dari  $K(G)$  adalah sama dengan banyaknya *arborescence* pada  $G$  dengan titik  $v_q$  sebagai *root*.

Bukti :

Misal  $K(G)$  adalah matriks *in-degree* dari graf berarah sederhana yang terdiri dari  $n$  buah vektor kolom untuk setiap matriks berordo  $n$ . Sehingga  $K(G) = [p_1, p_2, \dots, p_r, \dots, p_n]$

Pilih vektor kolom  $p_r$  dari  $K(G)$  yang memuat entri  $k_{11} > 1$ , sehingga  $p_r$  dapat dibentuk menjadi  $(p_i + p_{i'})$  dengan  $p_i$  dan  $p_{i'}$  adalah vektor kolom yang masing-masing memuat entri  $k_{ii} = 1$ . Sehingga  $K(G) = [p_1, p_2, \dots, (p_i + p_{i'}), \dots, p_n]$ .

Berdasarkan teorema 3, maka diperoleh:

$$\det K(G) = \det [p_1, p_2, \dots, p_i, \dots, p_n] + \det [p_1, p_2, \dots, p_{i'}, \dots, p_n].$$

Setiap vektor kolom  $p_j$  yang memuat entri  $k_{ii} > 1$  dibentuk menjadi  $(p_j + p_{j'})$  dengan  $p_j$  dan  $p_{j'}$  adalah vektor kolom yang memuat entri  $k_{ii} = 1$ . Sedangkan vektor kolom  $p_q$ ,  $q \neq j$  digunakan untuk membentuk kofaktor  $K_{qq}$ . Masing-masing kofaktor bersesuaian dengan sebuah subgraf  $g$  dari  $G$  yang memenuhi :

- a. Setiap titik di  $g$  punya derajat masuk tepat satu, kecuali  $v_n$ .
- b.  $g$  memuat  $n - 1$  titik dan  $n - 1$  sisi.

$$\text{Jadi } \det K_{qq}(G) = \sum_g \det K_{qq}(g)$$

Dari teorema 5, diperoleh:  $\det K_{qq} = 1$ , jika dan hanya jika  $g$  *arborescence* berakar di  $q = 0$  yang lain.

Jadi nilai dari kofaktor  $k_{qq}$  dari  $K(G)$  adalah sama dengan banyaknya *arborescence* pada  $G$  dengan titik  $v_q$  sebagai *root*. **(terbukti)**.

Dari kajian pustaka dan pembuktian di atas, penggunaan matriks *in-degree* dapat menjadi lebih efisien dalam mengkontruksi *spanning-tree* pada graf berarah. Tahapan yang harus dilakukan adalah sebagai berikut.

1. Buatlah graf  $G$  sebagai graf berarah sederhana.
2. Representasikan graf  $G$  ke dalam bentuk matriks *in-degree* (sebut matriks  $K(G)$ ).
3. Pilih sebuah titik  $v_q$  sebagai *root* dari graf  $G$  tersebut.
4. Lakukan penghapusan sebuah baris ke- $q$  dan kolom ke- $q$  dari matriks  $K(G)$ , sehingga diperoleh determinan submatriks  $K(G)$ .
5. Hitunglah kofaktor dari submatriks  $dK(G)$ .
6. Hitunglah banyaknya *arborescence* pada graf  $G$  dengan titik  $v_q$  sebagai

root dari jumlah *spanning-tree* yaitu dari jumlah determinan dari masing-masing matriks kofaktor  $K_{qq}$  dari bentuk setiap *arborescence* yang dihasilkan.

- Representasikan masing-masing matriks dari *arborescence* yang terbentuk dalam graf sebagai *spanning-tree* yang dikonstruksi..

**Contoh Permasalahan :**

Diberikan graf  $G = (V, E)$  adalah graf berarah sederhana dengan  $V = \{1,2,3,4\}$  dan  $E = \{(1,2), (1,3), (2,3), (2,4), (3,4)\}$  dengan matriks *in-degree*  $K(G)$ . Tentukan banyaknya *spanning-tree*

*arborescence* pada graf  $G$  yang dapat dikonstruksi!

Jawab :

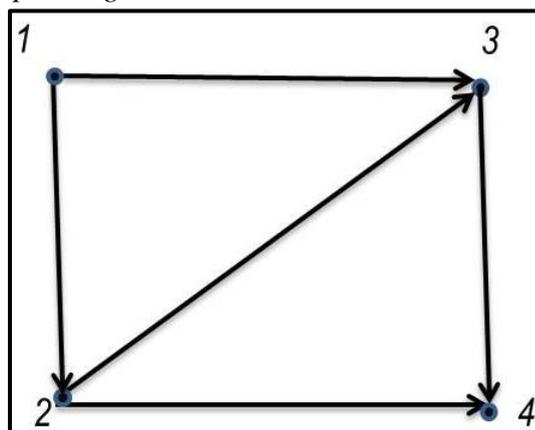
Diketahui :

Graf  $G = (V, E)$  graf berarah sederhana  
 $V = \{1,2,3,4\}$   
 $E = \{(1,2), (1,3), (2,3), (2,4), (3,4)\}$   
 dengan matriks *in-degree*  $K(G)$ .

Ditanya : Banyaknya *spanning-tree arborescence* pada graf  $G$  yang dapat dikonstruksi?

Penyelesaian :

- Membuat graf berarah sederhana dari yang diketahui (pada gambar 7)



Gambar. 7 Graf G

- Merepresentasikan graf  $G$  pada gambar 7 di atas ke dalam bentuk matriks *in-degree*.

$$K(G) = \begin{matrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & \begin{bmatrix} 0 & -1 & -1 & 0 \end{bmatrix} \\ v_2 & \begin{bmatrix} 0 & 1 & -1 & -1 \end{bmatrix} \\ v_3 & \begin{bmatrix} 0 & 0 & 2 & -1 \end{bmatrix} \\ v_4 & \begin{bmatrix} 0 & 0 & 0 & 2 \end{bmatrix} \end{matrix}$$

- Memilih titik 1 sebagai *root*.
- Melakukan penghapusan terhadap sebuah baris ke-1 dan kolom ke-1 yang bersesuaian dengan *root*  $v_1$ , maka akan di peroleh  $M_{11}$  dari

matriks  $K(G)$ , sehingga kita dapatkan determinan submatrik adalah

$$M_{11} = \begin{bmatrix} 1 & -1 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

$$M_{11} = ((1.2.2) + ((-1).0.0)) - (((-1).2.0) + ((-1).0.2))$$

$$M_{11} = 4 - 0$$

$$M_{11} = 4$$

5. Menhitung kofaktor dari  $K_{11}(G)$  adalah

$$C_{11} = (-1)^{1+1} \cdot 1 \cdot M_{11} = 1 \cdot 4 = 4$$

Berdasarkan teorema 6, banyaknya  $spanning-$

$$K(G) = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix} \end{matrix}$$

$$\det K_{11}(G) = \sum_g \det K_{11}(g)$$

$$\det \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix} \end{matrix} = \det \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix} \end{matrix} + \det \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix} \end{matrix}$$

$$\det \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix} \end{matrix} = \det \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} + \det \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

$$+ \det \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} + \det \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

$$\det K_{11}(G) = 1 + 1 + 1 + 1$$

$$\det K_{11}(G) = 4.$$

Banyaknya  $spanning-tree arborescence$  pada graf  $G$  yang dapat dikonstruksi adalah 4.

7. Setelah itu, merepresentasikan masing-masing matriks menjadi  $arborescence$  yang terbentuk dalam graf.

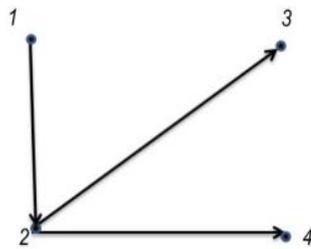
a) Matriks pertama

$tree arborescence$  pada graf  $G$  yang dapat dikonstruksi adalah 4.

6. Banyaknya  $spanning-tree arborescence$  pada graf  $G$  yang dapat dikonstruksi adalah :

$$\begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

Matriks di atas direpresentasikan pada gambar 8.

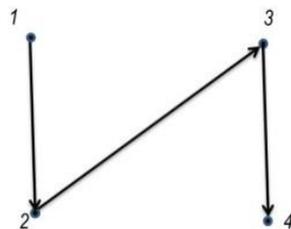


**Gambar 8. Graf dari Matrik Pertama**

b) Matriks kedua

$$\begin{array}{c}
 1 \quad 2 \quad 3 \quad 4 \\
 1 \quad \begin{bmatrix} 0 & -1 & 0 & 0 \end{bmatrix} \\
 2 \quad \begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix} \\
 3 \quad \begin{bmatrix} 0 & 0 & 1 & -1 \end{bmatrix} \\
 4 \quad \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}
 \end{array}$$

Matriks di atas direpresentasikan pada gambar 9.



**Gambar 9 Graf dari Matrik Kedua**

c) Matriks ketiga

$$\begin{array}{c}
 1 \quad 2 \quad 3 \quad 4 \\
 1 \quad \begin{bmatrix} 0 & -1 & -1 & 0 \end{bmatrix} \\
 2 \quad \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \\
 3 \quad \begin{bmatrix} 0 & 0 & 1 & -1 \end{bmatrix} \\
 4 \quad \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}
 \end{array}$$

Matriks di atas direpresentasikan pada gambar 10.

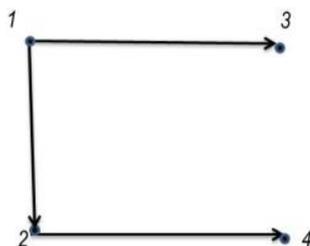


**Gambar 10 Graf dari Matrik Ketiga**

d) Matriks keempat

$$\begin{array}{c}
 1 \quad 2 \quad 3 \quad 4 \\
 1 \quad \begin{bmatrix} 0 & -1 & -1 & 0 \end{bmatrix} \\
 2 \quad \begin{bmatrix} 0 & 1 & 0 & -1 \end{bmatrix} \\
 3 \quad \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \\
 4 \quad \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}
 \end{array}$$

Matriks di atas direpresentasikan pada gambar 11.



**Gambar 11. Graf dari Matrik Keempat**

Penggunaan matriks *in-degree* di atas digunakan mengkontruksi *spanning-tree* pada graf berarah. *Spanning-tree* tersebut dapat digunakan untuk mendesain jaring seperti jaringan komputer, TV kabel, jaringan telekomunikasi, jaringan transportasi, dan jaringan listrik.

## KESIMPULAN DAN SARAN

### Kesimpulan

Kesimpulan dari pembahasan di atas adalah penggunaan matrik *in-degree* menjadi lebih efisien dalam mengkontruksi *spanning-tree* pada graf berarah sederhana yang *arborescence*. Banyaknya *spanning-tree* pada graf berarah sederhana yang dikonstruksi dapat dicari dengan menggunakan matriks *in-degree* ( $K(G)$ ) dengan teorema: banyaknya *arborescence* pada graf  $G$  dengan *root*  $v_q$  sama dengan kofaktor  $k_{qq}$  dari  $K(G)$ .

### Saran

Pada pembahasan ini, hanya terbatas pada graf berarah yang *arborescence*. Oleh sebab itu, saran yang diberikan penulis untuk penelitian selanjutnya adalah dapat mengkontruksi *spanning-tree* pada graf berarah yang bukan *arborescence*.

## DAFTAR PUSTAKA

- Anggraeni, W. (2015). Aplikasi Algoritma Sollin dalam Pencarian Pohon Perentang Minimum Provinsi Jawa Tengah. *Faktor Exacta*, 8(4), 381-391.
- Anton, H. (2004). *Aljabar Linier Elementer*. Jakarta: Erlangga.
- Arifanti, D. R. (2017). Visualisasi Efek Perubahan Fungsi Lahan Menggunakan Maksimum *Spanning Tree* dengan Pembobot Kolerasi. *Al-Khwarizmi: Jurnal Pendidikan Matematika dan Ilmu Pengetahuan Alam*, 5(2), 155-164.
- Aziza, F., Suyitno, A., & Mulyono. (2014). Teorema Pohon Matriks untuk Menentukan Banyaknya Pohon Rentang Graf Wheels  $W$  dan Kipas  $F$ . *UNNES Journal of Mathematics*, 3(2), 104-109
- Daswa & Riyadi, M. (2017). Aplikasi Pewarnaan graf pada Masalah Penyusunan Jadwal Perkuliahan di Universitas Kuningan. *JES-MAT*, 3(2), 217-226.
- Deo, N. (1997). *Graph Theory With Application to Engineering and Computer Science*. Prentice-Hall of India.
- Hadley, G. (1961). *Linier Algebra*. Adison-Wesley PublishingCo.

- Hayu, W., Yuliani, & Sam, M., (2017). Pembentukan Pohon Merentang Minimum dengan Algoritma Kruskal. *Jurnal Scientific Pinisi*, 392), 108-115.
- Kumar, A., & Kumar, G. (2015). An Efficient Methode to Construct Minimum Spanning Tree. *International Journal of Latest Tecnology Egeinering, Management & Application Science*, 5(10), 48-51.
- Lakshmi, K., & Meyyappan, T. (2016). Spanning Tree-Properties, Algorithms, and Applications. *International Journal of Computer Sciences and Engineering*, 5(10).
- Munir, R. (2003). *Matematika Diskrit*. Bandung: Informatika.
- Munir, R. (2012). *Matematika Diskrit*. Bandung: Informatika.
- Raharjo, R.S. (2009). *Penentuan Jumlah Spanning-tree pada Graf Berarah dengan menggunakan Metode Pertukaran Sisi dan Matriks In-Degree*. Diakses tanggal 26 Januari 2016 di [www.lib.unnes.ac.id/3106/1/4788.pdf](http://www.lib.unnes.ac.id/3106/1/4788.pdf).
- Rahmawati, N. D. (2016). Teorema Pohon Matriks untuk Menentukan Banyaknya Pohon Rentangan Graf bipartisi Komplit ( $K_{m,n}$ ). *Jurnal Kajian Keismlaman dan Kemasyarakatan*, 1(1), 87-100.
- Rubrik Matematika Majalah 1000guru. 2013 *Teka-teki Jembatan Konigsberg*. Diakses tanggal 1 Maret 2016 di <http://majalah1000guru.net/2013/08/jembatan-konigsberg/>
- Sam, M. & Yuliani. (2016). Penerapan Algoritma Prim untuk Membangun Pohon Merentang Minimum (Minimum Spanning Tree) dalam Pengoptimalan Jaringan Transmisi Nasional Prvinsi Sulsel. *Jurnal Dinamika*, 7(1), 50-61.
- Saxena, S., & Urooj. (2018). New Approaches for Minimum Spanning Tree. *International Journal of Advance in Science and Engineering*, 7(3), 353-361.
- Siang, J. (2002). *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: Andi.
- Wayangkau, I. H. (2015). Optimalisasi Particle Swarm pada Pemasangan Jaringan Pipa Air PDAM. *Jurnal Ilmiah Mustek Anim*, 4(1).
- Weistein, Eric W. *Minimum Cost Arborescences*. Diakses tanggal 26 Januari 2016 di <http://mathworld.worlfram.com/Arborescence.pdf>