MAKALAH TEORI GRAF

PENOMORAN *VERTEX* PADA TOPOLOGI JARINGAN GRAF *WHEEL* DAN GRAF *HELM*

Anggota Kelompok 6:

1. Abdul Jalil Nawawi (M0118001)
2. Noni Kurnia Dewi (M0118047)
3. Nur Fadila Marsaoly (M0119063)
4. Tiyas Astutiningsih (M0119088)
5. Tri Hartati (M0119089)

PROGRAM STUDI MATEMATIKA

UNIVERSITAS SEBELAS MARET SURAKARTA

2021

vhj,m

BAB I
PENDAHULUAN
1.1. Latar Belakang Masalah

Matematika merupakan ilmu dasar yang banyak diterapkan atau dimanfaatkan dalam kehidupan sehari-hari. Berbagai masalah dalam kehidupan dapat diselesaikan dengan menggunakan model matematika. Salah satu cabang ilmu yang dapat digunakan untuk memodelkan masalah dalam kehidupan sehari-hari yaitu Teori Graf. Sebagai contoh, teori graf bisa diterapkan dalam bidang teknologi jaringan.

Jaringan yang efisien dapat diperoleh melalui penomoran *vertex* pada suatu graf yang merepresentasikan topologi jaringan. Dari graf tersebut, kemudian diselesaikan dengan mencari rute terpendek sehingga dapat meminimumkan biaya dalam penyusunan topologi jaringan tersebut. Pada makalah ini akan dibahas mengenai pemberian nomor *vertex* pada topologi jaringan graf *wheel* dan graf *helm* untuk mendapatkan bentuk topologi jaringan graf yang efisien.

1.2. Perumusan Masalah

Berdasarkan pada latar belakang diperoleh tiga rumusan masalah, yaitu

1. apa yang dimaksud dengan topologi jaringan,
2. bagaimana memberikan nomor *vertex* pada topologi jaringan graf *wheel* dan graf *helm* secara sistematis dan efisien, dan
3. bagaimana penerapan penomoran *vertex* pada topologi jaringan graf *wheel* dan graf *helm*.

1.3. Tujuan

Berdasarkan rumusan masalah diperoleh tujuan dari penulisan makalah ini, yaitu

1. mampu mengetahui maksud dari topologi jaringan,
2. mampu memberikan nomor *vertex* pada topologi jaringan graf *wheel* dan graf *helm* secara sistematis dan efisien, dan
3. mampu mengetahui penerapan penomoran *vertex* pada topologi jaringan graf *wheel* dan graf *helm*.

BAB II

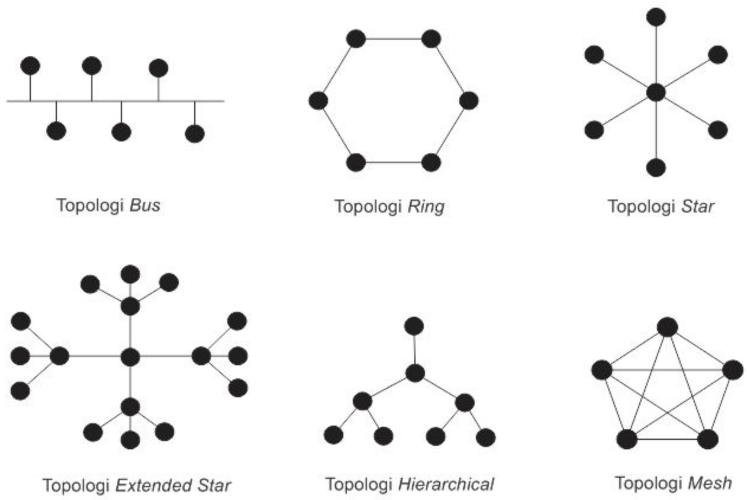
PEMBAHASAN

2.1 Pengertian Topologi Jaringan

Pengertian dasar topologi jaringan dan jenisnya menurut Behrouz dan Mosharraf [1] dijelaskan sebagai berikut.

Definisi 2.1.1. *Topologi jaringan dalam telekomunikasi adalah suatu cara menghubungkan perangkat telekomunikasi yang satu dengan yang lainnya sehingga membentuk jaringan.*

Adapun topologi yang umum digunakan dalam membuat jaringan adalah topologi jaringan *bus, ring, star, extended star, erarchical,* dan *mesh* yang disajikan pada Gambar 2.1.

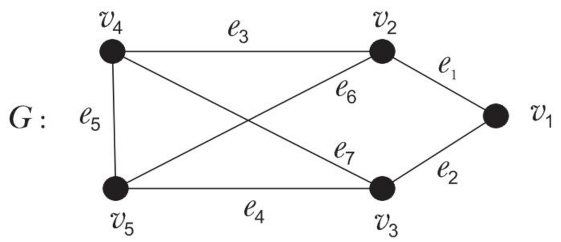


Gambar 2.1. Jenis-jenis topologi jaringan

2.2 Pengertian Dasar Graf

Pada pembahasan ini akan digunakan beberapa pengertian dasar graf, yaitu pengertian graf, *walk, cycle, bridge,* dan *minimum spanning tree* menurut Chartrand dan Lesniak [2], pengertian *adjacent* dan *incident* menurut Bondi dan Murty [3], serta pengertian graf berbobot menurut Munir [4].

Definisi 2.2.1. *Suatu graf G adalah himpunan tak kosong berhingga $V(G)$ yang disebut himpunan vertex (boleh tunggal) bersama-sama dengan himpunan* *(mungkin kosong) pasangan tidak berurutan dari vertex pada graf G yang disebut $E(G)$.*



Gambar 2.2. Graf G

Pada Gambar 2.2, graf G memiliki himpunan *vertex* $V = \{v_1, v_2, v_3, v_4, v_5\}$ dan $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$. Banyaknya *vertex* dalam suatu graf disebut *order* dan banyaknya *edge* dalam suatu graf disebut *size.*

Definisi 2.2.2. *Jika u dan v adalah sembarang dua vertex dari graf G yang dihubungkan oleh edge e, dinotasikan e=(u,v) maka dapat dikatakan bahwa u dan v adalah vertex yang adjacent. Kemudian vertex u dan v dikatakan incident dengan edge e.*

Pada Gambar 2.2, *vertex* v_1 dan v_2 merupakan *vertex* yang *adjacent.* Serta *vertex* v_1 dan v_2 *incident* dengan *edge* e_1.

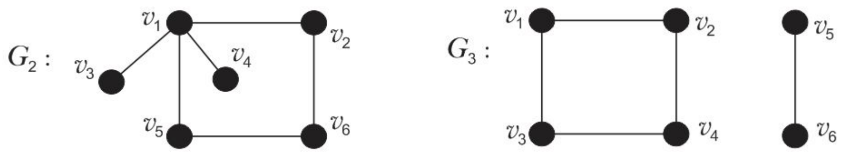
Definisi 2.2.3. *Suatu $u-v$ walk dari graf G adalah barisan bergantian antara vertex dan edge yang dimulai dari vertex u dan berakhir di vertex v. Suatu $u-v$ trail adalah $u-v$ walk yang tidak mengulang sembarang edge. Suatu $u-v$ path adalah $u-v$ walk yang tidak mengulang sembarang vertex.*

Pada Gambar 2.2, barisan $v_1, e_1, v_2, e_3, v_4, e_5, v_5, e_4, v_3, e_7, v_4, e_3, v_2, e_6, v_5, e_4, v_3, e_2, v_1$ adalah contoh *walk*, barisan $v_1, e_1, v_2, e_3, v_4, e_7, v_3$ adalah contoh *trail*, dan barisan $v_1, e_1, v_2, e_6, v_5, e_5, v_4, e_7, v_3$ adalah contoh *path.*

Definisi 2.2.4. *Suatu cycle dari graf G adalah walk $v_0, v_1, \ldots, v_n$ dengan $n \geq 3$ dimana $v_0 = v_n$ dan n vertex dari $v_0, v_1, \ldots, v_{n-1}$ yang saling berbeda.*

Pada Gambar 2.2 diperoleh $v_1, e_1, v_2, e_3, v_4, e_7, v_3, e_2, v_1$ yang merupakan salah satu *cycle* dari graf G.

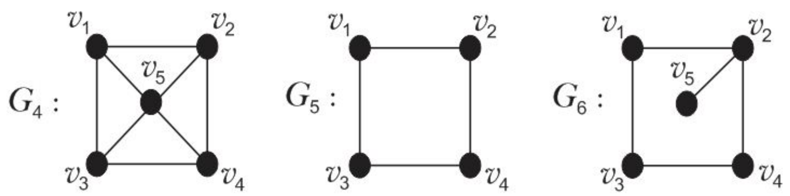
Definisi 2.2.5. *Suatu graf G dikatakan terhubung (connected) jika memuat suatu path di antara sembarang dua vertex u dan v dari G. Jika tidak demikian, maka graf G disebut tidak terhubung (disconnected).*



Gambar 2.3. Graf *connected* G_2 dan graf *disconnected* G_3

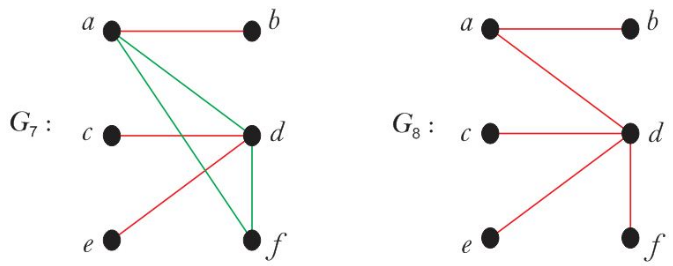
Jika diambil sembarang dua *vertex* dalam graf G_2 pada Gambar 2.3 maka terdapat suatu *path*, sedangkan pada graf G_3 dalam Gambar 2.3 tidak demikian. Gambar 2.3 graf G_2 merupakan contoh graf terhubung, sedangkan graf pada Gambar 2.3 graf G_3 merupakan contoh graf tak terhubung.

Definisi 2.2.6. *Graf P adalah spanning subgraf dari G jika $V(P) = V(G)$.*



Gambar 2.4. Graf G_5 subgraf graf G_4 dan graf G_6 *spanning* subgraf G_4

Definisi 2.2.7. *Tree adalah graf terhubung yang tidak memuat cycle (asiklik), dengan setiap edge pada tree adalah suatu bridge.*

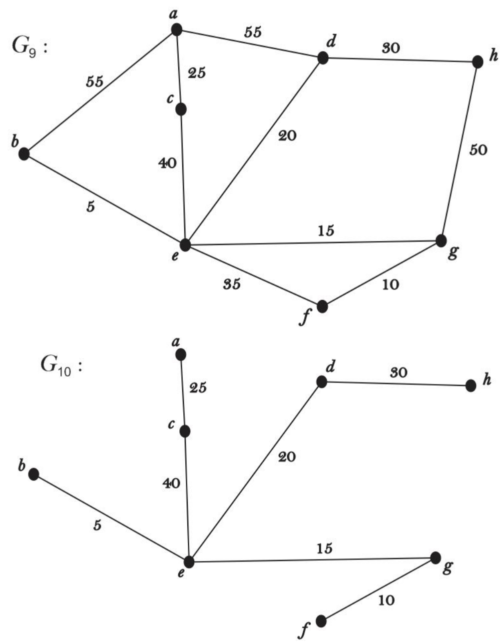


Gambar 2.5. Subgraf G_8 adalah subgraf dari G_7 yang berbentuk *tree*

Definisi 2.2.8. *Misalkan G adalah graf terhubung. Spanning tree dari G adalah spanning subgraf T dari G jika T adalah sebuah tree dan T memuat semua vertex dari G.*

Pada Gambar 2.5. subgraf G_8 adalah *spanning tree* dari G_7.

Definisi 2.2.9. *Misalkan G adalah graf terhubung berbobot. Spanning tree dengan total bobot edgenya paling kecil disebut minimum spanning tree (MST) dari G.*



Gambar 2.6. Graf G_{10} adalah MST dari G_9

2.3 *Breadth First Search* (BFS) Moore

Sebelum membahas algoritme BFS Moore, akan dijelaskan definisi jarak dua *vertex* pada suatu graf.

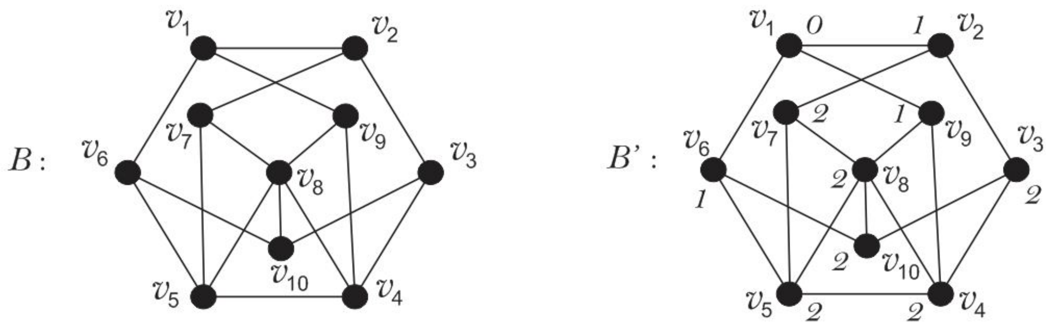
Definisi 2.3.1. *Jarak (distance) dari vertex u ke vertex v pada graf G adalah panjang path terpendek dari vertex u ke v, dinotasikan dengan $d(u, v)$. Jika tidak ada lintasan yang menghubungkan vertex u dan v maka $d(u, v) = 1$.*

Selanjutnya dalam penyelesaian masalah lintasan terpendek dalam graf G, digunakan algoritme BFS Moore menurut Chartrand *et al* [5] sebagai berikut.

1. Diambil salah satu *vertex*, misal u, dan dilabeli 0 yang menyatakan jarak dari u ke dirinya sendiri, sedangkan semua *vertex* selain u dilabeli ∞.

2. Semua *vertex* berlabel ∞ yang *adjacent* dengan *vertex* u diberi label 1.

3. Semua *vertex* berlabel ∞ yang *adjacent* dengan *vertex* berlabel 1 dilabeli dengan 2 demikian seterusnya sampai *vertex* yang dimaksud, misal v, sudah berlabel. Dalam hal ini, label dari setiap *vertex* yang menyatakan jarak dari *vertex* u.



Gambar 2.7. Graf tak berlabel B dan Graf berlabel B'

Sebagai ilustrasi, diberikan graf B pada Gambar 2.7 yang menggunakan algoritme BFS Moore untuk menentukan jaraknya ke *vertex* v_1 dari setiap *vertex* di B. Berikut adalah langkah-langkah pelabelan graf B dengan algoritme BFS Moore untuk membentuk MST B' dari graf B.

1. Memilih *vertex* v_1 sebagai *vertex* awal dan diberi label 0 serta *vertex* lainnya yang belum berlabel diberi label ∞.

2. *Vertex* yang *adjacent* dengan *vertex* v_1 yaitu *vertex* v_2, v_6, dan v_9 dilabeli dengan 1 dan *vertex* lain yang belum berlabel diberi label ∞.

3. *Vertex* berlabel ∞ yang *adjacent* dengan *vertex* v_2 yaitu *vertex* v_3 dan v_7, dilabeli dengan 2.

4. *Vertex* berlabel ∞ yang *adjacent* dengan *vertex* v_7 yaitu *vertex* v_5 dan v_8, dilabeli dengan 2.

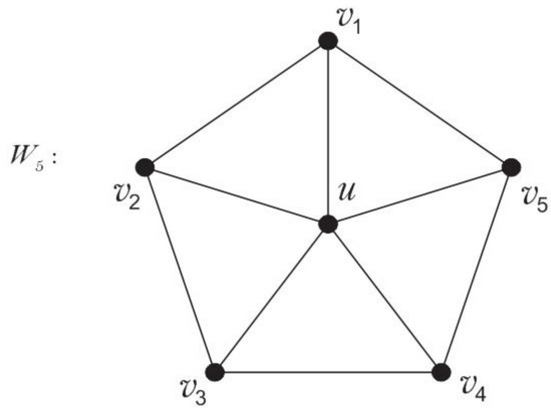
5. *Vertex* berlabel ∞ yang *adjacent* dengan *vertex* v_8 yaitu *vertex* v_4 dan v_{10}, dilabeli dengan 2. Semua *vertex* sudah berlabel maka *stop*.

2.4 Kelas-Kelas Graf

Berikut diberikan graf *wheel* menurut Gallian [6], dan graf *helm* menurut Wallis [7].

Definisi 2.4.1. *Graf wheel yang dinotasikan sebagai W_n adalah graf yang terdiri dari graf cycle dan satu tambahan vertex, dimisalkan u, yang adjacent ke semua vertex yang terdapat pada cycle.*

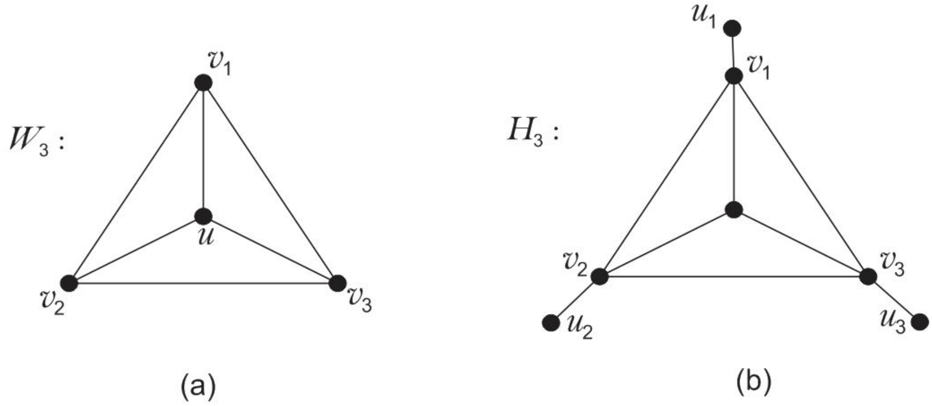
Contoh dari graf *wheel* dapat dilihat pada Gambar 2.8.



Gambar 2.8. Graf *wheel* W_5

Definisi 2.4.2. *Graf helm H_n adalah graf yang dikonstruksikan dari wheel W_n dengan menambahkan n vertex berdegree, yang adjacent pada setiap vertex terminal.*

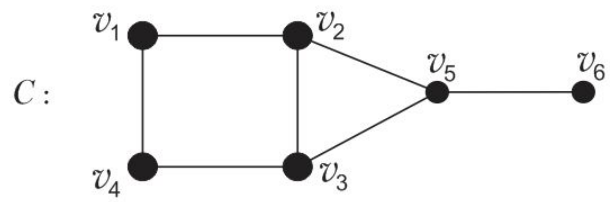
Contoh dari graf *helm* dapat dilihat pada Gambar 2.9.



Gambar 2.9. (a) Graf *wheel* W_3 (b) Graf *helm* H_3

2.5 Eksentrisitas

Definisi 2.5.1. *Eksentrisitas (eccentricity) vertex* u*, dinotasikan* $e(u)$*, dalam graf* G *adalah jarak terjauh (lintasan terpendek maksimum) dari vertex* u *ke setiap vertex di* G*.*



Gambar 2.10. Graf C

Tabel 2.1 Eksentrisitas dari graf C pada Gambar 2.10

vertex	eksentrisitas *vertex*	*vertex* eksentrik
v_1	$e(v_1) = 3$	v_6
v_2	$e(v_2) = 2$	v_4, v_6
v_3	$e(v_3) = 2$	v_1, v_6
v_4	$e(v_4) = 3$	v_6
v_5	$e(v_5) = 2$	v_1, v_4
v_6	$e(v_6) = 3$	v_1, v_4

2.5.1 Algoritme Kamalesh dan Srivatsa

Langkah-langkah pemberian nomor *vertex* pada topologi jaringan graf *wheel* dan graf *helm* dengan algoritme Kamalesh dan Srivatsa mengacu pada Kamalesh dan Srivatsa [8].

1. Membuat topologi jaringan graf *wheel* dan graf *helm*.
2. Memberikan label berupa (v_1, v_2, ..., v_n) pada *vertex* pada graf tersebut.
3. Menentukan bentuk *minimum spanning tree* pada topologi jaringan graf *wheel* dan graf *helm* dengan algoritme BFS Moore.

4. Mencari nilai eksentrisitas setiap *vertex* dari *minimum spanning tree* pada graf tersebut.

5. Memberikan nomor *vertex* pada graf topologi jaringan *wheel* dan graf *helm* berdasarkan nilai eksentrisitas, *vertex* yang memiliki nilai eksentrisitas yang lebih kecil maka diberikan penomoran *vertex* lebih kecil. Jika terdapat *vertex* dengan nilai eksentrisitas sama maka *vertex* dengan jarak dari titik awal yang lebih pendek diberikan penomoran yang lebih kecil. Jika terdapat *vertex* dengan jarak dari titik awal sama maka *vertex* dengan indeks lebih kecil diberikan penomoran *vertex* lebih kecil.

2.6 Penomoran *Vertex* Pada Topologi Jaringan Graf *Wheel* W_n dan Graf *Helm* H_n

Berikut diberikan teorema pada penomoran *Vertex* Pada Topologi Jaringan Graf *Wheel* W_n dan Graf *Helm* H_n menurut Sidiq *et al.*[9].

2.6.1 Penomoran *Vertex* Pada Topologi Jaringan Graf *Wheel* W_n

Graf *wheel* memiliki himpunan *vertex* $V(W_n) = \{u, v_1, v_2, ..., v_n\}$. Jika W'_n merupakan *minimum spanning tree* dari graf *wheel* W_n maka $V(W'_n) = V(w_n)$ dan urutan penomoran *vertex* dari *minimum spanning tree* graf *wheel* W'_n dinotasikan dengan $\lambda(u), u \in V(W'_n)$.

Teorema 2.6.1. *Jika* W'_n *merupakan bentuk minimum spanning tree dari graf wheel, dengan vertex awal* $1 \leq r \leq n$*, maka penomoran setiap vertex dalam* W'_n *mengikuti rumus sebagai berikut.*

Untuk $v_r = u$

$$\lambda(u) = 1,$$
$$\lambda(v_j) = 1 + j, \qquad j = 1, 2, ..., n.$$

Untuk $v_r = v_1$

$$\lambda(u) = 2,$$
$$\lambda(v_j) = \begin{cases} 1, & j = 1 \\ j + 1, & j = 2, 3, ..., n. \end{cases}$$

Untuk $v_r = 2 \leq r \leq n-1$

$$\lambda(u) = 2,$$
$$\lambda(v_j)=\begin{cases} 1, & j = r \\ j+2, & j = 1,2,...,r-1 \\ j+1, & j = r+1, r+2, ..., n. \end{cases}$$

Untuk $v_r = v_n$

$$\lambda(u) = 2,$$
$$\lambda(v_j)=\begin{cases} 1, & j = n \\ j+2, & j = 1,2,...,n-1. \end{cases}$$

Bukti. Untuk $v_r = u$, diberikan *minimum spanning tree* graf *Wheel* W_n' dengan pengambilan *vertex* awal u, sehingga urutan pemberian nomor *vertex* dari eksentrisitas *vertex* u, $\lambda(u)$ adalah 1, sedangkan *vertex* v_j, $\lambda(v_j)$ adalah $1+j$ untuk setiap $j = 1, 2, ..., n$.

Untuk $v_r = v_1$, diberikan *minimum spanning tree* graf *Wheel* W_n' dengan pengambilan *vertex* awal v_1, sehingga urutan pemberian nomor *vertex* dari eksentrisitas *vertex* u, $\lambda(u)$ adalah 2, dan *vertex* v_j, $\lambda(v_j)$ adalah 1, serta $\lambda(v_j)$ adalah $j+1$ untuk setiap $j = 2, 3, ..., n$.

Untuk $v_r = 2 \leq r \leq n-1$, diberikan *minimum spanning tree* graf *Wheel* W_n' dengan pengambilan *vertex* awal v_r untuk $2 \leq r \leq n-1$, sehingga urutan pemberian nomor *vertex* dari eksentrisitas *vertex* u, $\lambda(u)$ adalah 2, dan *vertex* v_j, $\lambda(v_j)$ adalah 1 untuk $j = r$, $\lambda(v_j)$ adalah $j+2$ untuk $j = 1, 2, ..., r-1$, serta $\lambda(v_j)$ adalah $j+1$ untuk $j = r+1, r+2, ..., n$.

Untuk $v_r = v_n$, diberikan *minimum spanning tree* graf *Wheel* W_n' dengan pengambilan *vertex* awal v_n, sehingga urutan pemberian nomor *vertex* dari eksentrisitas *vertex* u, $\lambda(u)$ adalah 2,*vertex* v_j, $\lambda(v_j)$ adalah $j = 1$ untuk $j = n$, serta $\lambda(v_j)$ adalah $j+2$, untuk $j = 1, 2, ..., n-1$.

2.6.2 Penomoran *Vertex* Pada Topologi Jaringan Graf *Helm* H_n

Graf *helm* memiliki himpunan *vertex* $V(H_n) = \{u, v_1, v_2, ..., v_n, w_1, w_2, ..., w_n\}$. Jika H'_n merupakan *minimum spanning tree* dari graf *helm* H_n dengan u sebagai *vertex* awal, maka $V(H'_n) = V(H_n)$ dan urutan penomoran *vertex* dari *minimum spanning tree* graf *helm* H'_n dinotasikan dengan $\lambda(u), u \in V(H'_n)$.

Teorema 2.6.2. *Jika H'_n merupakan bentuk minimum spanning tree dari graf helm H_n, maka penomoran setiap vertex dalam H'_n mengikuti rumus sebagai berikut.*

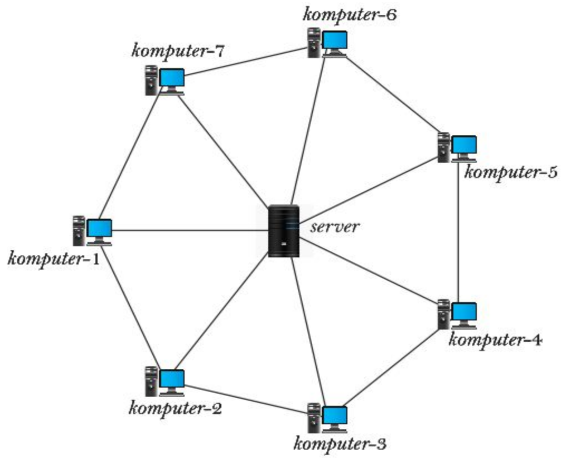
$$\begin{aligned} &\lambda(u) = 1 \\ &\lambda(v_j) = 1 + j && j = 1, 2, ..., n, \\ &\lambda(w_k) = 1 + n + k && k = 1, 2, ..., n. \end{aligned}$$

Bukti. Diberikan *minimum spanning tree* graf *helm* H'_n, sehingga urutan pemberian nomor *vertex* dari eksentrisitas *vertex* $u_i, \lambda(u)$ adalah 1 untuk setiap $i = 1$, dan *vertex* $v_j, \lambda(v_j)$ adalah $1 + j$ untuk setiap $j = 1, 2, ..., n$, serta *vertex* $w_k, \lambda(w_k)$ adalah $1 + n + k$ untuk setiap $k = 1, 2, ..., n$.

BAB III
PENERAPAN

Diberikan contoh-contoh penerapan kasus penomoran *vertex* pada topologi jaringan graf dalam kehidupan sehari-hari.

Contoh 3.1. Sebuah laboratorium komputer memiliki jaringan seperti pada Gambar 3.1. Representasikan jaringan dalam graf dan berilah label pada setiap *vertex* di graf menggunakan algoritme Kamalesh dan Srivatsa.

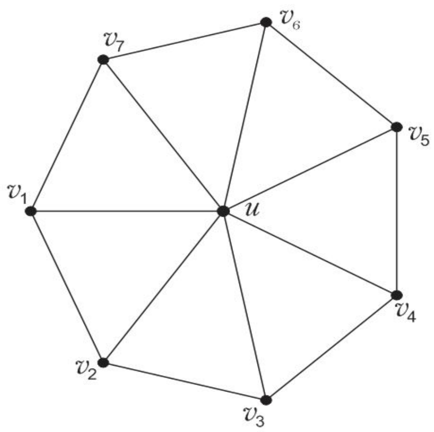


Gambar 3.1. Rangkaian Jaringan Komputer

Penyelesaian:

Akan digunakan algoritme Kamalesh dan Srivatsa sebagai berikut.

1. Merepresentasikan rangkaian jaringan komputer ke dalam graf W_7 dengan $V(W_7) = \{u, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ seperti pada Gambar 3.2.



Gambar 3.2. Graf W_7

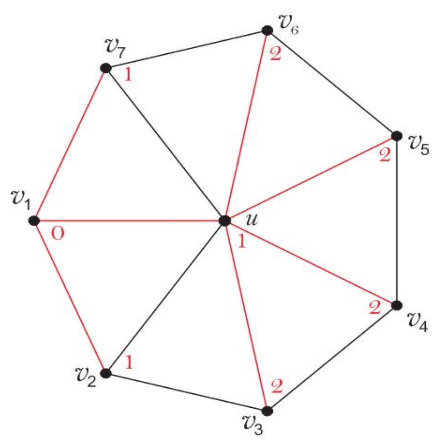
2. Menentukan *minimum spanning tree* W_7' dari graf W_7 menggunakan metode *BFS Moore* dengan langkah-langkah sebagai berikut..

    (a) Memilih *vertex* v_1 sebagai *vertex* awal dan diberi label 0 serta *vertex* lainnya yang belum berlabel diberi label ∞.

    (b) *Vertex* berlabel ∞ yang *adjacent* dengan *vertex* v_1 yaitu *vertex* u, v_2, dan v_7 dilabeli dengan 1 dan *vertex* lain yang belum berlabel diberi label ∞.

    (c) *Vertex* berlabel ∞ yang *adjacent* dengan *vertex* u yaitu *vertex* v_3, v_4, v_5, dan v_6, diberi label 2. Semua *vertex* sudah berlabel maka berhenti.

    Berikut ini diperoleh *minimum spanning tree* W_7' dari graf W_7 dengan menggunakan metode *BFS Moore* ditunjukkan pada Gambar 3.3.



Gambar 3.3. Graf berlabel W_7'

3. Mencari nilai eksentrisitas setiap *vertex* dari *minimum spanning tree* graf tersebut. Berikut nilai eksentrisitas setiap *vertex* dari *minimum spanning tree* W_7' dari graf W_7.

$$\begin{array}{ll} e(u) = 2 & e(v_4) = 3 \\ e(v_1) = 2 & e(v_5) = 3 \\ e(v_2) = 3 & e(v_6) = 3 \\ e(v_3) = 3 & e(v_7) = 3 \end{array}$$

4. Memberikan label baru sesuai aturan algoritme Kamalesh dan Srivatsa. Penomoran *vertex* dari *minimum spanning tree* W_7' dari graf W_7 adalah sebagai berikut,

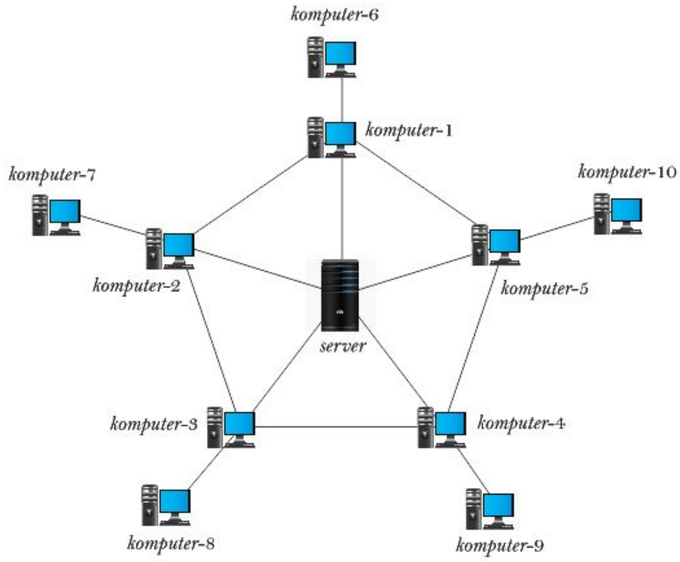
$$\begin{array}{ll} \lambda(u) = 2 & \lambda(v_4) = 5 \\ \lambda(v_1) = 1 & \lambda(v_5) = 6 \\ \lambda(v_2) = 3 & \lambda(v_6) = 7 \\ \lambda(v_3) = 4 & \lambda(v_7) = 8 \end{array}$$

Sehingga diperoleh hasil akhir penyelesaian masalah yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Jarak setiap *vertex* dalam MST graf W_7'

vertex	eksentrisitas *vertex*	nomor *vertex*
u	$e(u) = 2$	2
v_1	$e(v_1) = 2$	1
v_2	$e(v_2) = 3$	3
v_3	$e(v_3) = 3$	4
v_4	$e(v_4) = 3$	5
v_5	$e(v_5) = 3$	6
v_6	$e(v_6) = 3$	7
v_7	$e(v_7) = 3$	8

Contoh 3.2. Sebuah laboratorium komputer di sebuah universitas memiliki rangkaian jaringan seperti pada Gambar 3.4.

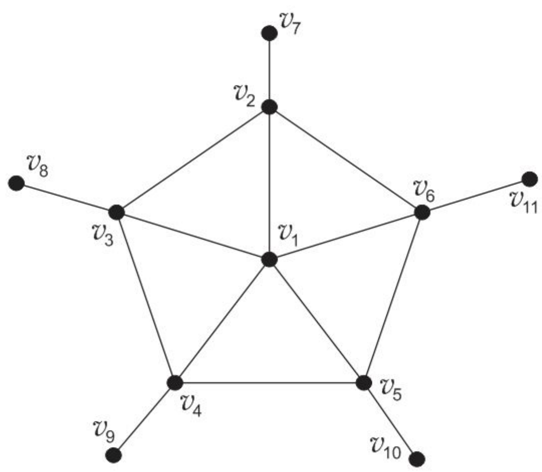


Gambar 3.4. Rangkaian Jaringan Komputer

Representasikan rangkaian jaringan tersebut dalam graf dan berilah label pada setiap vertex di graf tersebut dengan menggunakan algoritme Kamalesh dan Srivatsa.

Penyelesaian:

1. Merepresentasikan rangkaian jaringan komputer ke dalam Graf H_5 dengan $V(H_5) = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}\}$ seperti pada Gambar 3.5.



Gambar 3.5. Graf H_5

2. Menentukan *minimum spanning tree* dengan menggunakan metode BFS Moore.

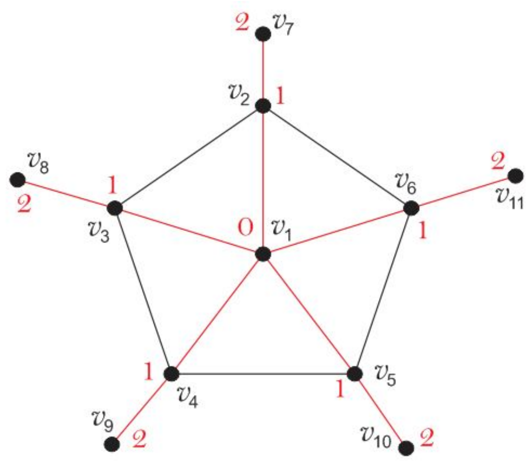
   Langkah-langkah pelabelan graf H_5 dengan algoritme *BFS Moore* untuk membentuk *minimum spanning tree* H'_5 dari graf H_5 sebagai berikut.

   (a) Memilih *vertex* v_1 sebagai *vertex* awal dan diberi label 0 serta *vertex* lainnya yang belum berlabel diberi label ∞.

   (b) *Vertex* yang *adjacent* dengan *vertex* v_1 yaitu *vertex* v_2, v_3, v_4, v_5, dan v_6 dilabeli dengan 1 dan *vertex* lain yang belum berlabel diberi label ∞.

   (c) *Vertex* yang *adjacent* dengan *vertex* berlabel 1 yaitu *vertex* v_7, v_8, v_9, v_{10}, dan v_{11}, dilabeli dengan 2. Semua *vertex* sudah berlabel maka berhenti.

   Berikut ini diperoleh *minimum spanning tree* H'_5 dari graf H_5 dengan menggunakan metode BFS Moore ditunjukkan pada Gambar 3.6.



Gambar 3.6. Graf berlabel H'_5

3. Mencari nilai eksentrisitas setiap *vertex* dari *minimum spanning tree* pada graf tersebut. Berikut ini nilai eksentrisitas setiap *vertex* dari *minimum spanning tree* H'_5 dari graf H_5 ditunjukkan pada Tabel 3.2.

Tabel 3.2 Eksentrisitas dari *minimum spanning tree* H_5'

vertex	eksentrisitas *vertex*	*vertex* eksentrik
v_1	$e(v_1) = 2$	$v_7, v_8, v_9, v_{10}, v_{11}$
v_2	$e(v_2) = 3$	v_8, v_9, v_{10}, v_{11}
v_3	$e(v_3) = 3$	v_7, v_9, v_{10}, v_{11}
v_4	$e(v_4) = 3$	v_7, v_8, v_{10}, v_{11}
v_5	$e(v_5) = 3$	v_7, v_8, v_9, v_{11}
v_6	$e(v_6) = 3$	v_7, v_8, v_9, v_{10}
v_7	$e(v_7) = 4$	v_8, v_9, v_{10}, v_{11}
v_8	$e(v_8) = 4$	v_7, v_9, v_{10}, v_{11}
v_9	$e(v_9) = 4$	v_7, v_8, v_{10}, v_{11}
v_{10}	$e(v_{10}) = 4$	v_7, v_8, v_9, v_{11}
v_{11}	$e(v_{11}) = 4$	$v_7, v_8, v_9, v_{10},$

4. Memberikan label baru sesuai aturan pada algoritme Kamalesh dan Srivatsa. Penomoran *vertex* dari *minimum spanning tree* H_5' dari graf H_5 adalah sebagai berikut,

$$\begin{aligned} \lambda(v_1) &= 1 & \lambda(v_7) &= 7 \\ \lambda(v_2) &= 2 & \lambda(v_8) &= 8 \\ \lambda(v_3) &= 3 & \lambda(v_9) &= 9 \\ \lambda(v_4) &= 4 & \lambda(v_{10}) &= 10 \\ \lambda(v_5) &= 5 & \lambda(v_{11}) &= 11 \\ \lambda(v_6) &= 6 \end{aligned}$$

Sehingga diperoleh hasil akhir penyelesaian masalah yang ditunjukkan pada Tabel 3.3.

Tabel 3.3 Jarak setiap *vertex* dalam *minimum spanning tree* H'_5

vertex	$e(v_i)$	nomor *vertex*
v_1	$e(v_1) = 2$	1
v_2	$e(v_2) = 3$	2
v_3	$e(v_3) = 3$	3
v_4	$e(v_4) = 3$	4
v_5	$e(v_5) = 3$	5
v_6	$e(v_6) = 3$	6
v_7	$e(v_7) = 4$	7
v_8	$e(v_8) = 4$	8
v_9	$e(v_9) = 4$	9
v_{10}	$e(v_{10}) = 4$	10
v_{11}	$e(v_{11}) = 4$	11

BAB IV
KESIMPULAN

Berdasarkan pembahasan diperoleh kesimpulan yaitu,

1. dalam penyusunan suatu jaringan yang efisien, dapat diselesaikan menggunakan graf. Dimana graf tersebut merepresentasikan topologi jaringan,

2. penomoran vertex pada graf *wheel* dan graf *helm*, dapat diselesaikan menggunakan algoritme Kamalesh dan Srivatsa, dan

3. konsep penomoran vertex pada jaringan graf *wheel* di contoh 3.1 dan graf *helm* di contoh 3.2 dapat diterapkan dalam penyelesaian masalah jaringan komputer dengan menggunakan algoritme Kamalesh dan Srivatsa.

DAFTAR RUJUKAN

[1] Behrouz, Forouzan and Firouz, Mosharraf, *Foundations of Computer Science*, Cengange Learning EMEA, United Kingdom, 2008.

[2] Chartrand, G. and L. Lesniak, *Graphs and Digraphs*, 2^{nd} ed., Wadsworth Inc., California, 1979.

[3] Bondi, J. and Murty, *Graf with Application*, New York.

[4] Munir, Rinaldi,*Matematika Diskrit*, Informatika Bandung, Bandung, 2010.

[5] Chartrand, G., Eroh, L., Johnson, M., and Oellermann, O., *Resolvability in Graphs and the Metric Dimension of Graph*, Discrete Appl. Math. 105 (2000). 99-113.

[6] Gallian, Joseph. A. , *A Dynamic Survey Of Graph Labeling, Eletronic Journal of Combinatorics : Dinamic Surveys, 2018.*

[7] Wallis, W. D., *Magic Graphs*, Birkhauser, Boston, 2001.

[8] Kamalesh, V. N. dan Srivatsa, S. K. S., *Node Numbering in a Topological Structure of Interconnection Network*, Indian Journal of Science Technologi, 2 (2009), no. 11, 37-40.

[9] Sidiq M, Kusmayadi T. A., dan Kuntari S., *Pemberian Nomor Vertex pada Topologi Jaringan Graf Wheel, Graf Helm dan Graf Lollipop*, Universitas Sebelas Maret, 2014.

Job Description

Nama	***Job Description***
Abdul Jalil Nawawi	Pendahuluan dan kesimpulan, Penyusunan *latex*, penyusunan *powerpoint*, *Editor*, dan *proof reader*
Noni Kurnia Dewi	Penerapan contoh 3.2, penyusunan *latex*, penyusunan *powerpoint*, *Editor* dan *proof reader*
Nur Fadila Marsaoly	Landasan teori, penyusunan *latex*, penyusunan *powerpoint*, *Editor* dan *proof reader*
Tiyas Astutiningsih	Landasan teori, penyusunan *latex*, penyusunan *powerpoint*, *Editor* dan *proof reader*
Tri Hartati	Penerapan contoh 3.1, penyusunan *latex*, penyusunan *powerpoint*, *Editor* dan *proof reader*