

BAB I TINJAUAN PUSTAKA

A. Penelitian Terkait

Mengenai adanya penelitian-penelitian yang telah dilakukan sebelumnya. Hasil penelitian-penelitian tersebut menjadi referensi pada penelitian pada penelitian ini.

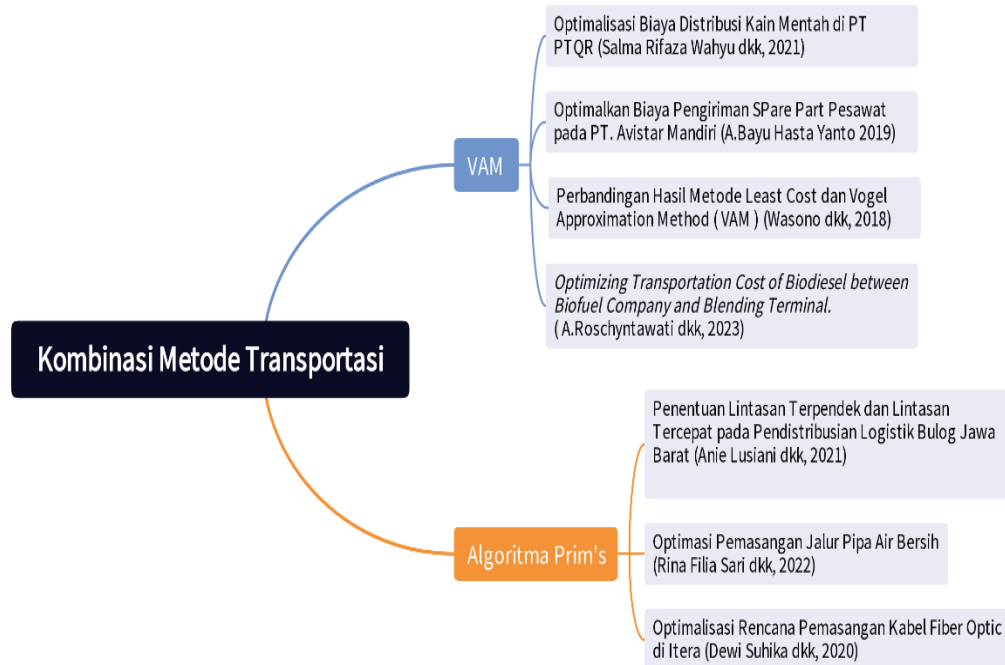
Tabel 1 Contoh tabel penelitian terkait

No (1)	Peneliti (2)	Keterangan (3)
1	Diah Rahmawati (2022)	Judul : Implementasi Metode Vogel's Approximation Method (VAM) dalam Optimasi Biaya Transportasi Metode : Algoritma VAM Hasil : Penelitian ini menghasilkan alokasi sumber daya yang lebih optimal dengan biaya yang lebih rendah
2	Aqilah Kamalia, Robertus Heri Soelistyo (2020)	Judul : Penyelesaian Masalah Transportasi Menggunakan Metode RCWMCAM dan Metode MODI Metode : Metode RCWMCAM dan Metode MODI Hasil : Penelitian ini menghasilkan model kombinasi algoritma RCWMCAM dan MODI yang mengurangi biaya transportasi dan meningkatkan efisiensi distribusi barang.
3	Salma Rifaza Wahyu, Ana Rohima, Kristina Fitri Handayani, Muchammad Fauzi (2021)	Judul : Optimalisasi Biaya Distribusi Kain Mentah di PT PQR Menggunakan Metode VAM dan Lingo Metode : Metode Vam dan Lingo Hasil : Dihasilkan solusi yang optimum dengan biaya pendistribusian minimum.
4	Andika Bayu Hasta Yamto (2019)	Judul : Penerapan Metode VAM dalam Optimalkan Biaya Pengiriman Spare Part Pesawat pada PT. Aviastar Mandiri Metode : Metode VAM Software : Pom QM

		Hasil : Dihasilkan biaya pengiriman spare pesawat yang optimal dengan validasi aplikasi pom QM.
5	Wasono, Fidia Deny Tisna A., Nanda Arista Rizki (2018)	Judul : Perbandingan Hasil Metode <i>Least Cost</i> dan <i>Vogel Approximation Method</i> (VAM) dalam Meminumkan Biaya Pendistribusian Tabung Gas LPG 3 Kg Pada PT. Tri Pribumi Sejati Samarinda Metode : Metode VAM dan Least Cost Hasil : Dalam penelitian ini Algoritma <i>Least Cost</i> menghasilkan solusi lebih optimum dan efisien dalam mengatasi masalah transportasi disbanding algoritma VAM
6	A.Roschyntawati, R. Pujiwat, D. P. Upahita, Y. N. R. Hendra, R. Fajar1, M. I. A. Saputro and A. Kismanto (2023)	Judul : Application of North West Corner, Least Cost and Vogel's Approximation Method for Optimizing Transportation Cost of Biodiesel between Biofuel Company and Blending Terminal. Kombinasi metode NWC, Lest Cost, dan VAM menghasilkan biaya transpostasi yang minimum dengan presentase 31,05%. Metode : Metode NWC, Lest Cost, dan VAM. Hasil : Dalam penelitian ini dihasilkan model Kombinasi metode NWC, Lest Cost, dan VAM menghasilkan biaya transpostasi yang minimum dengan presentase 31,05%.
7	Anie Lusiani, Euis Sartika, Endang Habinuddin, Agus Binarto, Irfan Azis (2021)	Judul : Algoritma Prim dalam Penentuan Lintasan Terpendek dan Lintasan Tercepat pada Pendistribusian Logistik Bulog Jawa Barat Metode : Prim Hasil : Berdasarkan minimum spanning tree yang diperoleh lintasan terpendek dan lintasan tercepat dari gudang Cibitung Bekasi ke gudang Bojong Cianjur berturut-turut 85,5 km dan 2 jam 34 menit.

8	Rina Filia Sari, Rina Widyasari, Fithtria Aidra Marpanng (2022)	Judul : Optimasi Pemasangan Jalur Pipa Air Bersih Melalui Minimum Spanning Tree Dengan Algoritma Prim Metode : Prim Hasil : Dalam penelitian ini hasil minimum spanning tree dengan menggunakan algoritma Prim secara manual yaitu diperoleh total panjang pipa yang akan dipasang sebesar 9.795 meter dengan 52 titik dan 56 sisi sehingga diperoleh biaya minimum sebesar Rp. 1.732.850.000 (satu miliar tujuh ratus tiga puluh dua juta delapan ratus lima puluh ribu).
9	Zindy Anitasari, Wasono Wasono, Fidia Deny Tisna Amijaya (2024)	Judul : Optimalisasi Biaya Transportasi Pendistribusian Produk Frozen Food Menggunakan Metode Vogel's Approximation dan Metode Stepping Stone Metode : Metode Vogel's Approximation dan Stepping Stone Hasil : Dalam penelitian ini menghasilkan solusi optimal menggunakan kombinasi VAM dan Stepping Stone
10	Dewi Suhika, Triyana Muliawati, Heru Ruwandar (2020)	Judul : Optimalisasi Rencana Pemasangan Kabel <i>Fiber Optic</i> di Itera dengan Algoritma Prim. Model: Prim Hasil: Hasil penelitian menunjukan jika menggunakan algoritma Prim diperoleh minimum spanning tree dengan total jarak 16.503 meter, dari graf awal ada 88 titik dan 84 sisi dengan total panjang jaringan 18.448 meter. Ada perbedaan total jarak yaitu 1945 meter, sehingga dapat disimpulkan bahwa algoritma Prim dapat digunakan sebagai metode untuk menemukan jaringan yang optimal pada jaringan FO ITERA.

Diagram *mind map* dicontohkan pada gambar berikut ini



Gambar 1 Mind map penelitian terkait

B. Landasan Teori

1. Optimasi

Optimalisasi merupakan suatu proses untuk mencari nilai terbaik. Tujuan dari optimalisasi ini adalah mencari rute. Pencarian rute dilakukan dengan meminimalkan jarak tempuh, waktu yang digunakan akan menjadi lebih singkat dan tidak terjadi pemborosan biaya [13].

2. Masalah Transportasi

Masalah transportasi yang dimaksudkan pada penelitian ini yaitu mengirimkan barang dari suatu sumber ke tujuan secara optimal dengan mempertimbangkan biaya minimal, pengalokasian dan rute yang efektif [14]. Masalah transportasi klasik adalah menentukan jadwal optimal dari pengiriman yang [15] :

- 1) Berasal dari sumber-sumber yang memiliki jumlah barang tertentu yang tersedia.

- 2) Barang dialokasikan dan dikirim langsung ke masing-masing tujuan di mana jumlah total yang diterima harus sama dengan jumlah total yang dibutuhkan.
- 3) *Supply* dan *demand* harus dihabiskan. Oleh karena itu total permintaan yang diberikan harus sama dengan total persediaan yang diberikan.
- 4) Biaya setiap pengiriman dari sumber ke tujuan menyesuaikan dengan kapasitas gudang dan total biaya adalah jumlah biaya dari semua biaya.

Selanjutnya, terkait dengan masalah transportasi berikut ini disajikan model matematisnya. Dimisalkan m adalah jumlah barang yang harus dikirim ke tujuan n untuk memenuhi permintaan. Khususnya sumber i memuat barang berjumlah a_i , dan tujuan j membutuhkan barang sebesar b_j . Diasumsikan bahwa system tersebut seimbang, artinya total pasokan sama dengan total permintaan [15]. Dapat dirumuskan sebagai berikut :

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (1)$$

Sehingga secara model matematis diperoleh [16],

$$\text{Meminumkan } Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} X_{ij} \quad (2)$$

Dengan kendala

$$\sum_{j=1}^n X_{ij} = S_i, \quad i = 1, 2, 3, \dots, m, \quad (3)$$

$$\sum_{i=1}^m X_{ij} = t_j, \quad j = 1, 2, 3, \dots, n, \quad (4)$$

$$X_{ij} \geq 0 \quad (i = 1, 2, 3, \dots, m \text{ dan } j = 1, 2, 3, \dots, n).$$

dengan

- Z : Biaya total transportasi.
 X_{ij} : Satuan barang yang harus didistribusikan dari sumber ii ke tujuan j .

- C_{ij} : Biaya distribusi per satuan barang dari sumber ii ke tujuan j .
 s_i : Banyaknya barang yang tersedia di tempat asal ke $-ii$.
 t_j : Banyaknya permintaan barang di tempat tujuan ke- j .
 m : Jumlah tempat asal.
 n : Jumlah tempat tujuan.

3. *Vogel's Approximation Method (VAM)*

Vogel's Approximation Method adalah salah satu metode transportasi yang populer karena pada penerapannya menghasilkan solusi mendekati optimal atau bahkan optimal [1]. Adapun beberapa asumsi yang dimiliki algoritma *vogel's Approximation Method (VAM)* yaitu :

- 1) Ada penawaran dan permintaan.
- 2) Jumlah permintaan dan penawaran sama.
- 3) Terdapat gudang dan tujuan.
- 4) Adanya biaya distribusi.

Berikut langkah-langkah VAM [16]:

- a) Hitung penalti untuk setiap baris dan kolom. Penalti untuk setiap baris i diperoleh dari selisih nilai C_{ij} terkecil pada baris tersebut dari nilai C_{ij} satu tingkat lebih besar pada baris yang sama. Penalti kolom diperoleh dengan cara yang sama.
- b) Pilih baris atau kolom dengan penalti terbesar (jika terdapat nilai yang sama, pilih secara sembarang). Alokasikan sebanyak mungkin ke kotak dengan nilai C_{ij} minimum pada baris atau kolom yang dipilih.
- c) Sesuaikan penawaran dan permintaan untuk menunjukkan alokasi yang sudah dilakukan. Hilangkan sel pada baris (kolom) ke i (j) ketika penawaran dan permintaan sudah habis.
- d) Ulangi langkah a sampai c jika penawaran dan permintaan belum habis.

Berikut contoh implementasi VAM,
 PT.Sejahtera memiliki tiga gudang dan tiga agen, berikut tabel biaya distribusi

Tabel 2 Biaya Distribusi PT. Sejahtera

	Agen D	Agen E	Agen F	Supply
Gudang A	200.000	60.000	50.000	1.000.000
Gudang B	150.000	200.000	100.000	500.000
Gudang C	250.000	90.000	180.000	50.000
Demand	500.000	1.100.000	400.000	

Penyelesaian

Tabel 3 Iterasi 1 Algoritma VAM

	Agen D	Agen E	Agen F	Supply	
Gudang A				1.000.000	60.000 – 50.000 = 10.000
	200.000	60.000	50.000		
Gudang B				500.000	150.000 – 100.000 = 50.000
	150.000	200.000	100.000		
Gudang C	-	500.000	-	500.000- 500.000 = 0	180.000 – 90.000 = 90.000
	250.000	90.000	180.000		
Demand	500.000	1.100.000-500.000 = 600.000	400.000		
	200.000 – 150.000 = 50.000	90.000 – 60.000 = 30.000	100.000 – 50.000 = 50.000		

Tabel 4 Iterasi 2 Algoritma VAM

Iterasi 2

	Agen D		Agen E		Agen F		Supply	
Gudang A							1.000.000	60.000 – 50.000 = 10.000
	200.000		60.000		50.000			
Gudang B							500.000	150.000 – 100.000 = 50.000
	150.000		200.000		100.000			
Gudang C	-		500.000		-		0	
	250.000		90.000		180.000			
Demand	500.000		600.000		400.000			
200.000 – 150.000 = 50.000		200.000 - 60.000=140.000		100.000 – 50.000 = 50.000				

Tabel 5 Iterasi 3 Algoritma VAM

Iterasi 3

	Agen D		Agen E		Agen F		Supply	
Gudang A			600.000				1.000.000-600.000 =400.000	60.000 – 50.000 = 10.000
	200.000		60.000		50.000			
Gudang B							500.000	150.000 – 100.000 = 50.000
	150.000		200.000		100.000			
Gudang C	-		500.000		-		0	
	250.000		90.000		180.000			
Demand	500.000		600.000- 600.000=0		400.000			
200.000 – 150.000 = 50.000		200.000 - 60.000=140.000		100.000 – 50.000 = 50.000				

Tabel 6 Iterasi 4 Algoritma VAM

Iterasi 4

	Agen D		Agen E		Agen F		Supply
Gudang A			600.000				400.000
	200.000		60.000		50.000		
Gudang B							500.000
	150.000		200.000		100.000		
Gudang C	-		500.000		-		0
	250.000		90.000		180.000		
Demand	500.000		0		400.000		

$$200.000 - 150.000 = 50.000$$

$$100.000 - 50.000 = 50.000$$

$$200.000 - 50.000 = 150.000$$

$$150.000 - 100.000 = 50.000$$

Tabel 7 Iterasi 5 Algoritma VAM

Iterasi 5

	Agen D		Agen E		Agen F		Supply
Gudang A			600.000		400.000		400.000-400.000=0
	200.000		60.000		50.000		
Gudang B							500.000
	150.000		200.000		100.000		
Gudang C	-		500.000		-		0
	250.000		90.000		180.000		
Demand	500.000		0		400.000-400.000=0		

$$200.000 - 150.000 = 50.000$$

$$100.000 - 50.000 = 50.000$$

$$200.000 - 50.000 = 150.000$$

$$150.000 - 100.000 = 50.000$$

Tabel 8 Iterasi 6 Algoritma VAM

Iterasi 6

	Agen D	Agen E	Agen F	Supply	
Gudang A	-	600.000	400.000	0	150.000
	200.000	60.000	50.000		
Gudang B	-	-	-	500.000	150.000
	150.000	200.000	100.000		
Gudang C	-	500.000	-	0	150.000
	250.000	90.000	180.000		
Demand	500.000	0	0		

Tabel 9 Iterasi 7 Algoritma VAM

Iterasi 7

	Agen D	Agen E	Agen F	Supply	
Gudang A	-	600.000	400.000	0	150.000
	200.000	60.000	50.000		
Gudang B	500.000	-	-	500.000- 500.000=0	150.000
	150.000	200.000	100.000		
Gudang C	-	500.000	-	0	150.000
	250.000	90.000	180.000		
Demand	500.000- 500.000=0	0	0		

Tabel 10 Iterasi 8 Algoritma VAM

Iterasi 8

	Agen D	Agen E	Agen F	Supply
Gudang A	-	600.000	400.000	0
	200.000	60.000	50.000	
Gudang B	500.000	-	-	0
	150.000	200.000	100.000	
Gudang C	-	500.000	-	0
	250.000	90.000	180.000	
Demand	0	0	0	

Dari Tabel di atas diperoleh total biaya minimum :

$$\begin{aligned}
 Z &= 500000 \times 150000 + 600000 \times 60000 + 500000 \times 90000 + 400000 \times 50000 \\
 &= 45.000.000.000 + 36.000.000.000 + 45.000.000.000 + 2000.000.000 \\
 &= 1.460.000.000.000
 \end{aligned}$$

4. Teori Dasar Graf

Menurut Chartrand [2] terkait dengan definisi dari beberapa dasar graf, seperti definisi gra , $u - v$ walk, $u - v$ trail, $u - v$ path, cycle, adjacent, incident dan graf terhubung.

Definisi 2.1 Graf G adalah himpunan tak kosong berhingga $V(G)$ disertai dengan relasi yang *irreflexive* dan *symmetric* R pada V .

Dalam graf G , $V(G)$ adalah himpunan *vertex* dan banyaknya *vertex* dalam G dinotasikan orde G atau $|V(G)|$, sedangkan setiap elemen $E(G)$ adalah himpunan *edge* dan banyaknya *edge* dalam G disebut *size* G .

Definisi 2.2 Dua *vertex* u dan *vertex* v dikatakan *adjacent* di graf G jika $uv \in E(G)$. Apabila suatu *vertex* u *incident* dengan *edge* e dan *vertex* v *incident* dengan *edge* e , jika $e = uv \in E(G)$, maka u dan v adalah *incident* dengan e .

Definisi 2.3 Suatu $u-v$ walk pada suatu graf G adalah barisan bergantian dari *vertex* dan *edge* G , dimulai dari *vertex* u dan diakhiri dengan *vertex* v , dengan setiap *edge* menghubungkan *vertex* terhadap barisan tersebut. Suatu $u-v$ trail pada suatu graf adalah lintasan $u-v$ walk yang tidak mengulang *edge* manapun, sedangkan suatu $u-v$ path adalah lintasan $u-v$ walk yang tidak mengulang *vertex* manapun.

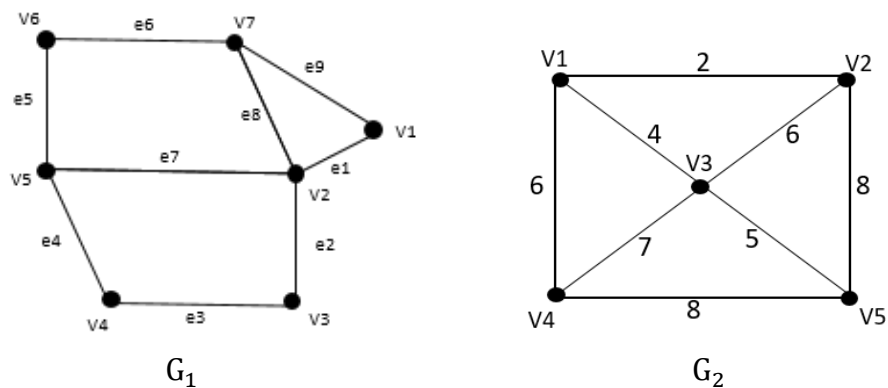
Definisi 2.4 Graf dikatakan terhubung jika untuk sembarang dua *vertex* dalam graf G terdapat suatu *path* yang menghubungkan dua *vertex* tersebut. Apabila tidak terjadi hal tersebut, maka graf tersebut tidak terhubung.

Definisi 2.5 *Circuit* adalah $u-v$ trail dimana $u = v$ yang mengandung setidaknya tiga *edge*, dengan kata lain sebuah *circuit* harus berakhir pada *vertex* yang sama dengan *vertex* awal. Sedangkan *cycle*, yaitu sebuah *circuit* yang tidak mengulangi setiap *vertex* (kecuali yang *vertex* pertama dan *vertex* yang terakhir).

Definisi 2.6 Suatu graf G dikatakan berbobot apabila setiap *edgenya* diberikan sebuah bobot (nilai).

Definisi 2.7 Jarak $d(u,v)$ dari u ke v adalah panjang $u-v$ path terpendek.

Definisi 2.8 Kardinalitas $|G|$ adalah banyaknya *vertex* pada graf G .



Gambar 2 Graf G_1 dan graf G_2

Gambar 1 graf G_1 merupakan contoh graf terhubung dengan $|V(G)| = 7$ dan $|E(G)| = 9$. Dari graf G_1 didapatkan bahwasannya v_1 adjacent dengan v_2 dan v_7 , e_1 incident dengan *vertex* v_1 dan v_2 . Contoh dari $u-v$ walk adalah $v_1, e_1, v_2, e_2, v_3, e_3, v_4$. Contoh dari $u-v$ trail adalah $v_1, e_1, v_2, e_8, v_7, e_9, v_1$. Contoh dari $u-v$ path adalah $v_1, e_1, v_2, e_7, v_5, e_5, v_6$. Contoh *cycle* adalah $v_1, e_1, v_2, e_8, v_7, e_9, v_1$. Contoh dari *circuit* adalah $v_1, e_1, v_2, e_2, v_3, e_3, v_4, e_4, v_5, e_7, v_2, e_8, v_7, e_9, v_1$. Sedangkan pada graf G_2 merupakan contoh dari graf berbobot atau setiap *edgenya* mempunyai nilai. Contoh dari jarak, misalkan dari $v_1 \rightarrow v_2$ adalah 2.

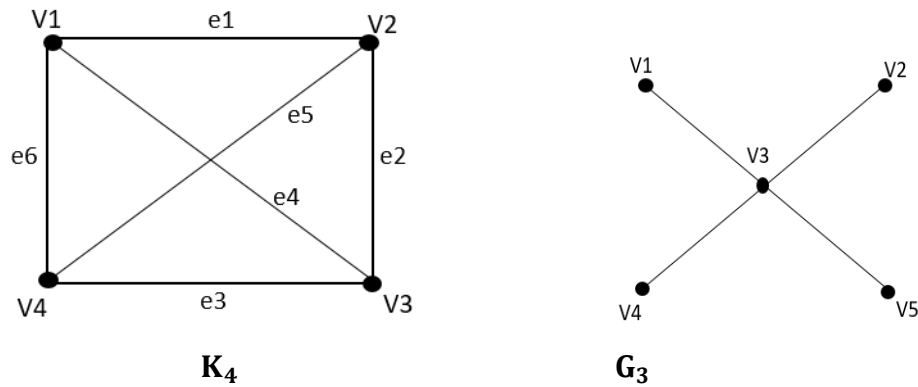
5. Jenis – Jenis Graf

Jenis Graf yang akan dibahas adalah definisi dari graf lengkap dan graf pohon :

Definisi 2.9 Graf lengkap adalah graf sederhana yang setiap *vertex* mempunyai *edge* ke semua *vertex* lainnya. Notasi dari graf lengkap dengan n buah *vertex* adalah K_n .

Definisi 2.10 Graf pohon adalah graf terhubung sederhana yang tidak mengandung sirkuit.

Contoh graf lengkap dan graf pohon :



Gambar 3 Graf lengkap K_4 dan graf pohon G_3

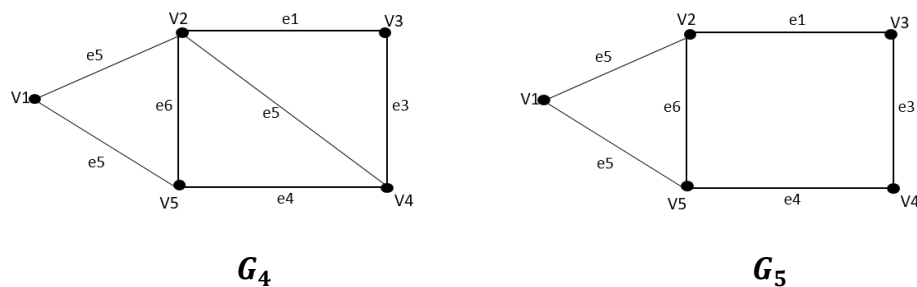
Pada gambar 3 graf K_4 merupakan contoh dari graf lengkap dengan 4 buah *vertex*, sedangkan graf G_3 adalah contoh dari graf pohon.

6. Pohon Rentang Minimum (*Minimum Spanning Tree*)

Berikut dijelaskan definisi dari sub graf, graf pohon, *spanning tree* dan *minimum spanning tree* menurut G. Chartrand, L. Lesniak, dan P. Zhang[2] :

Definisi 2.11 Sebuah graf H disebut subgraf dari graf G jika $V(H) \subseteq V(G)$ dan $E(H) \subseteq E(G)$, yang mana dapat ditulis $H \subseteq G$. Jika $V(H) = V(G)$ maka H adalah *spanning subgraph* dari G .

Contoh *spanning subgraph* :

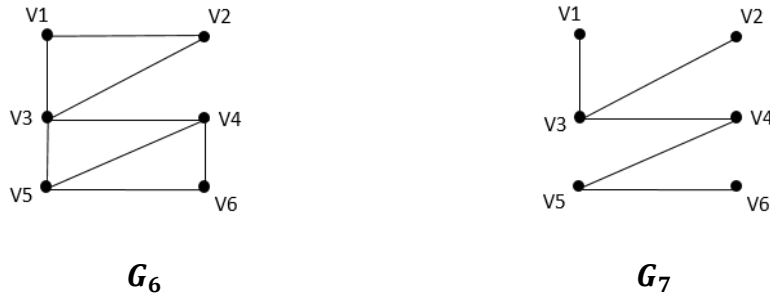


Gambar 4 Graf G_4 dan G_5

Berdasarkan graf G_5 merupakan *spanning subgraph* dari graf G_4 karena $V(H) \subseteq V(G)$.

Definisi 2.12 Sebuah *spanning tree* dari graf G adalah spanning subgraf dari G yang berupa graf pohon.

Contoh graf *spanning tree* :

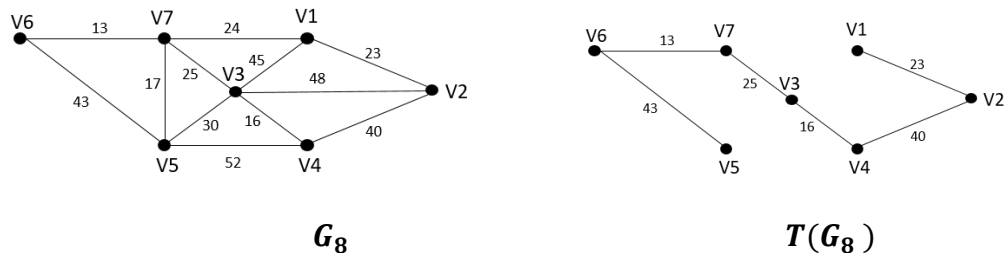


Gambar 5 Graf G_6 dan G_7

Berdasarkan gambar G_7 merupakan *spanning tree* dari graf G_6 .

Definisi 2.13 Sebuah *Spanning tree* dari graf G yang bobotnya minimum diantara semua *spanning tree* dari graf G disebut *minimum spanning tree*.

Contoh graf *minimum spanning tree* :



Gambar 6 Graf G_8 dan graf pohon $T(G_8)$

Pada gambar 5 Graf $T(G_4)$ merupakan contoh graf *minimum spanning tree* dari graf G_4 .

7. Algoritma Prim

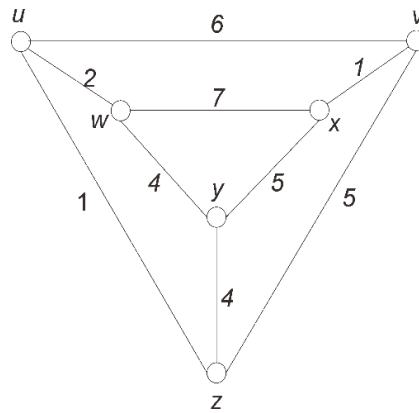
Menurut Chartrand [2] ,untuk graf terhubung berbobot G yang berorde n , *minimum spanning tree* T dikonstruksi sebagai berikut : Ambil sembarang *vertex* u dalam graf G , pilih *edge* dengan bobot minimum yang berhubungan dengan *vertex* u (e_1 dari graf T), untuk *edge* selanjutnya ($e_2, e_3, \dots, e_{n-1}$) dipilih *edge* dengan bobot minimum tepat satu dari *vertex* yang sudah dipilih.

Proses algoritma prim's adalah sebagi berikut :

- Pilih *vertex* awal $V(G) = V(T(G))$.

- b. i. pilih *edge* yang *incident* dengan $V(T(G))$ yang memiliki bobot terkecil, misalkan e_1 yang tidak menimbulkan *cycle*, jika menimbulkan *cycle* maka pilih *edge* yang *incident* dengan bobot terkecil ke-2.
- ii. pilih *vertex* yang *adjacent* dengan $V(T(G))$ dan *incident* dengan e_1 .
- c. Ulangi langkah bi dan bii hingga $E(T(G)) = n-1$.

Contoh algoritma Prim :



Gambar 7 Graf G_9

Berikut ini adalah penyelesaian *minimum spanning tree* menggunakan algoritma prim

Tabel 11 Iterasi Algoritma Prim

Iterasi	Vertex	Bobot	Graf
0	v	-	v •
1	$T:\{v\}$		
	v - u	6	
	v - x	1	
	v - z	5	
Dipilih	v - x	1	
2	$T:\{v,x\}$	1	
	v - u	6	

	v - z	5	
	x - y	5	
	x - w	7	
Dipilih	v - z	5	
3	T: { v,x,z }	1+5 = 6	
	v - u	6	
	z - y	4	
	z - u	1	
Dipilih	z - u	1	
4	T: { v,x,z,u }	1+5+1=7	
	z - y	4	
	u - w	2	
Dipilih	u - w	2	
5	T: { v,x,z,u,w }	1+5+1+2=9	
	z - y	4	
dipilih	z - y	4	

6	T: { v,x,z,u,w}	$1+5+1+2+4=13$	
---	--------------------	----------------	--

Hasil bobot minimum spanning tree dengan menggunakan algoritma prim's sesuai tabel di atas adalah $1+5+1+2+4=13$