

# PENYELESAIAN MASALAH TRANSPORTASI UNTUK Mencari SOLUSI OPTIMAL DENGAN Pendekatan *MINIMUM SPANNING TREE (MST)* MENGGUNAKAN ALGORITMA KRUSKAL DAN ALGORITMA PRIM

Yusufiani Nurlinawati Dili<sup>1, a)</sup>; Elis Ratna Wulan<sup>2, b)</sup>; Fadilah Ilahi<sup>3, c)</sup>

<sup>1,2,3</sup>Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Gunung Djati Bandung

<sup>a)</sup>email: [fianurlina04@gmail.com](mailto:fianurlina04@gmail.com)

<sup>b)</sup>email: [elis\\_ratna\\_wulan@uinsgd.ac.id](mailto:elis_ratna_wulan@uinsgd.ac.id)

<sup>c)</sup>email: [fadilah.ilahi@uinsgd.ac.id](mailto:fadilah.ilahi@uinsgd.ac.id)

## Abstrak

Penelitian ini membahas penyelesaian masalah transportasi dengan pendekatan *Minimum Spanning Tree (MST)* menggunakan algoritma Kruskal dan algoritma Prim untuk mencari solusi optimal. Algoritma Kruskal dan algoritma Prim merupakan algoritma dalam teori graf untuk mencari *Minimum Spanning Tree (MST)*. Langkah algoritma Kruskal yaitu mengurutkan biaya dari yang terkecil hingga terbesar. Selanjutnya, pilih biaya yang paling terkecil. Kemudian, lakukan perhitungan dengan melihat sumber persediaan dan permintaan di setiap tujuan sampai semuanya terpenuhi, sehingga terlihat bentuk *Minimum Spanning Tree (MST)* dari algoritma Kruskal. Sedangkan langkah algoritma Prim yaitu dengan memilih sembarang titik atau sumber. Selanjutnya, pilih *active edge* dengan biaya terkecil. Kemudian, lakukan perhitungan dengan melihat sumber persediaan dan permintaan di setiap tujuan sampai semuanya terpenuhi, sehingga terlihat bentuk *Minimum Spanning Tree (MST)* dari algoritma Prim. Bentuk dari *Minimum Spanning Tree (MST)* menghasilkan solusi yang optimal. Dari hasil penelitian ini, pendekatan *Minimum Spanning Tree (MST)* dengan algoritma Prim yang lebih unggul karena nilai solusi optimal dan jumlah iterasinya paling kecil.

*Kata kunci: Minimum Spanning Tree (MST), Graf, Solusi Optimal, Algoritma Kruskal, Algoritma Prim*

## Abstract

*This Research discusses solving transportation problems with the Minimum Spanning Tree (MST) approach using Kruskal's algorithm and Prim's algorithm to find optimal solutions. Kruskal's algorithm and Prim's algorithm are algorithms in graph theory to find the Minimum Spanning Tree (MST). The steps of Kruskal's algorithm are to sort the costs from highest to largest. Next, choose the highest cost. Then, do the calculations by looking at the source of supply and demand at each destination until everything is met, so that the Minimum Spanning Tree (MST) form of the Kruskal algorithm is seen. Meanwhile, the Prim algorithm step is to choose any point or source. Further, choose an active edge that costs forever. Thus, perform calculations by looking at the source of supply and demand at each destination until everything is met, so that the Minimum Spanning Tree (MST) form of the Prim algorithm is visible. The form of the Minimum Spanning Tree (MST) produces an optimal solution. From*

*the results of this study, the Minimum Spanning Tree (MST) approach with the Prim algorithm is superior because of the optimal solution value and the smallest number of iterations.*

*Keywords: Minimum Spanning Tree (MST), Graph, Optimal Solution, Kruskal Algorithm, Prim Algorithm*

## Pendahuluan

Persoalan jaringan kerja muncul pada sejumlah perencanaan dan berbagai bidang. Jaringan transportasi, listrik, dan komunikasi suatu yang dijumpai dalam kehidupan sehari-hari. Khusus masalah transportasi umumnya digunakan untuk mengatur distribusi dari beberapa sumber yang menyediakan produk yang sama ke beberapa tempat tujuan secara optimal dengan biaya transportasi minimum. Model transportasi dari sebuah jaringan dengan  $m$  sumber dan  $n$  tujuan, dimana sumber dan tujuan diwakili oleh sebuah node. Busur yang menghubungkan semua sumber dan tujuan mewakili rute pengiriman barang tersebut.

Masalah transportasi sangat erat dengan pengukuran atau perhitungan mulai dari segi biaya hingga jarak. Begitu halnya Allah SWT menentukan ukuran dan kapasitas tertentu bagi makhluk berdasarkan hikmahnya. Karena Allah adalah pencipta, maka dia pula yang akan memenuhi seluruh kebutuhan makhluknya dan dia pula sumber seluruh potensi dan kemampuan seluruh makhluk. Sebagaimana firman-Nya dalam Surat Al-Hijr ayat 21 :

وَإِنْ مِنْ شَيْءٍ إِلَّا عِنْدَنَا خَزَائِنُهُ وَمَا نُنزِّلُهُ إِلَّا بِقَدَرٍ  
مَّعْلُومٍ

Artinya : “Dan tidak ada sesuatupun melainkan pada sisi Kami-lah khazanahnya, dan kami tidak menurunkannya melainkan dengan ukuran yang tertentu.”

Persoalan dalam jaringan kerja terdiri atas persoalan rute terpendek, persoalan pohon rentang minimum (*Minimum Spanning Tree*), dan persoalan aliran maksimum. Persoalan rute terpendek merupakan lintasan dengan bobot yang minimum. Dalam hal ini bobot dapat berupa biaya transportasi dari satu titik ke titik lainnya yang berbentuk rute tertentu. Sedangkan persoalan *Minimum Spanning Tree* merupakan variasi dari persoalan rute terpendek perbedaannya terletak pada lintasan yang dicari, yaitu menentukan sisi-sisi yang menghubungkan titik-titik yang ada pada jaringan sehingga diperoleh panjang sisi total minimum [1].

Masalah transportasi erat kaitannya dengan sistem produksi [2],[3]. Masalah *Minimum Spanning Tree* memiliki sejumlah penerapan praktis yang penting untuk diaplikasikan, salah satunya yaitu menentukan jalur transportasi apa yang harus disediakan untuk melayani seluruh lokasi dengan total biaya yang paling minimum [1]. Hal ini menyebabkan masalah transportasi dapat dikaitkan dengan *Minimum Spanning Tree (MST)*. *Minimum Spanning Tree (MST)* cocok diaplikasikan pada perusahaan yang mendistribusikan produknya secara langsung kepada konsumennya tanpa melalui jasa agen maupun distributor.

Oleh karena itu perusahaan perlu membuat perencanaan yang tepat berkaitan dengan rute jarak tempuh optimal dalam mendistribusikan produknya agar dapat menghemat biaya transportasi perusahaan. Jarak tempuh optimal dapat dicari menggunakan *Minimum Spanning Tree (MST)* yang dapat menghasilkan panjang sisi total yang minimal dengan menghubungkan secara langsung seluruh titik yang ada pada jaringan transportasi, yang dimana jaringan tersebut direpresentasikan dengan distribusi produk.

## Metode

*Minimum Spanning Tree* adalah salah satu model jaringan yang menjelaskan pemilihan hubungan antar titik sedemikian rupa sehingga jaringan hubungan terpendek itu akan membuat seluruh titik terhubung dengan panjang hubungan total terpendek. Saat ini ada beberapa algoritma untuk menyelesaikan masalah *Minimum Spanning Tree* (MST) diantaranya algoritma Kruskal dan algoritma Prim [1]. Suatu algoritma tidak saja harus menghasilkan keluaran yang benar, tetapi juga harus optimal dan efisien. Pada prinsipnya pendekatan *Minimum Spanning Tree* (MST) ini berusaha mendapatkan solusi terbaik yang nilainya sangat dekat dengan solusi optimal diantara kedua algoritma yang digunakan.

### 1. Algoritma Kruskal

Langkah awal algoritma Kruskal adalah sisi-sisi di dalam graf diurutkan terlebih dahulu berdasarkan bobot dari yang kecil ke besar. Sisi-sisi yang sudah urut berdasarkan bobot akan membentuk hutan dengan masing-masing pohon di hutan hanya berupa satu buah simpul, disebut dengan hutan rentang (*spanning forest*). Misalkan  $T$  adalah pohon rentang yang sisi-sisinya diambil dari graf  $G$ . Sisi yang dimasukkan ke dalam himpunan  $T$  adalah sisi graf  $G$  sedemikian sehingga  $T$  adalah pohon. Sisi dari graf  $G$  ditambahkan ke  $T$  jika ia tidak membentuk sirkuit atau siklus di  $T$  [4],[6].

Berikut ini adalah langkah-langkah algoritma Kruskal yaitu:

- 1) Urutkan secara menaik berdasarkan bobotnya untuk semua sisi-sisi graf.  $T$  masih kosong.
- 2) Pilih *edge*  $(u,v)$  dengan bobot minimum yang tidak membentuk sirkuit di  $T$ . Masukkan *edge*  $(u,v)$  ke dalam  $T$ .
- 3) Ulangi langkah nomor 2 sebanyak  $n-1$  kali. [5]

Berikut pendekatan *Minimum Spanning Tree* menggunakan algoritma Kruskal:

**Langkah 1:** Masukkan sumber dan tujuan ke dalam *node*

**Langkah 2 :** Urutkan *edge* dari terkecil hingga terbesar.

**Langkah 3 :** Pilih *edge* terkecil yang tidak mempunyai sirkuit

- i. Jika *edge* tidak mempunyai sirkuit lanjut ke langkah 4.
- ii. Jika *edge* mempunyai sirkuit ambil *edge* terkecil berikutnya.

**Langkah 4:** Pasang ke dalam *tree*.

**Langkah 5 :** Ulangi langkah 3 sampai *Minimum Spanning Tree* terbentuk, yaitu ketika *edge* berjumlah  $n-1$  ( $n$  adalah jumlah *node*).

### 2. Algoritma Prim

Algoritma Prim digunakan untuk mencari pohon merentang minimum dari graf terhubung berbobot dengan cara mengambil sisi/ruas garis yang memiliki bobot terkecil dari graf, dimana ruas garis tersebut bersisian dengan pohon terentang yang telah dibuat dan yang tidak membentuk siklus.

Berikut ini Langkah-langkah algoritma Prim sebagai berikut :

- 1) Ambil satu simpul sembarang (misalkan  $v_1 \in G$ ) dan masukkan simpul tersebut ke dalam graf  $T$  yang merupakan graf kosong.
- 2) Tambahkan satu rusuk yang terhubung dengan  $v_1$  dengan bobot yang paling minimum (misalkan  $e_1$ ) dan titik ujung lainnya ke  $T$  sehingga  $T$  terdiri dari sebuah rusuk  $e_1$  dan dua buah

simpul yang merupakan titik-titik ujung dari rusuk  $e_1$  (salah satu titik ujung harus memuat simpul  $v_1$ ).

- 3) Pada langkah berikutnya, pilihlah sebuah rusuk dalam  $E(G)$  yang bukan  $E(T)$  dengan syarat yang harus dipenuhi sebagai berikut:
  - i. Rusuk tersebut menghubungkan salah satu simpul  $V(T)$ .
  - ii. Rusuk tersebut mempunyai bobot minimal.
- 4) Ulangilah langkah tersebut (langkah 2-3) hingga diperoleh  $(n-1)$  rusuk dalam  $E(T)$  dengan  $n$  adalah banyaknya simpul dalam  $G$ . Dengan kata lain ulangi iterasi tersebut sebanyak  $(n-2)$  [7].

Berikut pendekatan Minimum Spanning Tree menggunakan algoritma Prim:

**Langkah 1:** Masukkan sumber dan tujuan ke dalam *node*.

**Langkah 2 :** Ambil sembarang *node*.

**Langkah 3 :** Pilih semua *edge* yang belum dilewati dari *node* tersebut ke dalam *active edge* selama *edge* tidak mempunyai sirkuit.

- i. Jika *edge* tidak mempunyai sirkuit lanjut ke langkah 4.
- ii. Jika *edge* mempunyai sirkuit ambil *edge* berikutnya.

**Langkah 4 :** Pilih *node* dengan *edge* terkecil di dalam *active edge*.

**Langkah 5 :** Pasang ke dalam *tree*

## Hasil dan Diskusi

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari *International Journal of Mathematics and Its Applications* dengan judul “Balance An Unbalanced Transportation Problem By A Heuristic Approach” [8], *International Journal of Science and Research* yang berjudul “An Approach for Solving Transportation Problem Using Modified Kruskal’s Algorithm” [9], *Journal Business and Management Research* dengan judul “Modified VOGEL Method to Find Initial Basic Feasible Solution (IBFS) Introducing a New Methodology to Find Best IBFS” [10], dan *International Journal of Agricultural and Statistical Sciences* yang berjudul “A Revised Version Of ASM-Method For Solving Transportation Problem” [11].

Keterangan :

Edge dan biaya yang dilewati kasus seimbang 3x5 algoritma Kruskal

Edge dan biaya yang dilewati kasus seimbang 3x5 algoritma Prim

Edge dan biaya yang dilewati kasus seimbang 5x4 algoritma Kruskal

Edge dan biaya yang dilewati kasus seimbang 5x4 algoritma Prim

Edge dan biaya yang dilewati kasus tak seimbang 3x4 algoritma Kruskal

Edge dan biaya yang dilewati kasus tak seimbang 3x4 algoritma Prim

Edge dan biaya yang dilewati kasus tak seimbang 4x3 algoritma Kruskal

Edge dan biaya yang dilewati kasus tak seimbang 4x3 algoritma Prim

Metode yang terpilih dengan perhitungan manual



Data untuk kasus 2 data transportasi seimbang dengan jumlah sumber lebih banyak dari jumlah tujuan diperoleh dari *Journal Business and Management Research* dengan judul “Modified VOGEL Method to Find Initial Basic Feasible Solution (IBFS) Introducing a New Methodology to Find Best IBFS” karya dari Dr. Muwaffaq Al Kubaisi [10]. Adapun model transportasinya sebagai berikut :

**Tabel 1.** Data seimbang masalah transportasi kasus 2

Dari/Ke	Tujuan				Persediaan
	$D_1$	$D_2$	$D_3$	$D_4$	
$S_1$	9	9	5	7	200
$S_2$	13	9	12	8	300
Sumber $S_3$	4	15	7	9	200
$S_4$	14	7	4	2	400
$S_5$	3	7	3	1	400
Permintaan	500	600	100	300	1500

Setelah mengaplikasikan kedua metode, yaitu pendekatan MST menggunakan algoritma Kruskal dan algoritma Prim pada masalah transportasi dengan perhitungan manual kemudian diuji menggunakan aplikasi TORA, didapat hasil yang tersaji pada tabel sebagai berikut :

**Tabel 2.** Analisis Perhitungan Manual Studi Kasus 2 Data Seimbang

Kriteria	Masalah Transportasi	
	Algoritma Kruskal	Algoritma Prim
Solusi Optimal	9.200	7.800
Banyaknya Iterasi	8 Iterasi	8 Iterasi

Adapun hasil dengan menggunakan aplikasi TORA pada studi kasus 2 memiliki hasil sebesar 7.800. Terlihat pada tabel 2 pendekatan MST dengan algoritma Prim memiliki hasil sama dengan menggunakan aplikasi TORA. Sedangkan pendekatan MST dengan algoritma Kruskal memiliki selisih lebih banyak sebesar 1.500 dari hasil menggunakan aplikasi TORA. Begitupun dengan banyaknya iterasi yaitu 8 iterasi pada semua metode untuk studi kasus 2.

Sedangkan hasil perbandingan perhitungan manual untuk 4 studi kasus yang penulis teliti tampak seperti pada tabel 3 berikut ini.

**Tabel 3.** Analisis Hasil Perbandingan Perhitungan Manual Pendekatan MST Menggunakan Algoritma Kruskal dan Algoritma Prim

Studi Kasus	Masalah Transportasi			
	Algoritma Kruskal		Algoritma Prim	
	Solusi Optimal	Iterasi	Solusi Optimal	Iterasi
Studi Kasus 1	9.300	7 Iterasi	9.500	7 Iterasi
Studi Kasus 2	9.200	8 Iterasi	7.800	8 Iterasi
Studi Kasus 3	439	7 Iterasi	311	7 Iterasi
Studi Kasus 4	1.010	7 Iterasi	750	7 Iterasi

Dari hasil perbandingan 4 studi kasus yang telah penulis kaji. Terlihat kelebihan dan kelemahan metode penyelesaian masalah transportasi menggunakan pendekatan *Minimum Spanning Tree* (MST) dengan algoritma Kruskal dan algoritma Prim. Kelebihan algoritma Kruskal dibanding algoritma Prim adalah algoritma Kruskal sangat cocok diterapkan saat graf memiliki sisi (*edge*) berjumlah sedikit tetapi memiliki banyak simpul (*node*), karena orientasi kerja algoritma Kruskal adalah berdasarkan pada urutan bobot sisi, tidak berdasarkan simpul (*node*).

Sedangkan kelemahan algoritma Kruskal dibanding algoritma lain yaitu Prim adalah algoritma Kruskal kurang cocok digunakan saat graf lengkap atau yang mendekati lengkap, dimana setiap simpul (*node*) terhubung dengan semua simpul (*node*) yang lain. Karena algoritma Kruskal menitik beratkan pada pencarian sisi (*edge*), dimana sisi-sisi tersebut harus diurutkan dari terkecil hingga terbesar dan ini memakan waktu yang cukup lama. Kelemahan pada studi kasus yang telah diteliti, yaitu kurang memberikan solusi yang optimal pada masalah transportasi.

Terlihat kelebihan dan kelemahan metode penyelesaian masalah transportasi dengan pendekatan *Minimum Spanning Tree* (MST) dengan algoritma Prim. Kelebihan algoritma Prim dibanding algoritma Kruskal adalah algoritma Prim sangat cocok diterapkan saat graf memiliki banyak sisi (*edge*) karena banyaknya sisi hampir tidak diperhitungkan di sini, karena algoritma Prim lebih berorientasi kepada pencarian simpul (*node*). Kelebihan pada kasus yang telah diteliti, yaitu memberikan solusi yang optimal pada masalah transportasi

Sedangkan kelemahan algoritma Prim tidak ada pengecekan secara eksplisit apakah sisi yang dipilih akan membentuk siklus atau tidak. Karena pada algoritma Prim sisi yang dimasukkan ke dalam *tree* harus bersisian dengan simpul (*node*) di *tree*. Solusinya dengan melihat apakah simpul (*node*) dari sisi (*edge*) tersebut sudah ada dalam *tree* atau belum. Jika sudah ada maka tidak boleh memilih sisi (*edge*) tersebut karena pasti akan membentuk siklus.

## Kesimpulan

Setelah melakukan analisis data pada 4 kasus yang diambil sebagai objek penelitian dalam penelitian ini penulis dapat menarik kesimpulan sebagai berikut :

- 1) Solusi optimal masalah transportasi menggunakan pendekatan *Minimum Spanning Tree* (MST) dengan algoritma Kruskal. Untuk menentukan solusi optimal dengan pendekatan *Minimum Spanning Tree* (MST) terlebih dahulu membuat graf dengan mengubah data dari bentuk tabel ke bentuk graf. Pada kasus data tidak seimbang harus diseimbangkan dengan melakukan penambahan *dummy*. Kemudian urutkan biaya dari yang terkecil hingga terbesar. Pilih biaya yang paling terkecil kemudian lakukan perhitungan dengan melihat sumber persediaan dan permintaan di setiap tujuan. Lakukan hal tersebut sampai semuanya terpenuhi, sehingga terlihat bentuk *Minimum Spanning Tree* (MST) dari algoritma Kruskal. Maka didapat solusi optimal  $\text{Min } Z = \text{satuan mata uang}$ . Sehingga, dapat dikatakan solusi yang didapat dengan pendekatan MST menggunakan algoritma Kruskal adalah optimal.
- 2) Solusi optimal masalah transportasi menggunakan pendekatan *Minimum Spanning Tree* (MST) dengan algoritma Prim. Untuk menentukan solusi optimal dengan pendekatan *Minimum Spanning Tree* (MST) terlebih dahulu membuat graf dengan mengubah data dari bentuk tabel ke bentuk graf. Pada kasus data tidak seimbang harus diseimbangkan dengan melakukan penambahan *dummy*. Pilih sembarang titik atau sumber, kemudian pilih *active edge* dengan biaya terkecil. Lakukan perhitungan dengan melihat sumber persediaan dan

permintaan di setiap tujuan. Lakukan hal tersebut sampai semuanya terpenuhi, sehingga terlihat bentuk *Minimum Spanning Tree* (MST) dari algoritma Prim. Maka didapat solusi optimal  $\text{Min } Z = \text{satuan mata uang}$ . Sehingga, dapat dikatakan solusi yang didapat dengan pendekatan MST menggunakan algoritma Prim adalah optimal.

- 3) Perbandingan metode penyelesaian masalah transportasi dengan pendekatan MST menggunakan algoritma Kruskal dan algoritma Prim. Setelah membandingkan metode pendekatan MST menggunakan algoritma Kruskal dan algoritma Prim, penulis dapat menarik kesimpulan bahwa metode yang baik untuk menentukan solusi optimal masalah transportasi pada kasus yang telah diteliti adalah pendekatan *Minimum Spanning Tree* (MST) menggunakan algoritma Prim dimana dari 4 studi kasus, 3 diantaranya pada studi kasus 2, 3, dan 4 menghasilkan solusi yang paling optimal dibandingkan dengan menggunakan algoritma Kruskal dengan jumlah iterasi yang sama. Pengecekan solusi optimal menggunakan aplikasi TORA. TORA merupakan perangkat lunak yang memuat beberapa modul untuk menyelesaikan permasalahan optimasi salah satunya transportasi model. Ketika diuji dengan aplikasi TORA, tetap solusi optimal dengan algoritma Prim yang lebih unggul. Sedangkan penyelesaian menggunakan algoritma Kruskal cenderung memberikan solusi optimal dengan selisih yang jauh dari solusi pada algoritma Prim.

## Referensi

- [1] Akpan and Iwok, "A Minimum Spanning Tree Approach of Solving a Transportation Problem," *International Journal of Mathematics and Statistics Invention (IJMSI)*, vol. 5, no. 3, pp. 08-17, 2017.
- [2] S. Johannes, *Riset Operasi untuk pengambilan keputusan*, Jakarta: Salemba empat, 1988.
- [3] J. J. Siang, *Riset Operasi dalam Pendekatan Algoritmis*, Yogyakarta: C.V Andi Offset, 2011.
- [4] R. Munir, *Matematika Diskrit edisi ketiga revisi keempat*, Bandung, 2010.
- [5] J. Erickson, *Algorithms, USA: Department of Computer Science University of Illinois Urbana-Champaign*, 2013.
- [6] Y. S. Kusumah, *Matematika Diskrit*, Bandung: CV. Andira, 1998.
- [7] J. J. Siang, *Matematika Diskrit dan Aplikasinya Pada Ilmu Komputer*, Yogyakarta: C.V Andi, 2002.
- [8] N. Girmay and T. Shaima, "Balance An Unbalanced Transportation Problem By A Heuristic Approach," *International Journal of Mathematics and Its Applications (IJMAA)*, vol. 1, pp. 12-18, 2013.
- [9] K. Aljanabi and A. Jasim, "An Approach for Solving Transportation Problem Using Modified Kruskal's Algorithm," *International Journal of Science and Research (IJSR)*, vol. 4, no. 7, 2015.
- [10] D. M. Alkubaisi, "Modified VOGEL Method to Find Initial Basic Feasible Solution (IBFS) Introduction a New Methodology to Find Best IBFS," *Business and Management Research*, vol. 4, no. 2, 2015.
- [11] A. Quddos, S. Javaid and M. Khalid, "A Revised Version of ASM-Method For Solving Transportation Problem," *International Journal of Agricultural and Statistical Sciences*, vol. 12, pp. 267-272, 2016.