Application of Spanning Tree in Batam District for Designing Cost-Effective Communication Networks

Bryan Ho - 13523029[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523029@std.stei.itb.ac.id, [2]bryanho67@gmail.com

***Abstract*— Batam, the largest city in the Riau Archipelago is considered one of Indonesia's key industrial centers. Batam plays a significant role in Indonesia's economic development due to its infrastructure, such as ports, industrial parks, and growing residential areas. To attain connectivity across various districts in Batam, a cost-effective communication networks design is needed. This paper proposes a solution to design the most cost-effective communication networks by using the minimum spanning tree. The minimum spanning tree is implemented with Kruskal's algorithm.**

***Keywords*— Communication Networks, Cost-Effective, Kruskal's Algorithm, Minimum Spanning Tree.**

I. Introduction

Batam is the largest city in the Riau Archipelago, Indonesia. Being the largest city in the Riau Archipelago, Batam is an industrial city with a large population. The district plays a major role for both tourism and manufacturing. Communication systems are essential for ensuring connectivity between various regions of the district, facilitating business operations, improving tourism experiences, and supporting the daily lives of residents.

This paper proposes a solution using Kruskal's algorithm to identify the minimum spanning tree (MST) for connecting various districts in Batam. MST ensures that all vertices (representing district) are connected with the least total cable length while avoiding redundant connections. Using this method, the total expenses on the construction and maintenance of the communication network can be reduced.

The concept that is used in this paper can be applied not only for Batam, but also in any places around the world. Hopefully, this paper can be used as a guide to design the most cost-effective communication networks.

II. Theoretical Basis

A. Graph

Graph is a set of vertices (also called nodes) that are connected (by edges). Graph can be used to represent discrete objects and their relation. One can formally define graph with the notation $G = (V, E)$, which in this case, V is a set of finite vertices ($\{V_1, V_2, \ldots, V_n\}$) and E is a set of finite edges ($\{e_1, e_2, \ldots, e_n\}$).

Based on the existence of multiple edges and loops, graphs are classified into two types:

1. Simple Graph
   A graph that doesn't contain multiple edges or loops is called simple graph.
2. Unsimple-Graph
   A graph that contains either multiple edges or loops is called unsimple-graph. Unsimple-graph can further be divided into:
   - Multi-Graph
     A graph that contains multiple edges is called multi-graph. Note that multi-graph is not allowed to have any loops.
   - Pseudo-Graph
     A graph that contains loops is called pseudo-graph. Note that pseudo-graph is allowed to have multiple edges.

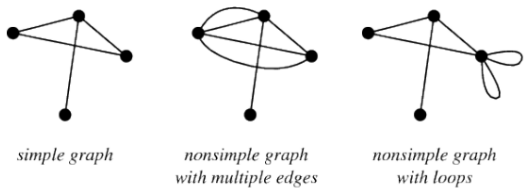


Fig 2.1 Graph based on the existence of multiple edges and loops
Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf

Based on the orientation of the edges, graphs are classified into two types:

1. Undirected Graph
   A graph in which the edges have no directional orientation is called undirected graph.
2. Directed Graph
   A graph in which each edge has a directional orientation is called directed graph.

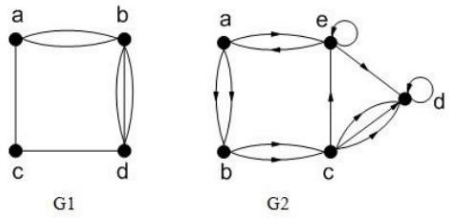


Fig 2.2 (G1) undirected graph, (G2) directed graph
Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf

There are several basic graph terminologies that are used in this paper. The terminology consists of:

1. Adjacent
   Two vertices in a graph are said to be adjacent if both are directly connected by an edge.
2. Incident
   An edge E is said to be incident to vertice V_j and vertice V_k if there are any $E = (V_j, V_k)$. In this case V_j and V_k are the endpoints of E.
3. Isolated Vertex
   An isolated vertex is a vertex that has no edges.
4. Null Graph
   Null Graph (also called empty graph) is a graph which has an empty set of edges.
5. Degree
   The degree of a vertice is the number of edges incident to that vertice.
6. Path
   Path is a sequence of vertices where each adjacent pair is connected by an edge.
7. Cycle or Circuit
   Cycle or circuit is a path that starts and ends at the same vertice.
8. Connected
   Two vertices are said to be connected if there is a path that connects both vertices.
9. Subgraph and Complement of a Subgraph
   Let there be a graph $G = (V, E)$ and $G_1 = (V_1, E_1)$. G_1 is said to be subgraph of G, if $V_1 \subseteq V$ and $E_1 \subseteq E$. The complement of subgraph G_1 to graph G is $G_2 = (V_2, E_2)$ such that $E_2 = E - E_1$ and V_2 is a set of vertices where all sets in E_2 are incident with.
10. Spanning Subgraph
    Let there be a graph $G = (V, E)$ and $G_1 = (V_1, E_1)$. G_1 is said to be spanning subgraph of G, if $V_1 = V$. In this case, G_1 contains all the vertices of G.
11. Cut-set
    The cut-set of a connected graph G is the set of edges that makes graph G disconnect when removed.
12. Weighted Graph
    Weighted graph is a graph where each edge is assigned with value.

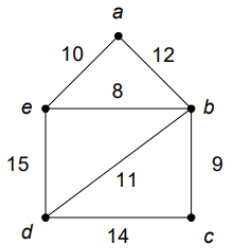
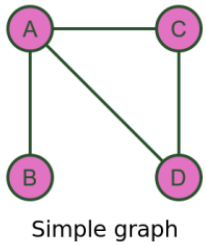


Fig 2.3 Weighted graph
Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf

There are multiple ways to represent a graph. In this paper, the author will only cover the method of representing a graph using an adjacency matrix.

1. Adjacency matrix
   Adjacency matrix is a square matrix of N size where N is the number of nodes in the graph and it is used to represent the connections between the edges of a graph.

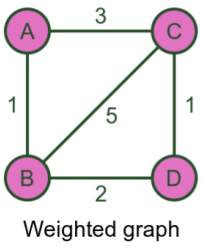
   For unweighted graph, the values are 0 and 1 where 0 indicates no edge and 1 indicates an edge.



	A	B	C	D
A	0	1	1	1
B	1	0	0	0
C	1	0	0	1
D	1	0	1	0

Fig 2.4 Adjacency matrix of a simple graph
Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf

For weighted graph, the values depend on the weight of each edge.



	A	B	C	D
A	0	1	3	0
B	1	0	5	2
C	3	5	0	1
D	0	2	1	0

Fig 2.5 Adjacency matrix of a weighted graph
Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf

B. Tree

Tree is an undirected graph that is connected and doesn't contain cycle or circuit. Based on the definition of tree, there are three conditions for a tree:

1. The graph must be undirected.
2. The graph must be connected.
3. The graph must not contain circuits.

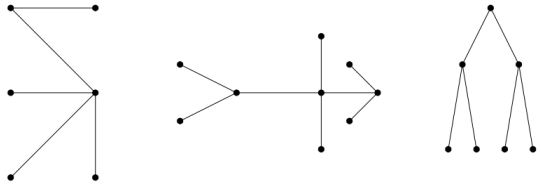


Fig 2.6 Tree

Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf

Let there be an undirected simple graph $G = (V, E)$ with n number of vertices, then all these statements are equivalent:

1. G is a tree.
2. Each pair of vertices in G is connected by a single path.
3. G is connected and has n-1 edges.
4. G doesn't contain circuits and has n-1 edges.
5. G doesn't contain circuits and adding one edge to the graph will form only one circuit.
6. G is connected and all its edges are bridges.

Spanning tree of a connected graph is a spanning subgraph in the form of tree. A spanning tree is obtained by removing the circuits in the graph. Every connected graph has at least one spanning tree.

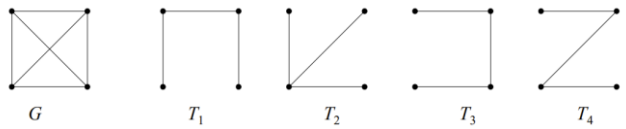


Fig 2.7 Connected graph (G) with its spanning trees (T_1, T_2, T_3, T_4)

Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf

Connected-weighted graph may have more than one spanning tree. Spanning tree which has minimum weight is called minimum spanning tree.

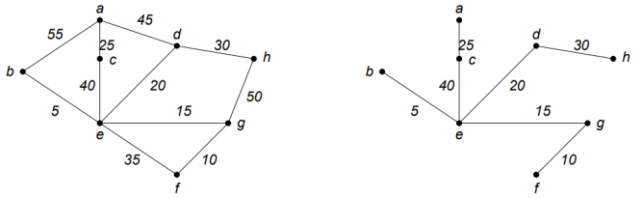


Fig 2.8 Connected-weighted graph and its minimum spanning tree

Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf

C. Kruskal's Algorithm

There are several ways to find the minimum spanning tree. In this paper, the author will use Kruskal's algorithm. Kruskal's algorithm is a minimum spanning tree algorithm that takes a graph as input and finds the subset of the edges of that graph which forms a tree that includes every vertices and has the minimum sum of weights among all the trees that can be formed from the graph.

These are three steps to implement Kruskal's algorithm.

1. Sort all edges ascendingly based on the weight.
2. Take the edge with the lowest weight and add it to the spanning tree. If adding the edge creates a circuit, reject this edge.
3. Keep adding edges until all vertices are included in the spanning tree.

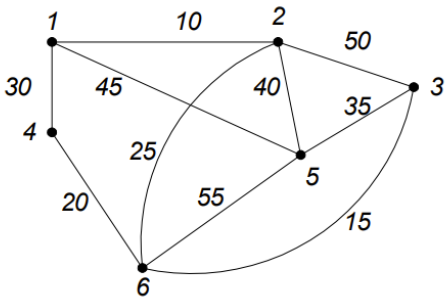


Fig 2.9 Connected-weighted Graph

Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf

Sisi-sisi diurut menaik:

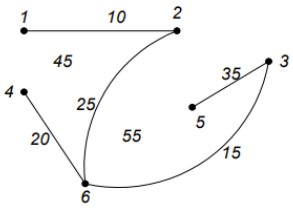
Sisi	(1,2)	(3,6)	(4,6)	(2,6)	(1,4)	(3,5)	(2,5)	(1,5)	(2,3)	(5,6)
Bobot	10	15	20	25	30	35	40	45	50	55

Langkah	Sisi	Bobot	Hutan merentang
0			
1	(1, 2)	10	
2	(3, 6)	15	
3	(4, 6)	20	
4	(2, 6)	25	
5	(1, 4)	30	ditolak
6	(3, 5)	35	

Fig 2.10 Kruskal's algorithm

Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf

Pohon merentang minimum yang dihasilkan:



Bobot = 10 + 25 + 15 + 20 + 35 = 105

Fig 2.11 Minimum spanning tree
Source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf

III. IMPLEMENTATION

A. Research Limitation

On writing this paper, the author uses some limitations to determine the most cost-effective design for communication networks Batam City through its district using spanning tree. The limitations are:

1. Connection distance is measured only by its distance. Factors like traffic, road condition, and cost between different terrain are ignored.
2. If there are many districts that can be connected from one district to another, connect to the other district with the shortest distance without forming a circuit.
3. If there is more than one district that has the same connection distance, free to choose where to connect without forming a circuit.
4. The starting point is the district with the shortest connection distance.

B. Research Site

The location of research that is used in this paper is the district in Batam, which consists of:

1. Batu Ampar
2. Lubuk Baja
3. Bengkong
4. Sekupang
5. Batam Kota
6. Nongsa
7. Batu Aji
8. Sagulung
9. Sei Beduk

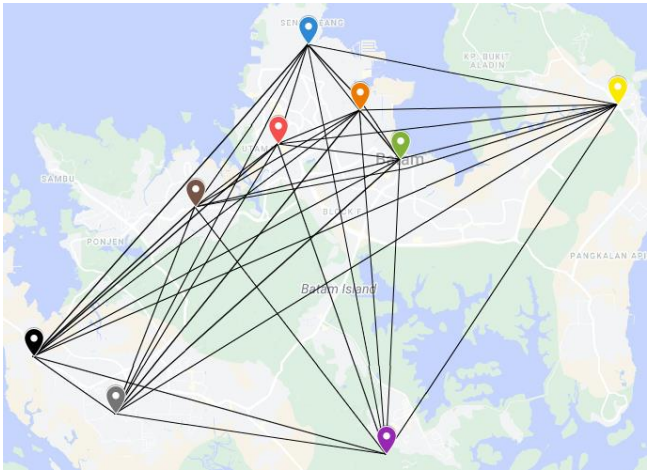


Fig 3.1 Batam district
Source: Google Maps

where,

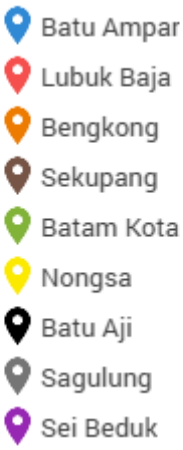


Fig 3.2 Detail of Batam district for each pointer
Source: Google Maps

The relation of each district on the research site is represented by using adjacency matrix. It shows a weighted graph where each vertice represents Batam district and each edge represents the connection distance between each district in kilometers (km).

Table I Adjacency matrix of Batam district

District	1	2	3	4	5	6	7	8	9
1	∞	4.2	3.4	8.1	6	12.8	16.9	16.9	17
2	4.2	∞	3.6	4.2	5	13.9	13.2	12.8	13.4
3	3.4	3.6	∞	7.8	2.6	10.4	16.6	15.9	14
4	8.1	4.2	7.8	∞	8.6	17.7	8.9	9	12.7
5	6	5	2.6	8.6	∞	9.1	16.9	15.5	12
6	12.8	13.9	10.4	17.7	9.1	∞	25.8	23.9	17
7	16.9	13.2	16.6	8.9	16.9	25.8	∞	4.1	14.9

8	16.9	12.8	15.9	9	15.5	23.9	4.1	∞	11.1
9	17	13.4	14	12.7	12	17	14.9	11.1	∞

C. Application of Spanning Tree Using Kruskal's Algorithm to Determine the Shortest Distance

The first step is to sort each edge ascendingly based on the connection distance.

Table II List of sorted edges based on the weight

Edge	(3,5)	(1,3)	(2,3)	(7,8)	(1,2)	(2,4)
Distance	2.6	3.4	3.6	4.1	4.2	4.2

Edge	(2,5)	(1,5)	(3,4)	(1,4)	(4,5)	(4,7)
Distance	5	6	7.8	8.1	8.6	8.9

Edge	(4,8)	(5,6)	(3,6)	(8,9)	(5,9)	(4,9)
Distance	9	9.1	10.4	11.1	12	12.7

Edge	(1,6)	(2,8)	(2,7)	(2.9)	(2,6)	(3.9)
Distance	12.8	12.8	13.2	13.4	13.9	14

Edge	(7,9)	(5,8)	(3,8)	(3,7)	(1,7)	(1,8)
Distance	14.9	15.5	15.9	16.6	16.9	16.9

Edge	(5,7)	(1,9)	(6,9)	(4,6)	(6,8)	(6,7)
Distance	16.9	17	17	17.7	23.9	25.8

After sorting all the connection distance, the next step is to connect all adjacent vertices. Choose the edge between *u* and *v* with the minimum weight that doesn't form a circuit. Include this edge and repeat this step until there are (*n*-1) edges in the spanning tree, where *n* indicates the number of vertices. In this case, *n* equals to 9.

Step 1: Choose edge (3,5). No circuit is formed, include it.

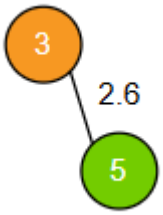


Fig 3.3 Step 1 add edge (3,5) into tree

Step 2: Choose edge (1,3). No circuit is formed, include it.

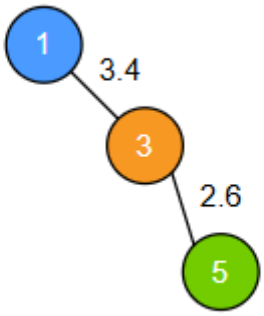


Fig 3.4 Step 2 add edge (1,3) into tree

Step 3: Choose edge (2,3). No circuit is formed, include it.

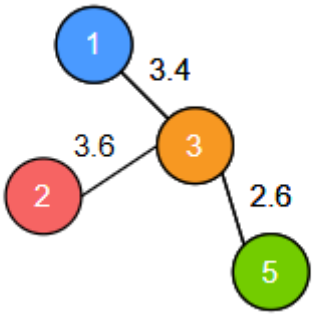


Fig 3.5 Step 3 add edge (2,3) into tree

Step 4: Choose edge (7,8). No circuit is formed, include it.

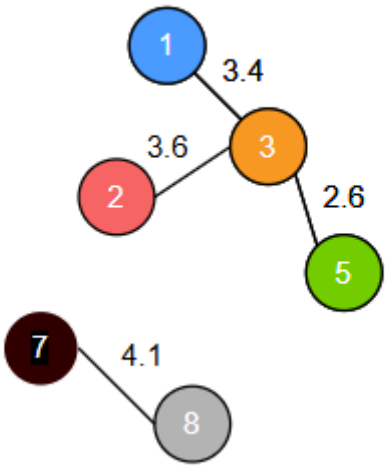


Fig 3.6 Step 4 add edge (7,8) into tree

Step 5: Choose edge (1,2). Circuit is formed, discard it. Choose edge (2,4). No circuit is formed, include it.

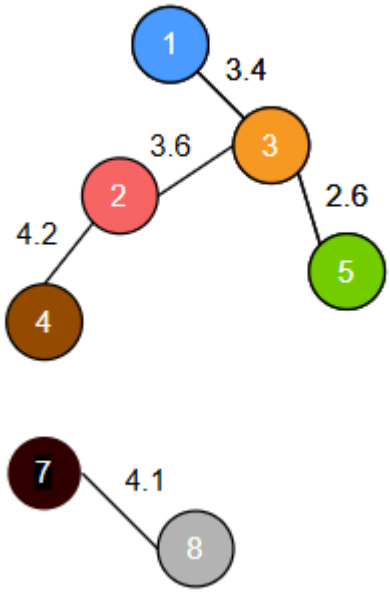


Fig 3.7 Step 5 add edge (2,4) into tree

Step 6: Choose edge (2,5). Circuit is formed, discard it. Choose edge (1,5). Circuit is formed, discard it. Choose edge (3,4). Circuit is formed, discard it. Choose edge (1,4). Circuit is formed, discard it. Choose edge (4,5). Circuit is formed, discard it. Choose edge (4,7). No circuit is formed, include it.

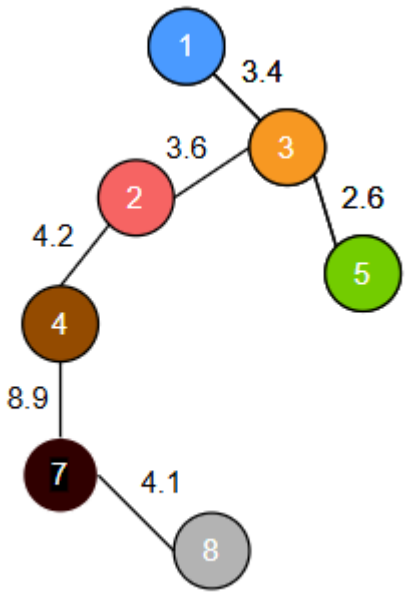


Fig 3.8 Step 6 add edge (4,7) into tree

Step 7: Choose edge (4,8). Circuit is formed, discard it. Choose edge (5,6). No circuit is formed, include it.

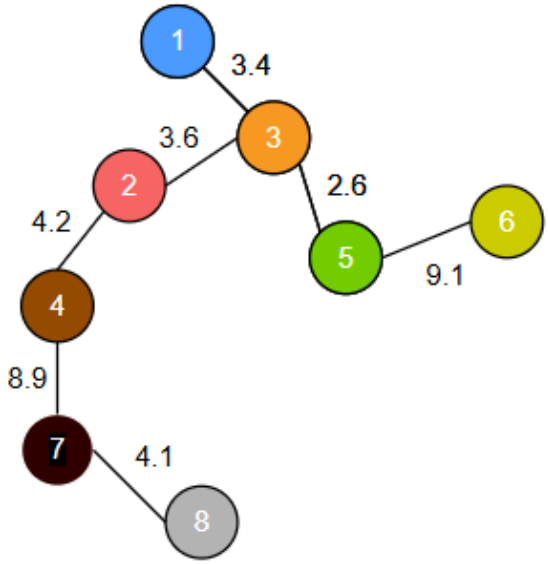


Fig 3.9 Step 7 add edge (5,6) into tree

Step 8: Choose edge (3,6). Circuit is formed, discard it. Choose edge (8,9). No circuit is formed, include it.

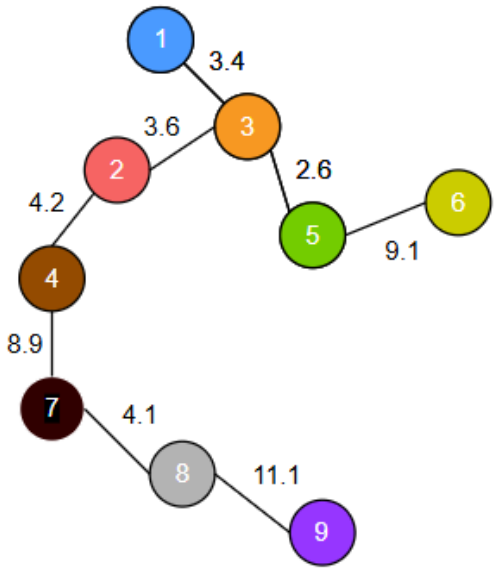


Fig 3.10 Step 8 add edge (8,9) into tree and minimum spanning tree is obtained

Since the number of edges included in the spanning tree is equal to (n-1) which indicates all vertices have been included in the spanning tree, the algorithm stops here.

The total weight of the spanning tree can be obtained by summing all values of each edge. In this case, the sum would be *2.6 + 3.4 + 3.6 + 4.1 + 4.2 + 8.9 + 9.1 + 11.1 = 47*.

IV. Result

Based on the calculation result using Kruskal's algorithm to determine the most cost-effective design for communication networks Batam City through its district, the shortest distance obtained for this problem is 47 kilometers.

V. Conclusion

In conclusion, determining the shortest distance to design the most cost-effective communication networks in Batam can be done by using spanning tree approach with the help of Kruskal's algorithm. By using Kruskal's algorithm, the shortest distance obtained is 47 kilometers.

The distance obtained can be used to determine the most optimal communication network which is seen from the minimum total length of cables or infrastructure needed to reduce construction and maintenance costs. The cost of installation and maintenance is directly proportional to the distance between nodes.

VI. Appendix

Video explanation of this paper: https://youtu.be/H9bXpcri2xQ

VII. Acknowledgment

This paper was completed by the grace, guidance, and protection of God which enabled the author to finish it on time. The author would like to extend heartfelt gratitude to all parties who supported the study and learning process in IF1220 Discrete Mathematics course, especially to Bapak Dr. Rinaldi Munir, Bapak Dr. Rila Mandala, and Bapak Arrival Dwi Sentosa, M.T as the course lecturers for their invaluable guidance. Lastly, the author apologizes for any shortcomings that may remain in this paper. It is sincerely hoped that this paper can serve as a useful reference for future studies or research purposes.

References

[1] Munir, Rinaldi. 2024. "Graf (Bagian 1)". https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf (accessed on 25 December 2024).

[2] Munir, Rinaldi. 2024. "Graf (Bagian 2)". https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf (accessed on 25 December 2024).

[3] https://math.gordon.edu/courses/mat230/handouts/graphs.pdf (accessed on 25 December 2024).

[4] https://www.geeksforgeeks.org/graph-terminology-in-data-structure/ (accessed on 25 December 2024).

[5] Munir, Rinaldi. 2024. "Pohon (Bagian 1)". https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf (accessed on 26 December 2024).

[6] https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/ (accessed on 26 December 2024).

[7] https://www.programiz.com/dsa/kruskal-algorithm (accessed on 26 December 2024).

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 31 Desember 2024

Bryan Ho (13523029)