PERBANDINGAN ALGORITMA *ENHANCED* SOLLIN DAN ALGORITMA *ENHANCED* KRUSKAL UNTUK MENYELESAIKAN *MINIMUM ROUTING-COST SPANNING TREE (MRCST) PROBLEM*

(Tesis)

Oleh

ASTRI REFORMASARI
2227031005

PROGRAM STUDI MAGISTER MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

ABSTRACT

THE COMPARISON OF *ENHANCED* SOLLIN ALGORITHM AND *ENHANCED* KRUSKAL ALGORITHM TO SOLVE THE MINIMUM ROUTING-COST SPANNING TREE (MRCST) PROBLEM

By

ASTRI REFORMASARI

A spanning tree that minimises the total pairwise distances between its vertices is the Minimum Routing Cost-Spanning Tree (MRCST) of a weighted graph. MRCST can be determined using several algorithms such as Sollin and Kruskal. In this study, the development (enhanced) of the Sollin and Kruskal algorithms was discussed to see whether the resulting MRCST can be more optimal and to see its effect and compare the results of the modification of the two algorithms. The data used was randomly generated data using a uniform distribution and the value was an integer. The data if represented in a graph, was a complete graph data that had a number of points of 10, 20, 30 to 100 with each number of nodes having 30 different problems. The results of the implementation using a python program show that the difference in MRCST values between the enhanced Sollin and enhanced Kruskal algorithms, compared with the original Sollin and Kruskal algorithms, tends to increase as graph order increases. For graph orders 40, 50, and 70–100, the enhanced Sollin algorithm produced lower MRCST values than the enhanced Kruskal algorithm. In contrast, the enhanced Kruskal algorithm performed better for graph orders 10–30 and 60.

Keywords: *MRCST, Sollin's Algorithm, Kruskal's Algorithm, Enhanced Sollin's Algorithm, Enhanced Kruskal's Algorithm.*

ABSTRAK

PERBANDINGAN ALGORITMA *ENHANCED* SOLLIN DAN ALGORITMA *ENHANCED* KRUSKAL UNTUK MENYELESAIKAN *MINIMUM ROUTING-COST SPANNING TREE (MRCST) PROBLEM*

Oleh

ASTRI REFORMASARI

Pohon merentang yang meminimumkan jumlah jarak berpasangan antara seluruh pasangan titiknya disebut sebagai *Minimum Routing-Cost Spanning Tree (MRCST)* dari suatu graf berbobot. MRCST dapat ditentukan dengan menggunakan beberapa algoritma seperti Sollin dan Kruskal. Pada penelitian ini, akan dibahas pengembangan (*enhanced)* algoritma Sollin dan Kruskal untuk melihat apakah MRCST yang dihasilkan dapat lebih optimal serta membandingkan hasil pengembangan kedua algoritma tersebut. Data dibangkitkan secara acak menggunakan distribusi *uniform* dan nilainya berupa bilangan bulat (*integer*). Data tersebut jika direpresentasikan dalam graf merupakan data graf lengkap yang memiliki banyaknya titik (*orde*) 10, 20, 30 sampai 100 dengan setiap *orde* memiliki 30 masalah berbeda. Hasil implementasi menggunakan program komputasi *Python* menunjukkan bahwa selisih nilai MRCST antara algoritma *enhanced* Sollin dan algoritma *enhanced* kruskal dengan algoritma Sollin dan algoritma Kruskal cenderung meningkat seiring bertambahnya *orde* graf. Pada *orde* 40, 50, dan 70 s.d. 100, hasil algoritma *enhanced* Sollin lebih unggul dibandingkan algoritma *enhanced* Kruskal sedangkan algoritma *enhanced* Kruskal unggul pada orde 10 s.d. 30 dan 60.

Kata Kunci: MRCST, Algoritma Sollin, Algoritma Kruskal, Algoritma *Enhanced* Sollin, Algoritma *Enhanced* Kruskal.

PERBANDINGAN ALGORITMA *ENHANCED* SOLLIN DAN ALGORITMA *ENHANCED* KRUSKAL UNTUK MENYELESAIKAN *MINIMUM ROUTING-COST SPANNING TREE (MRCST) PROBLEM*

Oleh

ASTRI REFORMASARI

(Tesis)

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
MAGISTER MATEMATIKA

Pada

Program Studi Magister Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

PROGRAM STUDI MAGISTER MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

Judul : **PERBANDINGAN ALGORITMA *ENHANCED* SOLLIN DAN ALGORITMA *ENHANCED* KRUSKAL UNTUK MENYELESAIKAN *MINIMUM ROUTING-COST SPANNING TREE (MRCST) PROBLEM***

Nama Mahasiswa : Astri Reformasari

Nomor pokok mahasiswa : 2227031005

Jurusan : Matematika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Prof. Dra. Wamiliana, M.A., Ph.D.	**Dr. Fitriani, S.Si., M.Sc.**
NIP. 19631108 198902 2 001	**NIP. 19840627 200604 2 001**

2. Koordinator Program Studi Magister Matematika

Dr. Fitriani, S.Si., M.Sc
NIP. 19840627 200604 2 001

MENGESAHKAN

1. Tim Penguji

Ketua	: **Prof. Dra. Wamiliana, M.A., Ph.D.**	....
Sekretaris	: **Dr. Fitriani, S.Si., M.Sc.**	....
Penguji		
Bukan Pembimbing	: **1. Prof. Dr. Asmiati, S.Si., M.Si.**	....
	2. Dr. Muslim Ansori, S.Si., M.Si.	....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 19711001 200501 1 002

3. Direktur Program Pascasarjana

Prof. Dr. Ir. Muhadi, M.Si.
NIP. 19640326 199802 1 001

4. Tanggal Lulus Ujian Tesis: 15 Juni 2026

PERNYATAAN TESIS MAHASISWA

Yang bertandatangan di bawah ini :

Nama	: **Astri Reformasari**
Nomor Pokok Mahasiswa	: **2227031005**
Program Studi	: **Magister Matematika**
Jurusan	: **Matematika**

Dengan ini menyatakan bahwa tesis saya yang berjudul **"PERBANDINGAN ALGORITMA *ENHANCED* SOLLIN DAN ALGORITMA *ENHANCED* KRUSKAL UNTUK MENYELESAIKAN *MINIMUM ROUTING COST SPANNING TREE (MRCST) PROBLEM*"** adalah hasil pekerjaan saya sendiri. Semua hasil tulisan yang tertuang dalam tesis ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila kemudian hari terbukti bahwa tesis ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, 15 Juni 2026

METERAI TEMPEL
10000
A3AB5AOX056273079

Astri Reformasari
NPM. 2227031005

RIWAYAT HIDUP

Penulis dilahirkan di Bandar Lampung pada tanggal 11 September 1998, sebagai anak ketiga dari empat bersaudara pasangan Bapak Didit Bustami dan Ibu Otik Nawansih.

Penulis telah menempuh Pendidikan di Sekolah Dasar (SD) Al-Azhar 2 tahun 2004-2010, Sekolah Menengah Pertama Negeri (SMPN) 10 Bandar Lampung tahun 2010-2013, Sekolah Menengah Atas Negeri (SMAN) 9 Bandar Lampung pada tahun 2013-2016, dan Strata Satu (S1) Program Studi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam (MIPA) Universitas Lampung (Unila) pada tahun 2016-2021.

Pada tahun 2022, penulis melanjutkan Pendidikan Strata Dua (S2) Program Studi Magister Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam (MIPA) Universitas Lampung (Unila)

Selama menjadi mahasiswi, penulis bekerja sebagai *Membership Staff* Divisi Kepesertaan di PT TASPEN (Persero) Kantor Cabang Surabaya dan aktif menjalankan usaha sebagai Pengawas di Yayasan Bimbingan Belajar Fokus yang berpusat di Bandar Lampung.

KATA INSPIRASI

"Janganlah engkau bersedih, sesungguhnya Allah Bersama kita"

(QS. At-Taubah : 40)

"Tidak semua yang kita inginkan terjadi. Ini bukan dunia dongeng"

(Tere Liye)

"Seekor burung yang duduk diatas pohon tidak pernah takut rantingnya patah, karena kepercayaannya bukan pada cabang dahannya, tetapi pada kemampuannya untuk terbang"

(Jalaluddin Rumi)

"Jika tulisanmu jelek, maka carilah orang yang bisa membacanya"

(Anonim)

PERSEMBAHAN

Dengan mengucap puji dan syukur kehadirat Allah SWT
yang telah memberikan petunjuk dan kemudahan untuk menyelesaikan tesis ini.
Kupersembahkan karya kecil dan sederhana ini kepada:

Suami dan Orangtua Tersayang

Sebagai tanda bakti, hormat, dan rasa terima kasih kepada suami, papa, mamak, Bapak dan khususnya Alm. Mama yang tidak penah lelah untuk selalu mendoakan, memberikan dukungan, kasih sayang yang tidak mungkin terbalas dengan selembar kata cinta dan persembahan ini. Semoga ini menjadi langkah awal untuk menjadi pribadi yang lebih baik.

Anak Tersayang

Yang telah memberikan semangat dan motivasi untuk terus tumbuh menjadi orangtua yang lebih baik, meskipun belum ada kata utuh yang terucap darinya.

Kakak dan Adik Tersayang

Yang telah membantu, memberikan semangat, motivasi, dan doa yang tidak akan terbayarkan oleh apapun,

Dosen Pembimbing dan Penguji

Yang senantiasa mengarahkan dan memberi motivasi kepada penulis

Almamater Tercinta, Universitas Lampung

SANWACANA

Alhamdulillah puji syukur kepada Allah SWT yang telah melimpahkan segala rahmat dan hidayah-Nya sehingga tesis ini dapat diselesaikan. Shalawat serta salam semoga selalu tercurah kepada Nabi Muhammad SAW, penuntun jalan bagi seluruh umat manusia. Dalam penulisan tesis ini, penulis banyak mendapatkan bantuan, bimbingan dan dorongan baik itu langsung maupun tidak langsung dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Ibu Prof. Dra. Wamiliana, MA., Ph.D., selaku Pembimbing I yang selalu bersedia memberikan arahan, bimbingan, saran serta dukungan kepada penulis sehingga dapat menyelesaikan tesis ini.
2. Ibu Dr. Fitriani, S.Si., M.Sc., selaku Pembimbing II dan Ketua Prodi Magister Matematika yang telah banyak memberikan pengarahan dan bimbingan, saran, nasihat, motivasi dan kritikan dalam penyusunan tesis.
3. Ibu Prof. Dr. Asmiati, S.Si., M.Si., selaku Penguji I yang telah bersedia memberikan kritik dan masukan serta evaluasi kepada penulis untuk lebih baik kedepannya.

4. Bapak Dr. Muslim Ansori, S.Si., M.Si., selaku Penguji II dan pembimbing akademik yang telah memberikan arahan, bimbingan, saran dan evaluasi yang membangun kepada penulis untuk bisa lebih baik kedepannya.
5. Bapak Dr. Aang Nuryaman, S.Si., M.Si., selaku Ketua Jurusan Matematika.
6. Bapak Dr. Eng. Heri Satria, S.Si., M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
7. Bapak Prof. Dr. Ir. Muhardi, M.Si., selaku Direktur Program Pascasarjana Universitas Lampung.
8. Seluruh dosen dan karyawan jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
9. Suami dan anak tercinta, Feri Irawan dan Flavia Zhyona Irawan yang selalu memberikan kasih sayang, semangat, motivasi, menyertai penulis dalam doanya, serta menemani penulis dalam melaksanakan dan menyelesaikan tesis.
10. Keluarga tercinta (Mama, Papa, Mamak, Bapak, Mas Danang, Mba Laras, Nadia, Kak Eca, Kak Sarwin, Mba Nina, Kak Cahyo, Mba Ike dan Pasirah) yang telah memberikan dukungan, motivasi dan yang selalu menyertai penulis dalam doanya untuk melaksanakan dan menyelesaikan tesis.
11. Sahabat-sahabat serta teman-teman terbaik angkatan 2022 atas pengalaman yang diberikan, semangat, dukungan, canda tawa, serta kebersamaannya selama ini.
12. Keluarga besar PT TASPEN (Persero) dan Yayasan Bimbingan Belajar Fokus yang telah memberikan dukungan dan kesempatan untuk meluangkan waktu serta pikiran dalam menyelesaikan tesis.

13. Seluruh pihak yang telah memotivasi, membantu, dan mendoakan penulis yang tidak dapat disebutkan satu per satu.

Penulis sangat menyadari tesis ini jauh dari kata sempurna. Oleh sebab itu, penulis sangat mengharapkan kritik dan saran yang membangun dan dapat memberikan manfaat bagi penulis pribadi dan bagi para pembaca.

Bandar Lampung, 15 Juni 2026
Penulis,

Astri Reformasari
NPM.2227031005

.

DAFTAR ISI

Halaman

DAFTAR ISI .... **xiv**

DAFTAR TABEL .... **xvi**

DAFTAR GAMBAR .... **xviii**

I. PENDAHULUAN

1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 3
1.4 Manfaat Penelitian 4

II. TINJAUAN PUSTAKA

2.1 Konsep Dasar Graf 5
2.2 Algoritma Sollin 10
2.3 Algoritma Kruskal 14

III. METODE PENELITIAN

3.1 Jenis dan Sumber Data 18
3.2 Waktu dan Tempat Penelitian 18
3.3 Tahapan Penelitian 19
3.4 Alur Penelitian 20

IV. HASIL DAN PEMBAHASAN

4.1 Algoritma Sollin..21

4.2 Algoritma *Enhanced Sollin*..27

4.3 Algoritma Kruskal 37

4.4 Algoritma *Enhanced* Kruskal 41

4.5 Perbandingan Hasil Pengujian Algoritma *Enhanced* Sollin dan *Enhanced* Kruskal..44

V. KESIMPULAN...62

DAFTAR PUSTAKA

DAFTAR TABEL

Tabel Halaman

2.3.1 Contoh Tabel Algoritma Kruskal untuk Penyelesaian Graf pada Gambar 2.2.1 17
4.1.1 Data 11 (Graf Lengkap dengan *Orde* 10) 22
4.1.2 Hasil Pengujian Algoritma Sollin untuk Banyaknya Titik (*Orde*) 10 s.d. 50 25
4.1.3 Hasil Pengujian Algoritma Sollin untuk Banyaknya Titik (*Orde*) 60 s.d. 100 26
4.2.1 Hasil Pengujian Algoritma *Enhanced* Sollin untuk Banyaknya Titik (*Orde*) 10 s.d. 50 34
4.2.2 Hasil Pengujian Algoritma *Enhanced* Sollin untuk Banyaknya Titik (*Orde*) 60 s.d. 100 35
4.3.1 Hasil Pengujian Algoritma Kruskal untuk Banyaknya Titik (*Orde*) 10 s.d. 50 40
4.3.2 Hasil Pengujian Algoritma Kruskal untuk Banyaknya Titik (*Orde*) 60 s.d. 100 41
4.4.1 Hasil Pengujian Algoritma *Enhanced* Kruskal untuk Banyaknya Titik (*Orde*) 10 s.d. 50 47

4.4.2 Hasil Pengujian Algoritma *Enhanced* Kruskal untuk Banyaknya Titik (*Orde*) 60 s.d. 100 48
4.5.1 Hasil Penyelesaian MRCST pada *Orde* (*n*) 10 49
4.5.2 Hasil Penyelesaian MRCST pada *Orde* (*n*) 20 50
4.5.3 Hasil Penyelesaian MRCST pada *Orde* (*n*) 30 52
4.5.4 Hasil Penyelesaian MRCST pada *Orde* (*n*) 40 53
4.5.5 Hasil Penyelesaian MRCST pada *Orde* (*n*) 50 54
4.5.6 Hasil Penyelesaian MRCST pada *Orde* (*n*) 60 55
4.5.7 Hasil Penyelesaian MRCST pada *Orde* (*n*) 70 56
4.5.8 Hasil Penyelesaian MRCST pada *Orde* (*n*) 80 57
4.5.9 Hasil Penyelesaian MRCST pada *Orde* (*n*) 90 58
4.5.10 Hasil Penyelesaian MRCST pada *Orde* (*n*) 100 59
4.5.11 Ringkasan Hasil Penyelesaian MRCST pada *Orde* (*n*) 10 s.d. 50 60
4.5.11 Ringkasan Hasil Penyelesaian MRCST pada *Orde* (*n*) 60 s.d. 100 61

DAFTAR GAMBAR

Gambar Halaman

2.1.1 Contoh Graf G dengan Lima Titik 6

2.1.2 Contoh Graf G dengan Tujuh Titik 7

2.1.3 Contoh Graf Berbobot 9

2.2.1 Contoh Graf G dengan 7 Titik dan 9 Sisi 11

2.2.2 Graf G pada Langkah 2 12

2.2.3 Graf G pada Langkah 3 12

2.2.4 Graf G pada Langkah 5 13

2.2.5 Graf G pada Langkah 6 13

2.2.6 Graf G pada Langkah 7 13

2.3.2 Graf G Hasil Algoritma Kruskal 16

4.1.1 Dua Komponen yang Terbentuk dari Langkah 2 22

4.1.2 Penggabungan Komponen pada Langkah 3 23

4.1.3 Contoh Graf Data 11 (n = 10) Hasil Algoritma Sollin 27

4.2.1 Graf Hasil Algoritma Enhanced Sollin 33

4.2.2 Contoh Graf Data 11 (n = 10) Hasil Algoritma *Enhanced* Sollin 36

4.3.1 Graf Data 11 Hasil Pengerjaan Algoritma Kruskal 39

4.4.1 Graf Hasil Algoritma *Enhanced* Kruskal 46

4.4.1 Contoh Graf Data 11 (n = 10) Hasil Algoritma *Enhanced* Kruskal 49

I. PENDAHULUAN

1.1. Latar Belakang

Perkembangan dan penerapan ilmu pengetahuan khususnya di bidang matematika semakin meningkat seiring berkembangnya teknologi. Ilmu matematika yang awalnya dianggap rumit dan sulit diaplikasikan dalam kehidupan sehari-hari, sekarang menjadi solusi untuk mendeskripsikan model persoalan serta menggambarkannya secara konkret dan jelas. Salah satu ilmu matematika yang sering digunakan adalah teori graf. Pada tahun 1736, Leonhard Euler memberikan kontribusi yang sangat besar terhadap perkembangan ilmu matematika melalui teori graf. Berawal dari permasalahan jembatan Konigsberg yang memiliki tujuh jembatan, penduduk kota tersebut ingin melewati jembatan tersebut tepat satu kali dan kembali lagi ke tempat awal keberangkatan. Dari permasalahan tersebut, Leonard Euler menemukan jawaban dengan cara merepresentasikan ke dalam graf sehingga dapat mempermudah menganalisis dan menemukan solusi dari suatu permasalahan yang membutuhkan dana besar serta waktu lama untuk membuktikannya secara langsung (Rizki, 2012).

Teori graf memiliki objek yang diskrit, dan dapat diterapkan dalam beberapa bidang seperti fisika, kimia, biologi, teknik, ekonomi, dan kesehatan. Graf sering dipergunakan untuk mempermudah menyelesaikan berbagai macam persoalan-persoalan yang sulit diselesaikan dengan perhitungan dan pertimbangan biasa. Graf G didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari titik (*vertex*) dan E adalah himpunan sisi (*edge*) yang menghubungkan sepasang titik.

Suatu graf dimungkinkan tidak mempunyai sisi, tetapi titiknya harus ada, minimal satu (Aryana dan Sangkala, 2023). Apabila graf $G(V, E)$ memuat sisi berbobot disebut sebagai graf berbobot.

Suatu graf dikatakan terhubung apabila untuk setiap titik u dan v terdapat lintasan yang menghubungkannya (Sumardi dkk., 2021). Dalam teori graf terdapat konsep *tree* (pohon) dan *spanning-tree* (pohon merentang). *Tree* diartikan sebagai graf terhubung yang tidak memuat siklus, sedangkan *spanning-tree* adalah sebuah pohon pada graf G yang memuat semua titik di G (Nurdiyanto dan Susanti, 2019).

Permasalahan pada graf yang cukup populer adalah menentukan rute terpendek (*Minimum Spanning Tree*). *Minimum spanning tree* (MST) pada graf berbobot G adalah graf terhubung yang memuat semua titik pada graf G tanpa sirkuit dan memiliki jumlah bobot paling minimum dari semua sisi. MST memiliki banyak kegunaan, seperti mencari jalan minimum untuk menghubungkan beberapa kota dan perutean pesan pada jaringan komputer. Selain MST, terdapat juga *Minimum Routing-Cost Spanning Tree* (MRCST) yang merupakan jarak minimum total antara semua pasangan titik untuk setiap titik pada graf. MST dan MRCST umumnya banyak diaplikasikan dalam masalah desain jaringan.

Meskipun tidak sepopuler permasalahan *Travelling Salesman Problem* (TSP), sudah banyak peneliti yang membahas MRCST diantaranya Hieu dkk. (2011) membahas algoritma semut yang diusulkan untuk menyelesaikan MRCST, Tan (2012a) yang mengusulkan tiga heuristik untuk menyelesaikan MRCST, dan Tan (2012b) yang mengusulkan Algoritma genetika untuk menyelesaikan MRCST.

MRCST dapat ditentukan dengan menggunakan beberapa algoritma seperti Sollin dan Kruskal. Algoritma Sollin melakukan pencarian MRCST dengan memilih sisi yang memiliki bobot terkecil disetiap titik kemudian dihubungkan menjadi satu komponen graf. Algoritma Kruskal melakukan pencarian MST dengan mengurutkan sisi-sisi dari bobot terkecil hingga terbesar terlebih dahulu, kemudian memilih sisi terkecil diantara keseluruhan sisi yang ada. Apabila sisi-sisi yang terpilih tidak membentuk siklus maka akan diperoleh MST dan kemudian dapat dicari juga nilai MRCST (Sumardi dkk., 2021). Pada penelitian ini, penulis tertarik untuk membuat algoritma baru yang disebut Algoritma *Enhanced* Sollin yang terkontruksi dari Algoritma Sollin demikian juga pada Algoritma *Enhanced* Kruskal. Kedua algoritma tersebut dibangun untuk melihat apakah MRCST yang dihasilkan dapat lebih optimal dan melihat pengaruhnya dari perbandingan hasil kedua algoritma baru tersebut.

1.2 Rumusan Masalah

Bagaimana pengaruh parameter-parameter pada Algoritma *Enhanced* Sollin dan Algoritma *Enhanced* Kruskal terhadap hasil yang diperoleh dalam menyelesaikan permasalahan MRCST?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membandingkan kinerja Algoritma *Enhanced* Sollin dan Algoritma *Enhanced* Kruskal dengan Algoritma Sollin dan Kruskal dalam menyelesaikan *Minimum Routing-Cost Spanning Tree.*

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

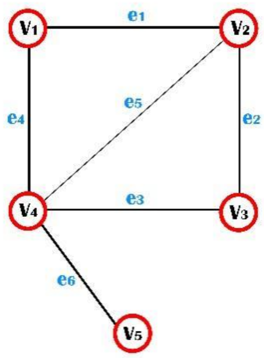
1. memberikan kontribusi dalam pengembangan desain jaringan yang lebih efektif dan efesien;
2. meningkatkan efesiensi dalam berbagai bidang atau industri yang menggunakan rute perjalanan dalam pendistribusiannya;
3. memberikan pengetahuan dan wawasan baru dalam penggunaan Algoritma Sollin dan Algoritma Kruskal khususnya dalam menyelesaikan masalah optimasi.

II. TINJAUAN PUSTAKA

2.1 Konsep Dasar Graf

Graf merupakan salah satu cabang matematika diskrit yang digunakan untuk merepresentasikan hubungan antara objek-objek tertentu. Graf G didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak-kosong dari titik (*vertex*) dan E adalah himpunan sisi (*edge*) yang menghubungkan sepasang titik pada graf. Suatu graf dimungkinkan tidak mempunyai sisi satu buahpun, tetapi harus memiliki satu titik (Aryana dan Sangkala, 2023).

Untuk memahami struktur suatu graf, terdapat beberapa konsep dasar yang perlu dipahami, seperti ketetanggaan, derajat, lintasan, sirkuit, dan jarak antar titik. Konsep-konsep tersebut menjadi dasar dalam berbagai permasalahan optimasi pada teori graf. Suatu titik disebut bertetangga dengan titik lainnya apabila kedua titik tersebut dihubungkan oleh sisi yang sama. Pada Gambar 2.1.1, titik v_1 bertetangga dengan titik v_2 namun tidak bertetangga dengan titik v_3.



Gambar 2.1.1. Contoh Graf G dengan Lima Titik.

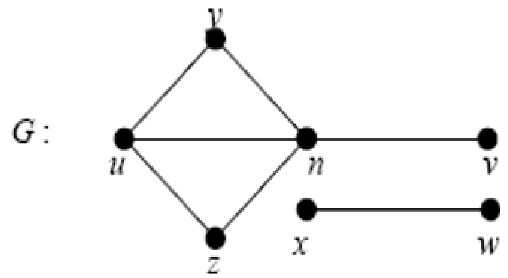
Selain ketetanggaan, konsep lain yang penting adalah derajat (*degree*) suatu titik. Derajat suatu titik menyatakan banyaknya sisi yang terhubung pada titik tersebut (Anggito, 2019). Sebagai contoh, pada Gambar 2.1.1, titik v_4 dikatakan berderajat 4 (empat) karena terhubung dengan 4 (empat) sisi yang menghubungkan titik v_4 dengan $v_1, v_2, v_3,$ dan v_5.

Hubungan antar titik dalam graf dapat ditinjau melalui lintasan yang menghubungkan titik-titik tersebut. Dalam teori graf dikenal beberapa istilah yang berkaitan dengan lintasan, yaitu *walk*, *trail*, dan *path*. *Walk* (jalan) adalah barisan berhingga dari titik dan sisi dimulai dan diakhiri sedemikian sehingga setiap sisi menempel dengan titik sebelum dan sesudahnya. Jika semua sisi (tetapi tidak perlu semua titik) suatu *walk* berbeda, maka *walk* tersebut disebut *trail* (jejak). Selanjutnya, jika semua titik dalam *trail* juga berbeda maka disebut lintasan (*path)* (Apriyani, 2024). Pada Gambar 2.1.1, contoh *walk* adalah $v_1 - e_1 - v_2 - e_2 - v_3 - e_3 - v_4 - e_5 - v_2 - e_1 - v_1 - e_4 - v_4 - e_6 - v_5$, contoh *trail* adalah $v_2 - e_2 - v_3 - e_3 - v_4 - e_5 - v_2 - e_1 - v_1 - e_4 - v_4 - e_6 - v_5$ sedangkan contoh lintasan adalah $v_1 - e_1 - v_2 - e_2 - v_3 - e_3 - v_4 - e_6 - v_5$.

Lintasan yang berawal dan berakhir pada titik yang sama disebut lintasan tertutup (*closed path*) atau sirkuit (*circuit*). Sirkuit dibedakan menjadi dua macam, yaitu sirkuit genap dan sirkuit ganjil. Sirkuit genap merupakan sirkuit dengan banyaknya titik genap, sedangkan sirkuit ganjil merupakan sirkuit dengan banyaknya titik ganjil. Selain itu, jejak tertutup (*closed trail*) yang memiliki titik awal dan titik akhir yang sama disebut siklus (*cycle*).

Suatu graf G dikatakan terhubung (*connected graph*) jika untuk setiap pasangan titik u dan v pada graf terdapat lintasan yang menghubungkan keduanya. Sebaliknya, graf G dikatakan tidak terhubung (*disconnected graph*) jika terdapat paling sedikit satu pasangan titik yang tidak memiliki lintasan penghubung. Pada graf tidak terhubung, himpunan titik dan sisi dapat terbagi menjadi beberapa komponen terhubung yang saling terpisah.

Konsep keterhubungan graf berkaitan erat dengan pengukuran jarak antar titik pada suatu graf. Jarak antara dua titik u dan v pada graf G adalah panjang lintasan terpendek dari titik u ke v dinotasikan dengan $d(u, v)$. Jika tidak ada lintasan dari titik u ke v, maka $d(u,v) = \infty$. Sebagai contoh pada Gambar 2.1.2, $d(u,v) = 2$ sedangkan $d(v,w) = \infty$ (Barahama, dkk., 2021).



Gambar 2.1.2 Contoh Graf G dengan Tujuh Titik.

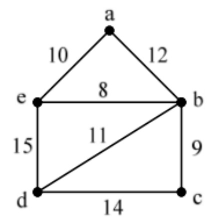
Berdasarkan jarak antar titik, dapat ditentukan nilai eksentrisitas suatu titik. Eksentrisitas merupakan jarak terjauh dari suatu titik ke seluruh titik lainnya pada graf. Nilai eksentrisitas dihitung untuk setiap titik, dan nilai eksentrisitas terbesar pada suatu graf disebut diameter graf. Diameter graf G dinotasikan dengan $D(G)$ dan didefinisikan sebagai nilai maksimum dari seluruh eksentrisitas titik pada graf tersebut (Hua dan Wang, 2013).

Selain ditinjau dari lintasan dan jaraknya, graf juga dapat dibedakan berdasarkan karakteristik sisi yang dimilikinya. Salah satu karakteristik tersebut adalah keberadaan *loop* dan sisi paralel. *Loop* (juga dikenal sebagai gelang) adalah sisi yang titik awalnya sama dengan titik akhirnya, sedangkan garis paralel (juga dikenal sebagai sisi ganda) adalah dua atau lebih sisi yang menghubungkan titik yang sama.

Berdasarkan ada tidaknya *loop* dan sisi paralel, graf secara umum dibedakan menjadi dua jenis, yaitu graf sederhana (*simple graph*) dan graf tidak sederhana (*unsimple graph*).

a. Graf sederhana (*simple graph*)
   Graf sederhana G didefinisikan sebagai graf yang tidak terdapat sisi gelang (*loop*) maupun sisi ganda (paralel) pada setiap titik.
b. Graf tidak sederhana (*unsimple graph*)
   Graf tidak sederhana G didefinisikan sebagai graf yang terdapat sisi ganda (paralel) atau berbentuk gelang (*loop*) pada salah satu atau lebih titik.

Selain klasifikasi tersebut, graf juga dapat dibedakan berdasarkan ada atau tidaknya bobot pada setiap sisi. Graf yang setiap sisinya memiliki nilai tertentu disebut graf berbobot (*weighted graph*). Bobot pada tiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak antara dua buah tiang listrik, kapasitas, biaya perjalanan antara dua buah kota, waktu tempuh pesan (*message*) dari sebuah titik komunikasi ke titik komunikasi lain, ongkos produksi, dan sebagainya (Nugraha, 2011). Untuk lebih jelasnya, graf berbobot dapat digambarkan pada Gambar 2.1.3.



Gambar 2.1.3. Contoh Graf Berbobot.

Pada graf berbobot, salah satu permasalahan yang paling banyak dikaji adalah pencarian pohon merentang dengan bobot minimum. Oleh karena itu, terlebih dahulu perlu dipahami konsep pohon (*tree*) dan pohon merentang (*spanning tree*).

Graf pohon (*Tree*) adalah graf terhubung yang di dalamnya tidak memuat *cycle* (Siregar, 2018). Kumpulan beberapa pohon disebut hutan (*forest*). Apabila G adalah graf berbobot, maka *Minimum Spanning Tree* (MST) dari G didefinisikan sebagai jumlah bobot semua sisi pada G. *Spanning Tree* yang berbeda mempunyai bobot yang berbeda pula. Diantara semua *Spanning Tree* dalam graf G, yang berbobot minimum dinamakan *Minimum Spanning Tree* (MST). MST mempunyai terapan yang luas dalam masalah riil (Munir, 2009).

Langkah-langkah menghitung total jarak minimum (*Minimum Spanning Tree*) dari suatu graf sebagai berikut (Nugraha, 2011):

1. Dari suatu graf yang terbentuk, perhatikan apakah memenuhi kriteria suatu *Spanning Tree*.
2. Lakukan pelacakan secara berurutan mulai dari titik pertama sampai dengan titik terakhir.
3. Pada setiap titiknya perhatikan nilai (bobot) tiap-tiap sisinya.
4. Ambil nilai yang paling kecil artinya jarak terpendek dari setiap sisi titik.
5. Lanjutkan sampai seluruh titik tergambar pada *Spanning Tree.*
6. Jumlahkan nilai yang telah dipilih atau jarak minimum yang menghubungkan titik-titik tersebut.

Meskipun MST mampu meminimalkan total bobot sisi yang digunakan, dalam beberapa aplikasi jaringan diperlukan ukuran optimasi lain yang tidak hanya mempertimbangkan bobot sisi, tetapi juga memperhatikan jarak antar seluruh pasangan titik pada pohon yang terbentuk. Permasalahan tersebut dikenal sebagai *Minimum Routing Cost Spanning Tree (MRCST)*.

Minimum Routing Cost Spanning Tree (MRCST) adalah *spanning tree* pada graf berbobot yang meminimalkan jumlah jarak antar pasangan titik-titik yang ada pada graf berbobot (Sari dkk., 2022). Dalam kata lain, MRCST mencari *spanning tree* dengan total bobot (*cost*) terkecil yang menghubungkan semua titik dalam graf.

MRCST banyak diterapkan pada berbagai bidang, seperti desain jaringan komputer, sistem telekomunikasi, dan perancangan infrastruktur komunikasi. Oleh karena itu, berbagai algoritma telah dikembangkan untuk memperoleh solusi MRCST, termasuk algoritma *enhanced* Sollin dan algoritma *enhanced* Kruskal yang menjadi fokus penelitian ini.

2.2 Algoritma Sollin

Algoritma Sollin merupakan metode deterministik yang digunakan untuk menyelesaikan permasalahan *Minimum Spanning Tree* (MST) pada graf berbobot. Cara kerja algoritma Sollin adalah dimulai dengan hutan, dan tumbuh pohon di *subset* dari titik-titik hutan tersebut sampai menjadi satu pohon yang memuat semua titik. Di setiap langkah, sisi yang berbobot minimum dipilih (Yingyu dkk., 2000).

Di setiap iterasi pada algoritma Sollin akan memperoleh jarak minimal diantara dua titik atau titik yang memakai konsep *nearest-neigbhbor*. Maka ditiap sisi dicari titik mana dari titik tersebut yang memperoleh jarak terkecil. Dipastikan tiap titik mempunyai pasangan dan dicek kembali apakah MST yang diperoleh *connected*, jika tidak dipilih *cutset* yang mempunyai jarak terkecil.

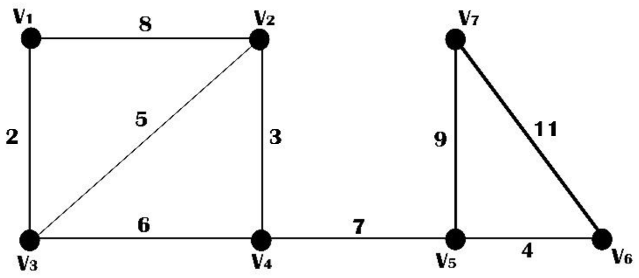
Langkah-langkah penyelesaian algoritma Sollin:

1. Daftarkan semua titik yang ada di graf G.
2. Untuk setiap titik: pilih sisi dengan bobot yang terkecil yang menempel dengan titik tersebut. Jika terjadi ada dua sisi dengan bobot terkecil yang sama, pilih satu. (Pada proses ini mungkin saja terbentuk *forest* yang merupakan gabungan dari k komponen dari Graf G).
3. Tandai sisi tersebut dengan ketentuan bahwa sisi-sisi yang terhubung diberi tanda yang sama. Banyak tanda menunjukkan banyaknya k komponen yang terbentuk.
4. Tentukan sisi terkecil yang berada diluar masing-masing komponen.
5. Hubungkan sisi terkecil tersebut, yang menghubungkan dua komponen. Untuk dua komponen yang baru terhubung, ganti tanda salah satu komponen menjadi sama dengan komponen lainnya (sehingga jumlah komponen menjadi $k - 1$).
6. Ulangi Langkah 4 sehingga $k = 1$.
7. Hapus sisi-sisi yang tidak terpakai.
8. Selesai, MST didapat.

   (Wamiliana, 2022)

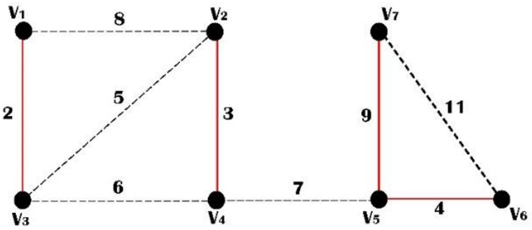
Contoh algoritma Sollin adalah sebagai berikut:

Diberikan graf G.



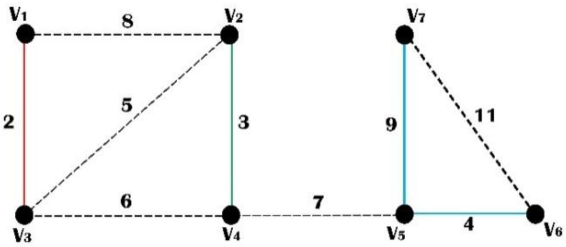
Gambar 2.2.1. Contoh Graf G dengan 7 Titik dan 9 Sisi.

1. $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$
2. Untuk setiap titik: pilih sisi dengan bobot yang terkecil yang menempel dengan titik tersebut. Sisi yang terpilih diberi label warna merah seperti pada Gambar 2.2.2



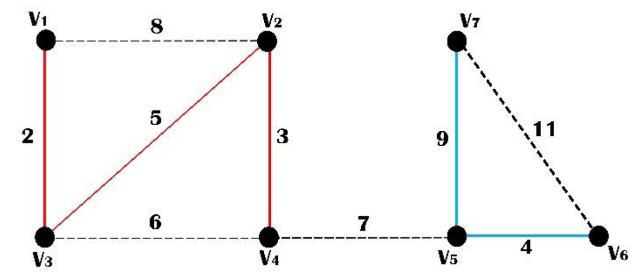
Gambar 2.2.2. Graf G pada Langkah 2.

3. Tandai sisi tersebut dengan ketentuan bahwa sisi-sisi yang terhubung diberi tanda yang sama. Pada Gambar 2.2.3 terbentuk 3 (tiga) komponen.



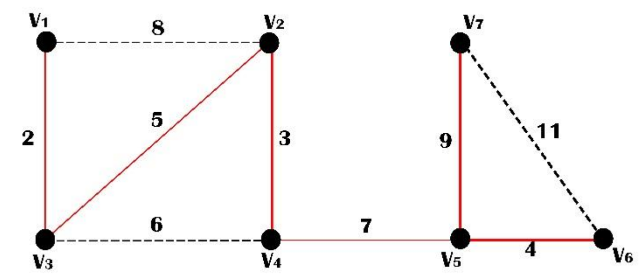
Gambar 2.2.3. Graf G pada Langkah 3.

4. Tentukan sisi terkecil yang berada diluar masing-masing komponen.
5. Hubungkan sisi terkecil tersebut, yang menghubungkan dua komponen. Untuk dua komponen yang baru terhubung, ganti tanda salah satu komponen menjadi sama dengan komponen lainnya.



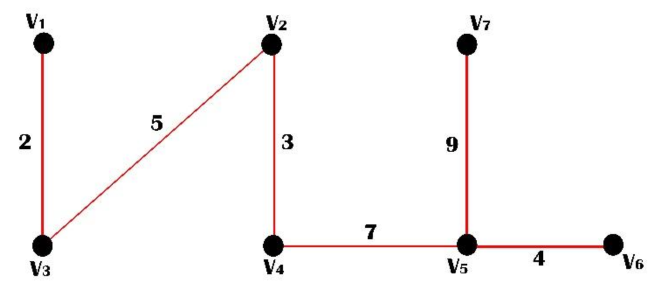
Gambar 2.2.4. Graf G pada Langkah 5.

6. Ulangi Langkah 4 sehingga $k = 1$.



Gambar 2.2.5. Graf G pada Langkah 6.

7. Hapus sisi-sisi yang tidak terpakai.



Gambar 2.2.6. Graf G pada Langkah 7.

8. Selesai, MST didapat.

Total bobot *minimum spanning tree* pada Gambar 2.2.6 adalah 30.

2.3 Algoritma Kruskal

Algoritma kruskal merupakan sebuah algoritma dalam teori graf yang mencari sebuah pohon merentang minimum atau sering dikenal dengan sebutan "*minimum spanning tree*" untuk sebuah graf berbobot yang terhubung (Marsudi, 2016). Algoritma kruskal pertama kali muncul pada tahun 1956 dalam sebuah tulisan yang ditulis oleh Joseph Kruskal (Putra dkk., 2016).

Pada algoritma Kruskal, sisi-sisi pada graf diurut terlebih dahulu berdasarkan bobotnya secara *increasing order* (dari kecil ke besar). Garis yang dimasukkan ke dalam himpunan T adalah garis di graf G sedemikian sehingga T adalah pohon. Pada keadaan awal, himpunan T berupa himpunan kosong, dan titik-titik di jaringan merupakan *null graf* (graf tanpa sisi). Garis dari graf G ditambahkan ke T jika garis tersebut tidak membentuk sirkuit di T (Wamiliana, 2022).

Langkah – langkah algoritma kruskal adalah sebagai berikut:
Set $T = \emptyset$, input $=$ graf banyaknya titik $= n$.

1. Sortir garis dari urutan bobot terkecil ke terbesar (*increasing orde*r).
2. Pilih garis terkecil dalam sortir dan masukkan ke T.
3. Pilih garis berikutnya dalam sortir.
4. Cek apakah penambahan garis tersebut pada T menyebabkan terjadinya sirkuit. Jika ya, buang garis tersebut dari sortir dan kembali ke Langkah 2. Jika tidak, masukkan garis tersebut pada T dan ke Langkah 4.
5. Cek apakah $|T| = n - 1$. Jika Ya, STOP, solusi didapat. Jika tidak, kembali ke Langkah 2.
   (Wamiliana, 2022)

Contoh Algoritma Kruskal adalah sebagai berikut:

Diberikan graf $G(V, E)$ yang ditunjukkan pada Gambar 2.2.1.

1. Urutkan sisi dari bobot terkecil ke terbesar (*increasing order*).
   $n = 7$ (banyaknya titik)
   $e_{13} = 2$
   $e_{24} = 3$
   $e_{56} = 4$
   $e_{23} = 5$
   $e_{34} = 6$
   $e_{45} = 7$
   $e_{12} = 8$
   $e_{57} = 9$
   $e_{67} = 11$

2. Pilih $e_{13} = 2, T = \{e_{13}\}$, $|T| = 1$
3. Pilih $e_{24} = 3$
4. Penambahan e_{24} di T tidak menyebabkan terbentuknya sirkuit,
   jadi $T = \{e_{13}, e_{24}\}$, $|T| = 2$
5. $|T| = 2, \neq 7 - 1$. Kembali ke Langkah 2.
   Pilih $e_{56} = 4$.
   Penambahan e_{56} di T tidak menyebabkan terbentuknya sirkuit,
   jadi $T = \{e_{13}, e_{24}, e_{56}\}$, $|T| = 3$.
   $|T| = 3, \neq 7 - 1$. Kembali ke Langkah 2.

   Pilih $e_{23} = 5$.
   Penambahan e_{23} di T tidak menyebabkan terbentuknya sirkuit,
   jadi $T = \{e_{13}, e_{24}, e_{56}, e_{23}\}$, $|T| = 4$.
   $|T| = 4, \neq 7 - 1$. Kembali ke Langkah 2.

Pilih $e_{34} = 6$.

Penambahan e_{34} di T menyebabkan terbentuknya sirkuit,

Sehingga e_{34} dibuang, kembali ke Langkah 2.

Pilih $e_{45} = 7$.

Penambahan e_{45} di T tidak menyebabkan terbentuknya sirkuit,

jadi $T = \{e_{13}, e_{24}, e_{56}, e_{23}, e_{45}\}$, $|T| = 5$.

$|T| = 5, \neq 7 - 1$. Kembali ke Langkah 2.

Pilih $e_{12} = 8$..

Penambahan e_{12} di T menyebabkan terbentuknya sirkuit,

Sehingga e_{12} dibuang, kembali ke Langkah 2.

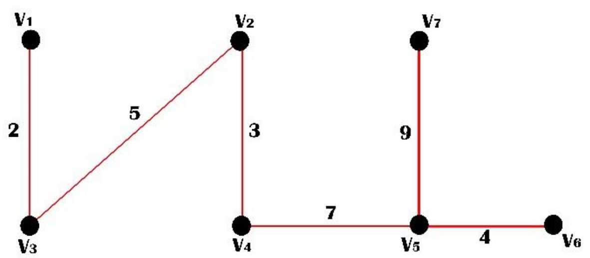
Pilih $e_{57} = 9$.

Penambahan e_{57} di T tidak menyebabkan terbentuknya sirkuit,

jadi $T = \{e_{13}, e_{24}, e_{56}, e_{23}, e_{45}, e_{57}\}$, $|T| = 6$.

$|T| = 6$, STOP.

Karena banyaknya elemen di T sama dengan $n - 1$, maka telah terbentuk *Spanning Tree* T, yang juga sekaligus *Minimum Spanning Tree* sesuai Gambar 2.3.1 dengan total bobot MST = 30.



Gambar 2.3.2. Graf G Hasil Algoritma Kruskal.

Proses Algoritma Kruskal ini juga dapat dilakukan dengan menggunakan tabel sebagai berikut.

Tabel 2.3.1. Contoh Tabel Algoritma Kruskal untuk Penyelesaian Graf pada Gambar 2.2.1

Iterasi	Garis yang dipilih	Membentuk *cycle* atau sirkuit?		T	|T| = n-1?		Keterangan
		Ya	Tidak		Ya	Tidak	
1	e_{13}		√	$\{e_{13}\}$		√	
2	e_{24}		√	$\{e_{13}, e_{24}\}$		√	
3	e_{56}		√	$\{e_{13}, e_{24}, e_{56}\}$		√	
4	e_{23}		√	$\{e_{13}, e_{24}, e_{56}, e_{23}\}$		√	
5	e_{34}	√		$\{e_{13}, e_{24}, e_{56}, e_{23}\}$		√	Sisi e_{34} dibuang dari list. Elemen di T tetap.
6	e_{45}		√	$\{e_{13}, e_{24}, e_{56}, e_{23}, e_{45}\}$		√	
7	e_{12}	√		$\{e_{13}, e_{24}, e_{56}, e_{23}, e_{45}\}$		√	Sisi e_{12} dibuang dari list. Elemen di T tetap.
8	e_{57}		√	$\{e_{13}, e_{24}, e_{56}, e_{23}, e_{45}, e_{57}\}$	√		Karena |T| = n-1 = 6, maka STOP.
MST yang terbentuk dapat dikonstruksi dengan menggambarkan *tree* sesuai dengan urutan garis yang masuk ke dalam T. Dari konstruksi ini diketahui bahwa pada proses pembentukan MST dengan algoritma Kruskal dimungkinkan terbentuknya *forest* (hutan).							

III. METODE PENELITIAN

3.1. Jenis dan Sumber Data

Data yang digunakan penulis dalam penelitian ini ialah data sekunder. Penelitian ini menggunakan data yang dibangkitkan secara acak menggunakan distribusi *uniform* dan nilainya berupa bilangan bulat (*integer*). Data tersebut jika direpresentasikan dalam graf merupakan data graf lengkap yang memiliki jumlah titik 10, 20, 30 sampai 100 dengan setiap jumlah titik memiliki 30 kasus berbeda. Data tersebut diambil dari penelitian yang dilakukan Wamiliana dkk. (2015).

3.2. Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada Tahun Ajaran 2025/2026 di Program Studi Magister Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

3.3. Tahapan Penelitian

Adapun langkah – langkah yang digunakan dalam penelitian ini sebagai berikut :

1. Identifikasi masalah

   Pada penelitian ini diawali dengan melakukan kajian pustaka tentang algoritma yang tepat dalam menyelesaikan masalah MRCST yaitu dengan menggunakan Algoritma Sollin dan Algoritma Kruskal.

2. Pencarian MST dan MRCST

   a. Pencarian menggunakan Algoritma Sollin

      Setelah membuat graf berbobot pada salah satu sampel data, dilakukan perhitungan dan analisis data menggunakan Algoritma Sollin untuk menentukan MST yang kemudian dilanjutkan dengan menentukan MRCST.

   b. Pencarian menggunakan Algoritma Kruskal

      Setelah membuat graf berbobot pada salah satu sampel data, dilakukan perhitungan dan analisis data menggunakan Algoritma Kruskal untuk menentukan MST yang kemudian dilanjutkan dengan menentukan MRCST.

3. Setelah diperoleh MST dan MRCST melalui Algoritma Sollin dan Kruskal, maka dibuat algoritma baru yang merupakan kelanjutan dari Algoritma Sollin disebut Algoritma *Enhanced* Sollin serta algoritma baru yang merupakan kelanjutan dari Algoritma Kruskal disebut Algoritma *Enhanced* Kruskal. Kemudian dicari nilai MST dan MRCST dari kedua algoritma baru tersebut.

4. Mengulangi Langkah 2 dan 3 untuk semua data lainnya yang akan dicari menggunakan bahasa pemrograman *Python* agar mempermudah dalam perhitungan.

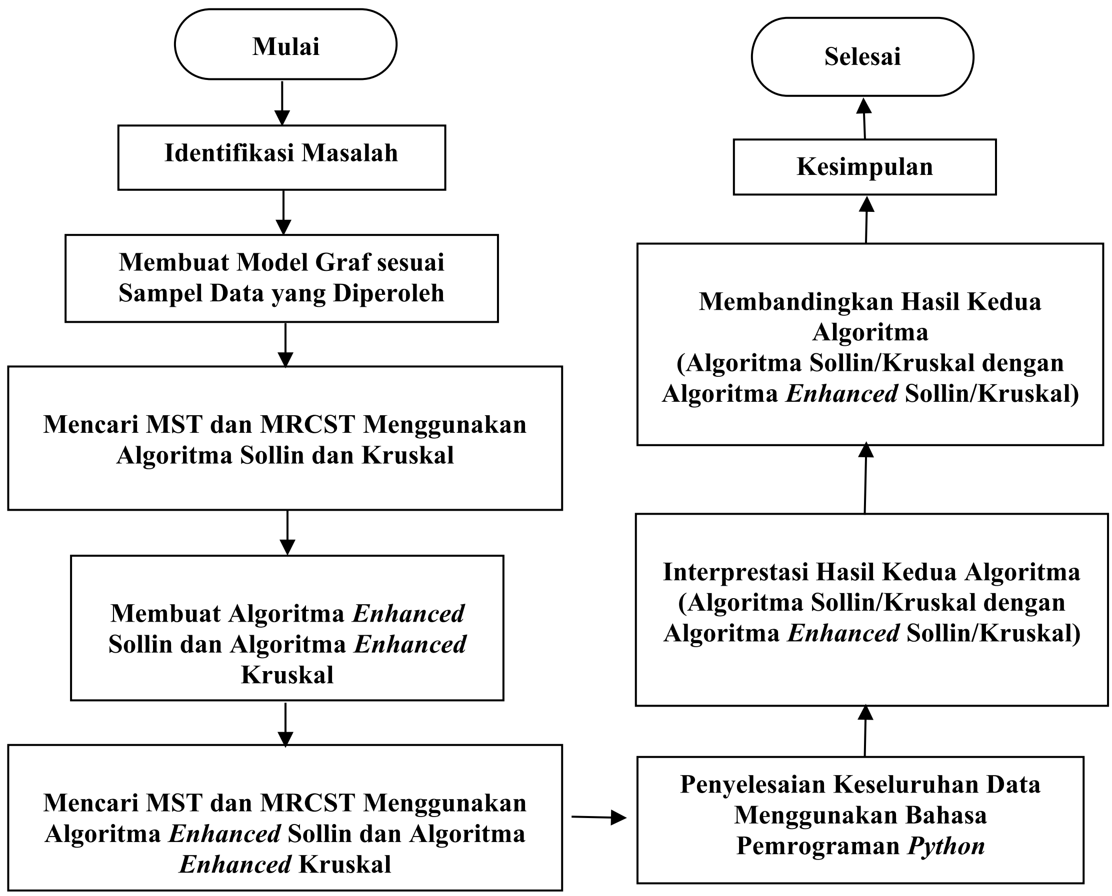
5. Membandingkan hasil MRCST

   Dilakukan perbandingan hasil MRCST yang telah didapatkan dari kedua algoritma serta memperhatikan banyaknya iterasi dari masing-masing algoritma untuk menentukan algoritma mana yang lebih efektif.

6. Penarikan kesimpulan

   Diambil kesimpulan setelah dilakukan analisis terhadap hasil yang diperoleh.

3.4 Alur Penelitian



V. KESIMPULAN

Berdasarkan hasil observasi terhadap Algoritma *Enhanced* Sollin dan *Enhanced* Kruskal, dapat disimpulkan bahwa semakin besar *orde* graf atau jumlah titik yang digunakan, maka selisih nilai MRCST antara lgoritma asli dan algoritma hasil pengembangan cenderung semakin meningkat. Pada data yang diuji, selisih terbesar diperoleh oleh Algoritma *Enhanced* Sollin pada graf dengan *orde* 90, yaitu sebesar 48.458,87.

Selain itu, Algoritma *Enhanced* Sollin dan *Enhanced* Kruskal masing-masing menunjukkan karakteristik yang berbeda. Pada graf dengan *orde* 10–30 dan 60, Algoritma *Enhanced* Kruskal menghasilkan peningkatan yang lebih baik dibandingkan Algoritma *Enhanced* Sollin, ditinjau dari besarnya penurunan nilai MRCST terhadap algoritma aslinya. Sebaliknya, pada graf dengan *orde* 40, 50, dan 70–100, Algoritma *Enhanced* Sollin menunjukkan kinerja yang lebih unggul dibandingkan Algoritma *Enhanced* Kruskal.

DAFTAR PUSTAKA

Anggito, B.H.P.B. 2019. Aplikasi Graf dalam Permainan. *Makalah IF2120 Matematika Diskrit.* Institut Teknologi Bandung.

Apriyani, D.C.N. 2024. *Bahan Ajar Matematika Diskret (Graf).* LPPM Press STKIP PGRI, Pacitan. 57 hlm.

Aryana, R., & Sangkala, N.S. 2023. Implementation Of Set Theory In Developing The Definition Of Graph Theory. *Journal of Mathematics Education and Learning,* **2** (1), pp. 12-16.

Barahama, R.M. 2021. *Eksentrisitas Digraf pada Graf Gir Menggunakan Algoritma Breadth First Search.* (Skripsi). Universitas Sam Ratulangi. Manado. 41 p.

Hieu, N.M., Quoc, P., & Nghia, N.D. 2011. An Approach of Ant Algortihm for Solving Minimum Routing Cost Spanning Tree Problem. *In Proceedings of the 2nd Symposium on Information and Communication Technology,* pp. 5-10.

Hua, H., & Wang, M. 2013. On Harary Index and Traceable Graphs. *Communication in Mathematical and in Computer Chemistry,* (70), pp. 297-300.

Marsudi. 2016. *Teori Graf.* Universitas Brawijaya Press, Malang. 172 hlm.

Munir, R. 2009. *Matematika Diskrit.* Edisi 3. Informatika, Bandung. 561 hlm.

Nugraha, D.W. 2011. Aplikasi Algoritma Prim untuk Menentukan Minimum Spanning Tree Suatu Graf Berbobot dengan Menggunakan Pemrograman Berorientasi Objek. *Jurnal Ilmiah Foristek,* **1** (2), pp. 70-79.

Nurdiyanto, T., & Susanti, E. 2019. Efesiensi Penggunaan Matriks In-Degree untuk Mengkontruksi Spanning-Tree Pada Graf Berarah. *JES-MAT,* **5** (1), pp. 1-15.

Putra, E.D.A., Ernawati & Coastera, F.F. 2016. Penerapan Open Street Map untuk Mencari Lokasi ATM Terdekat dengan Algoritma Kruskal Berbasis enhancedphone Android (Studi Kasus: Lokasi ATM di Kota Bengkulu). *Jurnal Rekursif,* **4**, pp. 196-208.

Rizki, S. (2012). Penerapan teori graf untuk menyelesaikan masalah minimum spanning tree (MST) menggunakan algoritma Kruskal. **1** (2), pp. 142-152.

Sari, R.P., Wamiliana, Junaidi, A., & Susanty, W. 2022. The Diameter and Maximum Link of The Minimum Routing Cost Spanning Tree Problem. *Sciend and Technology Indonesia,* **7** (4), pp. 481-485.

Siregar, M.K. 2018. *Matematika Diskrit.* Perahu Litera, Lampung. 111 hlm.

Sumardi, H., Afnaria, & Panggabean, S. 2021. Pengembangan Algoritma Prim untuk Menentukan Minimum Spanning Forest. *Majamath: Jurnal Matematika dan Pendidikan Matematika,* **4** (1), pp. 54-61.

Suparman, J. 2024. *Perbandingan Penerapan Algoritma Modifikasi Sollin dan Algoritma Modifikasi Prim untuk Menyelesaikan Minimum Routing-Cost Spanning Tree (MRCST) Problem.* (Tesis). Universitas Lampung. Manado. 70 p.

Tan, Q.P. 2012a. A Heuristic Approach for Solving Minimum Routing Cost Spanning Tree Problem. *International Journal of Machine Learning and Computing,* **2** (4), pp. 406.

Tan, Q.P. 2012b. A Genetic Approach for Solving Minimum Routing Cost Spanning Tree Problem. *International Journal of Machine Learning and Computing,* **2** (4), pp. 410.

Wamiliana, Elfaki, F.A.M., Usman, M. & Azram, M. 2015. Some greedy based algorithms for multi periods degree constrained minimum spanning tree problem. *ARPN Journal of Engineering and Applied Science*, **10** (21), pp. 10147-10152.

Wamiliana. 2022. *Minimum Spanning Tree dan Desain Jaringan.* Pusaka Media, Lampung. 127 hlm.

Yingyu, W., Yinlong, X., Xiadong, G., & Gouliang, C. 2000. Efficient Minimum Spanning Tree Algorithms On The Reconfigurable Mesh. *Journal Computation Science & Technology,* **15** (2), pp. 116-125.