

PERBANDINGAN ALGORITMA KRUSKAL DAN PRIM DALAM MENENTUKAN MINIMUM SPANNING TREE UNTUK OPTIMASI JARINGAN DISTRIBUSI LISTRIK

Euis Sartika¹, Anie Lusiani²

^{1,2} Politeknik Negeri Bandung

sartikaeuis@gmail.com¹, ani.lusiani@polban.ac.id²

Received 15 September 2025; revised 27 February 2026; accepted 31 July 2026..

ABSTRAK

Penelitian ini bertujuan membandingkan performa algoritma Kruskal dan Prim dalam menyelesaikan permasalahan *Minimum Spanning Tree* (MST) pada jaringan distribusi listrik PLN UP3 Cimahi menggunakan Google Colab berbasis Python. Penelitian menggunakan metode penelitian terapan dengan data berupa panjang kabel jaringan listrik yang dimodelkan sebagai graf berbobot. Analisis dilakukan menggunakan pustaka *NetworkX* pada *Google Colab* untuk membentuk MST menggunakan kedua algoritma. Hasil penelitian menunjukkan bahwa kedua algoritma menghasilkan total bobot MST yang sama sebesar 1258,05 meter, lebih kecil dibandingkan panjang jaringan awal sebesar 1424,61 meter, sehingga diperoleh efisiensi sebesar 11,69%. Meskipun menghasilkan solusi yang identik, algoritma Kruskal lebih sederhana diterapkan pada jaringan dengan jumlah simpul yang besar karena proses pengurutan bobot dilakukan di awal. Penelitian ini menunjukkan bahwa *Google Colab* merupakan media yang efektif untuk pembelajaran maupun implementasi optimasi jaringan berbasis teori graf.

Kata kunci: pohon rentang minimum; teori graf; algoritma Kruskal; algoritma prim; google colab

ABSTRACT

This study aims to compare the performance of the Kruskal and Prim algorithms in solving the Minimum Spanning Tree (MST) problem for the PLN UP3 Cimahi electricity distribution network using Python-based Google Colab. The study employs an applied research methodology, utilising data on the lengths of electricity network cables modelled as a weighted graph. The analysis was carried out using the NetworkX library on Google Colab to construct the MST using both algorithms. The results show that both algorithms produced the same total MST weight of 1,258.05 metres, which is less than the initial network length of 1,424.61 metres, resulting in an efficiency of 11.69 per cent. Although they produced identical solutions, the Kruskal algorithm is simpler to apply to networks with a large number of nodes because the weight sorting process is carried out at the

beginning. This study demonstrates that Google Colab is an effective platform for both learning and implementing graph theory-based network optimisation.

Keywords: minimum spanning tree; graph theory; kruskal algorithm; prim algorithm; google colab

PENDAHULUAN

Energi listrik sangat penting bagi kehidupan modern. Agar semua kehidupan, termasuk ekonomi, dapat berjalan, energi listrik harus ada. Berbagai bentuk aktivitas keseharian masyarakat, termasuk TV, lemari es, dispenser, alat pendingin, dan perangkat rumah tangga lainnya, semakin bergantung pada sumber tenaga listrik (Alham et al., 2024). Selain itu, meningkatnya kebutuhan akan sumber tenaga listrik untuk sektor industri adalah fakta yang tidak terelakkan lagi. Pada tahun 2023, setiap orang Indonesia mengonsumsi 1.285 kilowatt-jam listrik per hari. Untuk tahun 2024, targetnya adalah 1.408 kilowatt-jam per hari. Untuk menghadapi peningkatan konsumsi listrik masyarakat, pemerintah berkomitmen untuk mempersiapkan pasokan listrik. Salah satu langkah yang diambil adalah mengoptimalkan sistem transmisi listrik nasional, dengan maksud agar Biaya Pokok Penyediaan (BPP) pembangkit dapat dikurangi, sehingga jumlah subsidi listrik dapat dikurangi. Keunggulan kompetitif yang penting bagi sebuah perusahaan adalah pengelolaan jaringan distribusi yang optimal. Ketepatan waktu dan biaya akan terganggu jika manajemen distribusi tidak berfungsi dengan baik. Penelitian ini berfokus pada mengkaji model analisis jaringan. Istilah umum untuk jaringan termasuk jaringan kerja, jaringan transportasi, jaringan listrik, dan jaringan komunikasi. Fokus masalah jaringan terdiri dari lintasan terpendek (shortest path), pohon rentang minimum (minimum spanning tree), dan aliran maksimum (Simbolon, 2021). Hal ini dimaksudkan untuk mencapai kondisi yang optimal, di mana penggunaan jaringan listrik menjadi efisien (Wamiliana et al., 2020).

Cabang ilmu matematika yang dikenal sebagai graph theory mempelajari struktur graf dalam konteks matematis. Graph theory dapat diterapkan dalam berbagai bidang pengetahuan dan bahkan dalam aktivitas sehari-hari (Alham et al., 2024). Dalam praktiknya, teori graf banyak digunakan dalam bidang ilmu informatika, terutama untuk aplikasinya dalam analisis jaringan; dalam ilmu kimia,

graf digunakan untuk menunjukkan pemodelan bahan kimia; dan dalam bidang kelistrikan, konsep jaringan listrik juga dapat digambarkan melalui grafik, yang merupakan aplikasi yang paling umum dalam bidang kelistrikan: sistem saluran listrik (Sembiring, 2022). Graf menunjukkan objek diskrit dan hubungan mereka. Satu Pewarnaan graf, masalah penjual keliling, dan perolehan pohon rentang minimum adalah aplikasi teori graf yang paling banyak digunakan (Sembiring, 2022). Graf menunjukkan objek diskrit dan hubungan mereka. Sebuah graf G adalah pasangan dari dua himpunan (V, E) , yang ditulis dengan notasi $G = (V, E)$. Himpunan V adalah himpunan simpul (vertex atau node) yang tidak kosong yang diwakili oleh titik, dan himpunan E adalah himpunan sisi (edges) yang diwakili oleh garis yang menghubungkan dua simpul (Munir, 2016). Pohon Rentang Minimum (MST) adalah komponen dari graf berbobot yang menghubungkan semua simpul tanpa membentuk siklus, sambil memastikan total bobot tepi diminimalkan. MST banyak digunakan dalam desain jaringan untuk secara efektif mengoptimalkan biaya dalam proyek seperti desain jaringan listrik, telekomunikasi, dan distribusi air. Konsep MST sangat relevan dalam desain jaringan pipa distribusi air, karena dapat meminimalkan total panjang pipa yang digunakan, yang secara langsung berdampak pada penghematan biaya konstruksi dan pemeliharaan (Ruhimat et al., 2024).

Google Colab atau *Google Colaboratory* adalah sebuah layanan yang disediakan oleh Google yang memungkinkan kita menulis dan menjalankan kode Python di browser, secara langsung dan interaktif. Ini adalah alat yang sangat populer di kalangan peneliti, pengembang, dan pelajar karena menyediakan lingkungan yang berbasis cloud untuk pengembangan aplikasi, analisis data, dan pembelajaran mesin tanpa perlu konfigurasi perangkat keras yang rumit. Colab menyediakan integrasi langsung dengan Google Drive, sehingga memungkinkan Anda untuk menyimpan dan berbagi proyek. (Rofiq & Sitio, 2024). *Google Colab* adalah alat berbasis web yang memungkinkan kita untuk menulis dan menjalankan kode Python di browser dengan kemampuan komputasi cloud yang disediakan oleh Google. Colab mendukung eksekusi kode berbasis Python secara interaktif dengan menggunakan notebook Jupyter, yang memungkinkan kita untuk menggabungkan kode, teks, visualisasi data, dan lain-lain dalam satu dokumen.

Penelitian sebelumnya lebih banyak membandingkan algoritma Prim dan Kruskal secara teoritis atau pada jaringan transportasi maupun jaringan pipa. Penelitian yang mengimplementasikan kedua algoritma menggunakan Google Colab berbasis Python pada optimasi jaringan distribusi listrik PLN masih sangat terbatas. Oleh karena itu penelitian ini membandingkan kedua algoritma dalam lingkungan komputasi berbasis cloud sehingga dapat digunakan sebagai media analisis maupun pembelajaran teori graf.

Data berupa panjang kabel antar simpul jaringan listrik diperoleh dari PLN UP3 Cimahi. Data dimodelkan menjadi graf tak berarah berbobot menggunakan NetworkX. Selanjutnya MST dibentuk menggunakan algoritma Prim dan Kruskal. Total bobot hasil MST dibandingkan dengan panjang jaringan awal untuk memperoleh tingkat efisiensi menggunakan Persamaan. Tujuan penelitian adalah menentukan dan membandingkan MST yang dihasilkan oleh algoritma Prim dan Kruskal, menghitung reduksi panjang terhadap seluruh sisi kandidat, memverifikasi keunikan solusi optimal, serta mengidentifikasi solusi terbaik kedua. Hasil penelitian diharapkan memberikan dasar komputasional yang lebih transparan bagi analisis jaringan, sekaligus menegaskan batas interpretasi MST dalam konteks jaringan listrik.

METODE PENELITIAN

Penelitian ini dilakukan untuk menentukan optimasi jaringan kabel listrik menggunakan metode *Minimum Spanning Tree* (MST) melalui algoritma Prim dan Kruskal (Abhisyeka et al., 2025). Penelitian ini merupakan penelitian terapan (*applied research*) yang bertujuan menerapkan konsep teori graf, khususnya *Minimum Spanning Tree* (MST), untuk mengoptimalkan jaringan distribusi listrik menggunakan algoritma Prim dan Kruskal. Penelitian ini tidak hanya membandingkan hasil kedua algoritma, tetapi juga mengimplementasikannya menggunakan bahasa pemrograman *Python* pada *platform Google Colab* sehingga proses perhitungan dan visualisasi graf dapat dilakukan secara otomatis dan terdokumentasi dengan baik.

Data penelitian diperoleh dari PLN UP3 Cimahi, berupa data jaringan distribusi listrik yang terdiri atas titik-titik jaringan (node) dan panjang kabel antar

titik dalam satuan meter. Pada penelitian ini, setiap titik jaringan direpresentasikan sebagai simpul (*vertex*), sedangkan setiap jalur kabel direpresentasikan sebagai sisi (*edge*) yang memiliki bobot sesuai panjang kabel. Dataset terdiri atas 30 simpul dan 32 sisi yang membentuk graf berbobot tak berarah (*undirected weighted graph*). Bobot setiap sisi menjadi parameter utama dalam proses pembentukan *Minimum Spanning Tree*.

Data jaringan distribusi listrik dimodelkan ke dalam bentuk graf berbobot menggunakan teori graf. Suatu graf dinyatakan sebagai

$$G = (V, E)$$

Keterangan :

V : himpunan simpul (*vertex*) yang merepresentasikan titik jaringan listrik.

E : himpunan sisi (*edge*) yang menghubungkan dua simpul.

Setiap sisi memiliki bobot yang menunjukkan panjang kabel dalam satuan meter.

$$W: E \rightarrow \mathbb{R}^+$$

Model graf tersebut selanjutnya digunakan sebagai masukan untuk pembentukan *Minimum Spanning Tree* (MST). Tujuan pembentukan MST adalah memperoleh jaringan yang mampu menghubungkan seluruh simpul tanpa membentuk siklus dengan total panjang kabel minimum.

Implementasi algoritma dilakukan menggunakan bahasa pemrograman *Python* pada *platform Google Colab* dengan memanfaatkan pustaka *NetworkX* untuk pembentukan graf dan pencarian *Minimum Spanning Tree*, serta *Matplotlib* untuk visualisasi hasil. Tahapan implementasi meliputi:

- a) Memasukkan data simpul dan panjang kabel ke dalam graf berbobot.
- b) Membentuk graf menggunakan fungsi *add_weighted_edges_from()*.
- c) Menjalankan algoritma Prim dan Kruskal untuk memperoleh *Minimum Spanning Tree*.
- d) Menghitung total bobot MST yang dihasilkan oleh masing-masing algoritma.
- e) Memvisualisasikan jaringan hasil optimasi.
- f) Membandingkan hasil kedua algoritma berdasarkan total panjang kabel yang diperoleh.

Proses perhitungan dan simulasi dilakukan menggunakan bahasa

pemrograman *Python* pada *Google Colab*. Selanjutnya dilakukan perbandingan hasil algoritma Prim dan Kruskal berdasarkan total panjang kabel dan efisiensi pembentukan minimum spanning tree untuk menentukan algoritma yang lebih optimal dalam penyelesaian masalah jaringan listrik. Berikut adalah langkah-langkah penelitian yang dilakukan:

$$Efisiensi = \frac{(L_{awal} - L_{MST})}{L_{MST}} \times 100\%$$

Keterangan:

L_{awal} : total Panjang jaringan kabel sebelum dilakukan optimasi (meter);

L_{MST} : total Panjang jaringan kabel hasil Minimum Spanning Tree menggunakan algoritma Prim atau Kruskal (meter);

Efisiensi (%) = persentase penghematan Panjang jaringan setelah dilakukan optimasi.

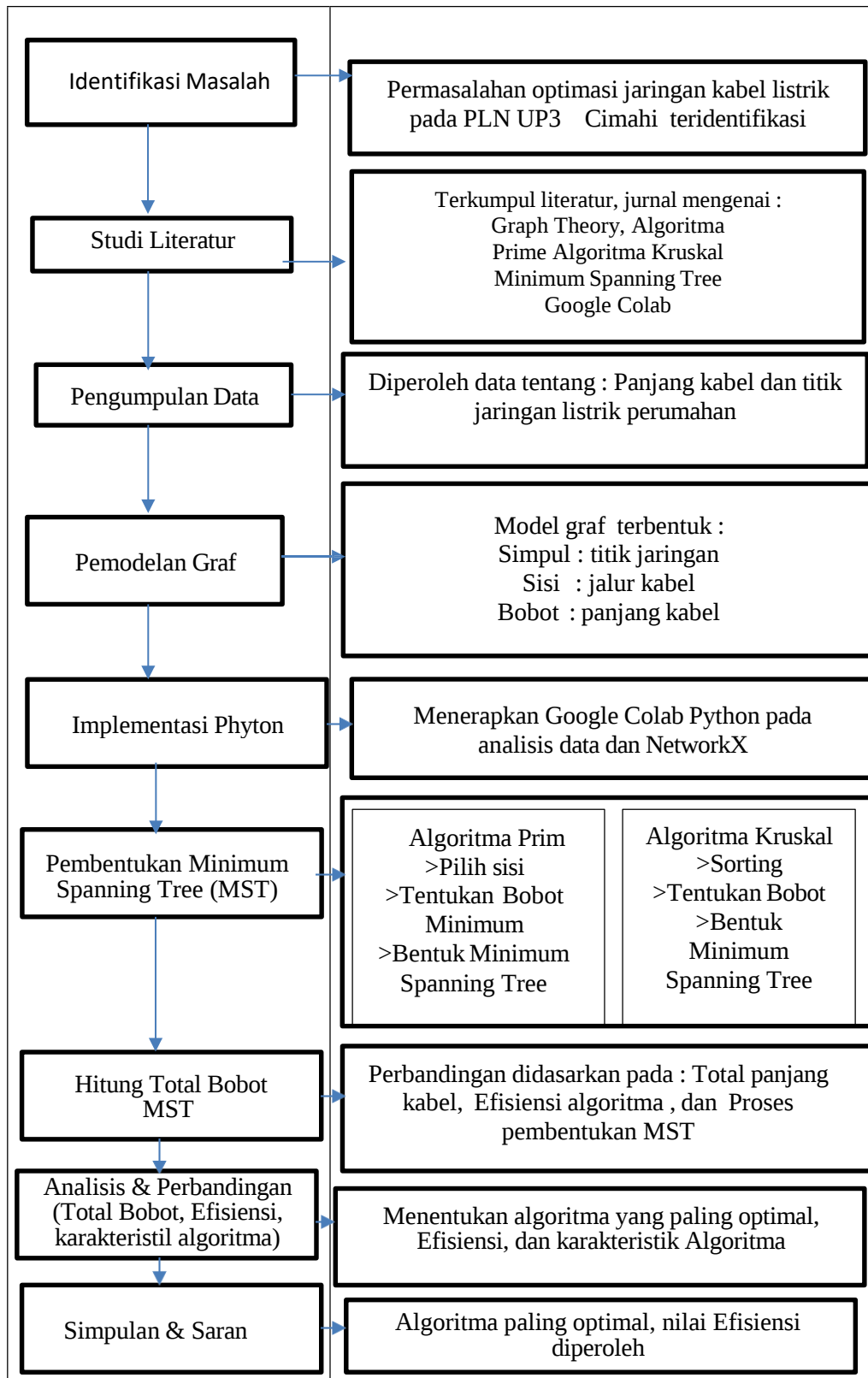
Nilai efisiensi menunjukkan persentase pengurangan panjang jaringan setelah dilakukan optimasi. Semakin besar nilai efisiensi, semakin besar penghematan panjang kabel yang diperoleh.

Analisis data dilakukan dengan membandingkan hasil Minimum Spanning Tree yang diperoleh menggunakan algoritma Prim dan Kruskal. Perbandingan dilakukan berdasarkan tiga aspek, yaitu:

- Total panjang kabel Minimum *Spanning Tree*, yang menunjukkan hasil optimasi jaringan.
- Persentase efisiensi, yaitu penghematan panjang kabel dibandingkan dengan kondisi awal.
- Karakteristik proses algoritma, meliputi mekanisme pemilihan sisi dan kemudahan implementasi pada graf berbobot.

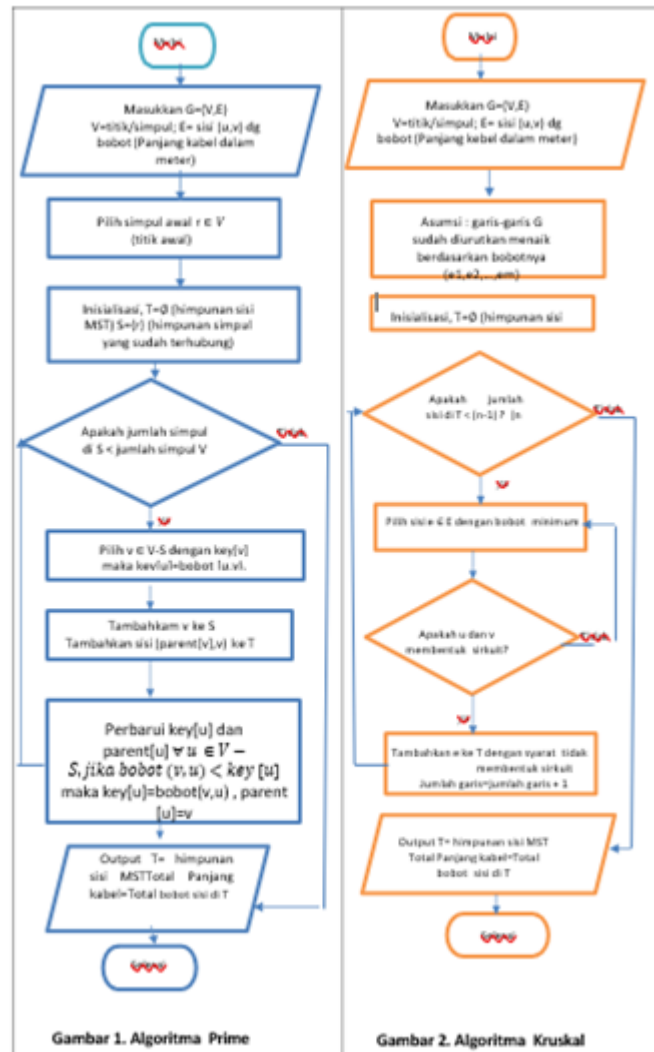
Hasil kedua algoritma kemudian dianalisis secara deskriptif untuk menentukan algoritma yang lebih sesuai diterapkan pada optimasi jaringan distribusi listrik. Secara garis besar, tahapan pelaksanaan penelitian dinyatakan sebagai berikut :

Perbandingan Algoritma Kruskal dan Prim dalam Menentukan Minimum Spanning Tree untuk Optimasi Jaringan Distribusi Listrik



HASIL PENELITIAN DAN PEMBAHASAN

Berikut adalah perbandingan antara algoritma Prim dan Kruskal dalam bentuk flowchart yang menggambarkan langkah-langkah masing-masing algoritma dalam menemukan Minimum Spanning Tree (MST) dalam sebuah graf berbobot (Fitriana, 2014) (Niluminda & EMUSB, 2022).

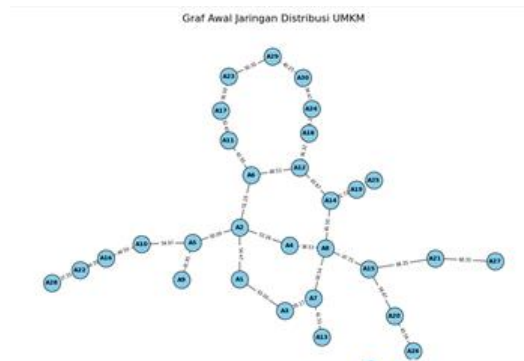


Gambar 1. Perbandingan Algoritma Kruskal dan Prim

Graf awal pada Gambar 1 merupakan representasi jaringan kabel distribusi listrik PLN UP3 Cimahi yang dibangun berdasarkan data titik-titik jaringan dan jarak antar titik yang saling terhubung. Setiap simpul (vertex) merepresentasikan titik pada jaringan distribusi listrik, seperti gardu distribusi, titik percabangan, atau titik sambung kabel, sedangkan sisi (edge) menunjukkan hubungan langsung berupa jaringan kabel yang menghubungkan dua titik tersebut. Bobot pada setiap

Perbandingan Algoritma Kruskal dan Prim dalam Menentukan Minimum Spanning Tree untuk Optimasi Jaringan Distribusi Listrik

sisi menyatakan panjang kabel antar titik dalam satuan meter sesuai dengan data penelitian.



Gambar 2. Graf Awal dari Jaringan Kabel Listrik PLN UP3 Cimahi

Graf awal ini menggambarkan keseluruhan struktur jaringan distribusi yang tersedia sebelum dilakukan proses optimasi. Pada kondisi awal, jaringan masih memuat seluruh hubungan antar titik tanpa mempertimbangkan efisiensi total panjang kabel. Akibatnya, terdapat beberapa jalur alternatif yang dapat menghubungkan titik-titik tertentu sehingga total panjang jaringan berpotensi lebih besar daripada yang diperlukan untuk mempertahankan keterhubungan seluruh jaringan.

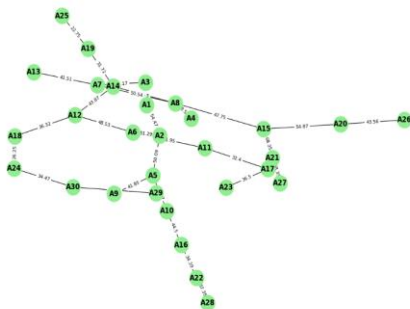
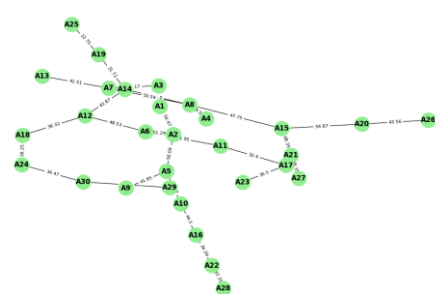
Pada penelitian ini, graf awal terdiri atas 30 simpul (A1–A30) yang saling terhubung melalui 32 sisi berbobot. Struktur jaringan menunjukkan bahwa beberapa simpul memiliki tingkat konektivitas yang lebih tinggi dan berfungsi sebagai titik percabangan utama, seperti A2, A8, A12, dan A15, karena terhubung dengan lebih dari dua titik lainnya. Sebaliknya, beberapa simpul berada pada ujung jaringan (leaf node), seperti A1, A9, A13, A25, A26, A27, dan A28, yang hanya memiliki satu hubungan langsung dengan simpul lain. Keberadaan beberapa jalur alternatif pada graf awal menyebabkan total panjang jaringan kabel belum tentu minimum, meskipun seluruh titik distribusi telah saling terhubung. Oleh karena itu, dilakukan proses optimasi menggunakan pendekatan Minimum Spanning Tree (MST) melalui Algoritma Prim dan Algoritma Kruskal. Kedua algoritma tersebut bertujuan memperoleh jaringan distribusi yang tetap menghubungkan seluruh titik pada sistem dengan total panjang kabel minimum tanpa membentuk siklus (cycle). Hasil optimasi ini diharapkan dapat memberikan alternatif rancangan jaringan yang

lebih efisien dari sisi kebutuhan panjang kabel serta mendukung perencanaan dan pengembangan jaringan distribusi listrik PLN.

Setelah graf awal berhasil dibentuk, tahap berikutnya adalah melakukan optimasi jaringan menggunakan dua algoritma Minimum Spanning Tree, yaitu Algoritma Prim dan Algoritma Kruskal. Kedua algoritma diterapkan pada graf yang sama sehingga hasil yang diperoleh dapat dibandingkan berdasarkan total bobot Minimum Spanning Tree, jumlah sisi yang dipilih, serta waktu komputasi. Perbandingan ini bertujuan untuk mengetahui algoritma yang memberikan efisiensi terbaik dalam membentuk jaringan distribusi kabel PLN.

Setelah dilakukan implementasi Algoritma Prim dan Algoritma Kruskal pada graf jaringan distribusi, langkah selanjutnya adalah membandingkan kinerja kedua algoritma tersebut. Perbandingan dilakukan berdasarkan beberapa aspek, yaitu total bobot Minimum Spanning Tree (MST) yang dihasilkan, jumlah sisi (edge) yang terpilih, serta karakteristik proses pemilihan sisi selama pembentukan MST. Hasil perbandingan kedua algoritma disajikan pada Tabel 1 berikut.

Tabel 1. Perbandingan Algoritma Kruskal dan Prim Berdasarkan Output Google Collab Phyton.

Hasil Algoritma Kruskal	Hasil Algoritma Prim
	
Sisi-sisi MST: A19 -- A25 = 22.75 A19 -- A14 = 31.72 A18 -- A24 = 26.25 A18 -- A12 = 36.32 A24 -- A30 = 34.47 A14 -- A12 = 43.87 A14 -- A8 = 45.5 A11 -- A17 = 32.4 A11 -- A6 = 41.95 A17 -- A23 = 36.5 A16 -- A22 = 34.39	Sisi-sisi MST: A19 -- A14 = 31.72 A19 -- A25 = 22.75 A18 -- A12 = 36.32 A18 -- A24 = 26.25 A24 -- A30 = 34.47 A14 -- A8 = 45.5 A14 -- A12 = 43.87 A11 -- A6 = 41.95 A11 -- A17 = 32.4 A17 -- A23 = 36.5 A16 -- A10 = 44.5

Perbandingan Algoritma Kruskal dan Prim dalam Menentukan Minimum Spanning Tree untuk Optimasi Jaringan Distribusi Listrik

Hasil Algoritma Kruskal	Hasil Algoritma Prim
A16 -- A10 = 44.5	A16 -- A22 = 34.39
A22 -- A28 = 37.35	A22 -- A28 = 37.35
A30 -- A29 = 40.25	A30 -- A29 = 40.25
A3 -- A7 = 35.17	A3 -- A7 = 35.17
A7 -- A13 = 41.51	A7 -- A8 = 50.54
A7 -- A8 = 50.54	A7 -- A13 = 41.51
A12 -- A6 = 48.53	A12 -- A6 = 48.53
A4 -- A8 = 38.53	A4 -- A8 = 38.53
A8 -- A15 = 47.75	A8 -- A15 = 47.75
A5 -- A9 = 41.85	A5 -- A2 = 50.09
A5 -- A2 = 50.09	A5 -- A9 = 41.85
A5 -- A10 = 54.97	A5 -- A10 = 54.97
A6 -- A2 = 51.29	A6 -- A2 = 51.29
A20 -- A26 = 43.56	A20 -- A15 = 54.87
A20 -- A15 = 54.87	A20 -- A26 = 43.56
A15 -- A21 = 68.35	A15 -- A21 = 68.35
A2 -- A1 = 54.47	A2 -- A1 = 54.47
A21 -- A27 = 68.35	A21 -- A27 = 68.35
Total Bobot MST : 1258.05	Total Bobot MST : 1258.05
Jumlah Edge : 29	Jumlah Edge : 29
Jumlah Node : 30	Jumlah Node : 30
Waktu Komputasi : 0.00128940 detik	Waktu Komputasi : 0.00082503 detik
Waktu Komputasi : 1.2894 ms	Waktu Komputasi : 0.8250 ms

Berdasarkan hasil penelitian dapat disimpulkan bahwa Algoritma Kruskal dan Algoritma Prim sama-sama mampu menghasilkan Minimum Spanning Tree (MST) yang optimal pada jaringan distribusi kabel listrik PLN UP3 Cimahi. Kedua algoritma menghasilkan jaringan dengan 29 sisi, 30 simpul, dan total bobot minimum sebesar 1258,05, sehingga tidak terdapat perbedaan dari sisi kualitas solusi yang dihasilkan. Namun demikian, dari aspek efisiensi komputasi,. Pada penelitian ini, Algoritma Prim menunjukkan kinerja yang lebih baik dengan waktu komputasi 0,8250 ms, lebih cepat dibandingkan Algoritma Kruskal yang memerlukan waktu 1,2894 ms. Namun, selisih waktu hanya sekitar 0,46 ms dan graf yang digunakan relatif kecil (30 simpul, 32 sisi), maka agar kesimpulan lebih kuat, pengukuran waktu komputasi sebaiknya dilakukan berulang (misalnya 100–1000 kali) untuk mendapatkan nilai rata-ratanya.. Berdasarkan analisis yang telah dilakukan, algoritma Kruskal dinilai lebih tepat digunakan dalam optimasi jaringan kabel listrik karena proses pengurutan bobot sisi dilakukan sejak awal, sehingga tahapan pemilihan jalur menjadi lebih sistematis dan efisien. Hasil tersebut sejalan dengan penelitian (Didiharyono & Soraya, 2018), yang menyatakan bahwa algoritma Prim dan Kruskal sama-sama mampu menghasilkan minimum spanning

tree melalui pendekatan greedy. Namun, pada jaringan listrik berbentuk jala dengan jumlah sisi relatif sedikit dan jumlah simpul yang banyak, algoritma Kruskal dinilai lebih efektif karena lebih mudah dalam proses seleksi sisi berdasarkan bobot terkecil (Julistia & Mardhotillah, 2025).

Secara umum, kedua algoritma memiliki tujuan yang sama, yaitu meminimalkan total panjang jaringan kabel, tetapi perbedaan mekanisme pemilihan sisi menyebabkan tingkat efisiensi proses menjadi berbeda. Oleh karena itu, pemilihan algoritma perlu disesuaikan dengan karakteristik jaringan yang digunakan agar proses optimasi dapat berjalan lebih optimal dan menghasilkan jalur distribusi yang lebih efisien.

Berdasarkan hasil di atas, total panjang jaringan sebelum optimasi adalah 1424,61 meter, sedangkan hasil Minimum Spanning Tree menggunakan algoritma Prim maupun Kruskal adalah 1258,05 meter. Dengan demikian diperoleh efisiensi optimasi sebesar 11,69%, yang menunjukkan bahwa penggunaan kedua algoritma mampu mengurangi kebutuhan panjang kabel tanpa mengurangi keterhubungan seluruh simpul pada jaringan.

Pada penelitian ini, algoritma Kruskal dinilai lebih praktis diterapkan pada jaringan distribusi listrik PLN UP3 Cimahi. Hal ini disebabkan proses pengurutan bobot dilakukan pada tahap awal sehingga proses seleksi sisi menjadi lebih sistematis dan mudah diimplementasikan menggunakan pustaka NetworkX pada Google Colab. Karakteristik tersebut sesuai dengan penelitian Julistia dan Mardhotillah (2025) yang menyatakan bahwa algoritma Kruskal lebih efektif diterapkan pada jaringan dengan jumlah simpul yang relatif banyak karena proses pemilihan sisi dilakukan berdasarkan urutan bobot terkecil.

Selain menghasilkan solusi optimasi, penelitian ini juga menunjukkan bahwa Google Colab merupakan lingkungan komputasi yang efektif untuk implementasi algoritma graf. Integrasi Python dengan pustaka NetworkX memungkinkan proses pembentukan graf, pencarian MST, visualisasi jaringan, dan perhitungan bobot dilakukan secara otomatis serta mudah direproduksi. Dengan demikian, penggunaan Google Colab tidak hanya meningkatkan efisiensi analisis, tetapi juga mendukung proses dokumentasi dan kolaborasi penelitian.

Meskipun demikian, penelitian ini masih memiliki keterbatasan karena hanya menggunakan satu parameter optimasi, yaitu panjang kabel jaringan. Pada implementasi nyata, optimasi jaringan distribusi listrik juga dipengaruhi oleh kapasitas penghantar, rugi-rugi daya (power loss), keandalan sistem, biaya investasi, serta pertumbuhan beban listrik. Oleh karena itu, hasil penelitian ini lebih tepat dipandang sebagai model optimasi topologi jaringan daripada optimasi sistem kelistrikan secara menyeluruh.

Penelitian ini memperkuat penerapan teori graf, khususnya konsep Minimum Spanning Tree, sebagai salah satu pendekatan matematis yang efektif untuk menyelesaikan permasalahan optimasi jaringan. Hasil penelitian menunjukkan bahwa algoritma Prim dan Kruskal mampu menghasilkan solusi optimum dengan total bobot yang sama, sehingga mendukung teori bahwa kedua algoritma greedy tersebut memiliki kemampuan yang setara dalam memperoleh Minimum Spanning Tree pada graf berbobot.

Dari sisi praktis, hasil penelitian memberikan alternatif bagi pengelola jaringan distribusi listrik dalam merancang jaringan yang lebih efisien. Pengurangan panjang kabel sebesar 11,69% menunjukkan adanya potensi penghematan biaya pembangunan maupun pemeliharaan jaringan tanpa mengurangi keterhubungan seluruh titik distribusi. Selain itu, implementasi menggunakan Google Colab memungkinkan proses analisis dilakukan secara cepat, mudah direplikasi, dan tidak memerlukan perangkat lunak berlisensi.

SIMPULAN

Berdasarkan hasil pembahasan di atas, dapat disimpulkan bahwa algoritma Kruskal lebih cocok untuk mengoptimalkan jaringan kabel listrik. Hal ini disebabkan oleh fakta bahwa proses pengurutan bobot sisi dilakukan sejak awal, sehingga fase pemilihan jalur menjadi lebih sistematis, langsung, dan efisien. Penggunaan Google Colab untuk Python juga memfasilitasi perhitungan dan visualisasi algoritma yang lebih cepat, sehingga menyederhanakan analisis pohon rentang minimum. Algoritma Prim dan Kruskal sama-sama menerapkan metode greedy dalam membentuk minimum spanning tree. Hasil optimasi menggunakan Algoritma Prim dan Algoritma Kruskal menghasilkan jaringan dengan efisiensi

panjang kabel sebesar 11,69% dibandingkan jaringan awal. Efisiensi tersebut menunjukkan bahwa jaringan dapat dirancang lebih optimal dengan tetap mempertahankan seluruh keterhubungan titik distribusi, sehingga berpotensi mengurangi kebutuhan material kabel serta biaya pembangunan dan pemeliharaan jaringan.

Secara teoritis, Algoritma Kruskal sering dinilai efektif pada graf yang memiliki jumlah sisi relatif sedikit dibandingkan jumlah simpul karena proses pemilihan sisi dilakukan berdasarkan urutan bobot terkecil. Namun, pada jaringan distribusi kabel listrik PLN UP3 Cimahi yang dianalisis dalam penelitian ini, Algoritma Prim menunjukkan waktu komputasi yang lebih rendah, meskipun kedua algoritma menghasilkan MST dengan total bobot yang sama. Temuan ini menunjukkan bahwa pemilihan algoritma sebaiknya didasarkan pada karakteristik jaringan dan hasil evaluasi kinerja pada kasus yang dianalisis.

Selain memberikan kontribusi pada optimasi jaringan distribusi listrik, implementasi algoritma Prim dan Kruskal menggunakan Google Colab juga memiliki implikasi dalam pembelajaran matematika dan komputasi. Visualisasi graf serta implementasi algoritma melalui Python dapat membantu mahasiswa memahami konsep teori graf, Minimum Spanning Tree, dan algoritma greedy secara lebih interaktif. Dengan demikian, Google Colab tidak hanya berfungsi sebagai alat komputasi, tetapi juga sebagai media pembelajaran yang mendukung penguasaan konsep matematika terapan dan pemrograman.

Diharapkan pada penelitian selanjutnya dapat memperluas model optimasi dengan memasukkan beberapa kriteria sekaligus, seperti biaya instalasi, kapasitas penghantar, rugi-rugi daya, dan keandalan jaringan sehingga diperoleh model optimasi multiobjektif. Selain itu, perbandingan dapat dikembangkan dengan melibatkan algoritma lain, seperti Borůvka, Dijkstra (untuk permasalahan lintasan terpendek), atau pendekatan metaheuristik seperti Genetic Algorithm dan Ant Colony Optimization, sehingga dapat diketahui algoritma yang paling sesuai untuk karakteristik jaringan distribusi listrik yang lebih kompleks..

DAFTAR PUSTAKA

- Abhisyeka, B., Bagaskara, J. A., Valensy, I., Permana, R. H. P., & Suroni, A. (2025). Minimum Spanning Tree Approach for Optimization and Clustering: Algorithms, Applications, and Comparisons. *Journal of Advanced Systems Intelligence and Cybersecurity*, 1(01).
- Alham, N. R., Budiasih, A. F., Utomo, R. M., Muslimin, M., & Wirawan, A. P. (2024). Studi Konservasi Energi Pada Beban Rumah Tangga Dalam Upaya Efisiensi Penggunaan Energi Listrik. *JTEIN: Jurnal Teknik Elektro Indonesia*, 5(2), 309–320.
- Didiharyono, D., & Soraya, S. (2018). Penerapan Algoritma Greedy Dalam Menentukan Minimum Spanning Trees Pada Optimisasi Jaringan Listrik Jala. *Jurnal VARIAN*, 1(2), 1–10. <https://doi.org/10.30812/varian.v1i2.66>
- Fitriana, R. (2014). Perbandingan Algoritma Prim dan Kruskal untuk Menyelesaikan Masalah Spanning Tree. *Procedia Manufacturing*, 1(22 Jan), 1–17.
- Julistia, A., & Mardhotillah, B. (2025). Pengoptimalan Pendistribusian Jaringan Listrik Menggunakan Kruskal Algorithm. *JISTech (Journal of Islamic Science and Technology)*, 10(1), 64–71.
- Munir, R. (2016). *Matematika Diskrit* (Edisi Ketiga. Informatika).
- Niluminda, K. P. O., & EMUSB, E. (2022). An efficient method to solve minimum spanning tree problem using graph theory and improved ant colony optimization algorithm. *North American Academic Research*, 5(12), 34–43.
- Nugraha, D. W. (2011). Aplikasi algoritma prim untuk menentukan minimum spanning tree suatu graf berbobot dengan menggunakan pemrograman berorientasi objek. *Foristek*, 1(2).
- Nurlinawati Dili, Y., Wulan, E. ., & Ilahi, F. (2021). Penyelesaian Masalah Transportasi untuk Mencari Solusi Optimal dengan Pendekatan Minimum Spanning Tree (MST) Menggunakan Algoritma. *Academia.Edu*.
- Rofiq, N., & Sitio, S. L. M. (2024). *Pengenalan Dasar Analisis Data dengan Python di Google Colab*.
- Ruhimat, Q. A. A., Slamini, S., & Malinda, A. (2024). Efektivitas Algoritma Kruskal dalam Mengoptimalkan Jalur Terpendek pada Jaringan Intranet. *Jurnal Sains Natural*, 2(3), 59–67.
- Sembiring, R. R. (2022). Penerapan Algoritma Prim dalam Menentukan Minimum Spanning Tree (MST)(Studi Kasus: Jaringan Pipa PDAM Tirta Muaro Jambi). *Universitas Jambi*.
- Simbolon, A. F. A. (2021). *Aplikasi Minimum Spanning Tree pada Jaringan Listrik di Desa Simatorkis Kecamatan Dolok Kabupaten Paluta*. Unimed.
- Situmorang, Y. M., & Mansyur, A. (2023). Pengoptimalan Jaringan Pipa Primer PDAM Tirtanadi Cabang Tuasan Dengan Menggunakan Algoritma Kruskal. *Jurnal Riset Rumpun Matematika Dan Ilmu Pengetahuan Alam*, 2(2), 221–237.
- Wamiliana, W., Usman, M., Warsito, W., Warsono, W., & Daoud, J. I. (2020). Using Modification of Prim's Algorithm and Gnu Octave and to Solve the Multiperiods Installation Problem. *IIUM Engineering Journal*, 21(1), 100–112.