



# Implementasi Algoritma Kruskal untuk Menentukan Minimum Spanning Tree Rute Tempat Pembuangan Sampah Berbasis GUI

Moch. Anang Karyawan<sup>1,\*</sup>, Yusuf Effendi<sup>2</sup>, Muhammad Zakariya Alif Ridwan<sup>2</sup>, Maulana Muhammad Adityawarman<sup>2</sup>, Muhammad Khairul Ikhwan<sup>2</sup>

<sup>1</sup> Fakultas Teknik dan Desain, Program Studi Sistem Informasi, Universitas Hayam Wuruk Perbanas, Surabaya, Indonesia

<sup>2</sup> Fakultas Teknik dan Desain, Program Studi Informatika, Universitas Hayam Wuruk Perbanas, Surabaya, Indonesia

Email: <sup>1,\*</sup>[anang.karyawan@perbanas.ac.id](mailto:anang.karyawan@perbanas.ac.id), <sup>2</sup>[yusuf@perbanas.ac.id](mailto:yusuf@perbanas.ac.id), <sup>3</sup>[202202011003@mhs.hayamwuruk.ac.id](mailto:202202011003@mhs.hayamwuruk.ac.id),

<sup>4</sup>[202202011013@mhs.hayamwuruk.ac.id](mailto:202202011013@mhs.hayamwuruk.ac.id), <sup>5</sup>[202202011012@mhs.hayamwuruk.ac.id](mailto:202202011012@mhs.hayamwuruk.ac.id)

Email Penulis Korespondensi: [anang.karyawan@perbanas.ac.id](mailto:anang.karyawan@perbanas.ac.id)

**Abstrak**—Pengelolaan sampah di kawasan perkotaan seperti Kota Surabaya menghadapi tantangan ganda berupa volume sampah yang terus meningkat dan efisiensi operasional. Produksi sampah harian di Surabaya yang mencapai kurang lebih 1.800 ton memerlukan sistem logistik yang presisi untuk menekan biaya transportasi yang menyerap 60% hingga 85% dari total anggaran manajemen persampahan daerah. Penelitian ini bertujuan mengoptimalkan jarak rute pengangkutan sampah di wilayah Surabaya Timur yang termasuk dalam kawasan yang terus tumbuh. Implementasi algoritma Kruskal digunakan untuk menentukan *Minimum Spanning Tree* (MST) dari jaringan Tempat Pembuangan Sampah (TPS). Model pengembangan perangkat lunak *Waterfall* digunakan untuk mengembangkan aplikasi berbasis *Java Graphical User Interface* (GUI) yang memvisualisasikan rute optimal dengan total jarak minimal. Data jarak antar enam TPS diperoleh dari pemetaan jarak jalan aktual dan dimodelkan sebagai graf berbobot. Jumlah TPS yang terbatas pada enam lokasi dipilih sebagai studi kasus untuk memudahkan pemodelan awal, sehingga hasil penelitian ini memiliki keterbatasan dalam generalisasi namun tetap relevan untuk menunjukkan potensi optimasi rute. Hasil penelitian menunjukkan bahwa algoritma Kruskal menghasilkan struktur MST dengan total jarak minimum sebesar 11 km. Aplikasi yang dikembangkan telah diuji dan terbukti mampu membantu perencanaan rute pengangkutan sampah serta menyajikan visualisasi grafis yang mudah dipahami oleh pengguna. Penelitian ini diharapkan dapat mendukung pengambilan keputusan dalam optimasi logistik pengangkutan sampah perkotaan.

**Kata Kunci:** Algoritma Kruskal; Java GUI; Minimum Spanning Tree; Tempat Pembuangan Sampah (TPS); Rute Optimal

**Abstract**—Waste management in urban areas such as Surabaya faces dual challenges of increasing waste volume and operational efficiency. The city's daily waste production of approximately 1,800 tons requires a precise logistics system to reduce transportation costs, which account for 60% to 85% of the total municipal waste management budget. This study aims to optimize waste collection routes in East Surabaya, a rapidly growing area. Kruskal's algorithm was implemented to determine the Minimum Spanning Tree (MST) of the waste disposal network. The Waterfall software development model was applied to build a Java-based Graphical User Interface (GUI) application that visualizes optimal routes with minimal total distance. Distance data among six waste disposal sites (TPS) were obtained from actual road mapping and modeled as a weighted graph. The selection of only six TPS was intended as a case study to simplify the initial modeling process, which presents certain limitations in generalization but remains relevant for demonstrating route optimization potential. The results show that Kruskal's algorithm produced an MST structure with a minimum total distance of 11 km. The developed application was tested and proved effective in supporting route planning and providing user-friendly graphical visualization. This research is expected to contribute to decision-making in optimizing urban waste collection logistics.

**Keywords:** Kruskal's Algorithm; Java GUI; Minimum Spanning Tree; Waste Disposal Sites (TPS), Optimal Route

## 1. PENDAHULUAN

Pengelolaan sampah perkotaan merupakan tantangan besar bagi kota-kota besar di Indonesia, termasuk Kota Surabaya. Peningkatan jumlah penduduk dan aktivitas ekonomi menyebabkan volume sampah terus bertambah setiap hari. Data menunjukkan bahwa Kota Surabaya menghasilkan sekitar 1.800 ton sampah per hari[1]–[3], sehingga memerlukan sistem pengelolaan yang efisien dan berkelanjutan. Wilayah Surabaya Timur adalah kawasan pemukiman di Kota Surabaya yang terus tumbuh[4] dan termasuk kawasan dengan densitas penduduk yang tinggi[5], [6]. Salah satu permasalahan yang muncul pada kawasan yang tumbuh dan densitas tinggi adalah permasalahan logistik berupa rute pengangkutan sampah[7]. Rute yang tidak terstruktur menyebabkan perjalanan berulang, peningkatan konsumsi bahan bakar, serta waktu tempuh yang lebih lama berakibat pada peningkatan emisi karbon[8]. Beberapa studi menunjukkan bahwa biaya pengumpulan dan transportasi sampah di perkotaan sering kali menyerap 60% hingga 85% dari total anggaran manajemen persampahan daerah[2]. Optimalisasi rute pengangkutan menjadi strategi kunci dalam menurunkan beban keuangan dan dampak lingkungan. Beberapa studi menunjukkan bahwa penerapan algoritma optimasi dan sistem cerdas dapat mengurangi jarak tempuh dan biaya operasional secara signifikan[9][10].

Permasalahan optimasi rute pengumpulan dapat dimodelkan menggunakan teori graf melalui pendekatan *Minimum Spanning Tree* (MST)[11]. Pemodelan ini bertujuan menghubungkan seluruh simpul dari Tempat Pembuangan Sementara (TPS) atau Tempat Pembuangan Akhir (TPA) dengan total bobot sisi minimum tanpa siklus[12]. Di antara algoritma klasik untuk MST, Kruskal menonjol sebagai metode *greedy*[12] yang efisien terutama pada graf yang bersifat *sparse* (jarang), seperti jaringan jalan antar TPS[13]. Algoritma ini bekerja dengan mengurutkan sisi berdasarkan bobot dan menambahkan sisi terkecil selama tidak membentuk siklus (menggunkan struktur *union-find*)[12].

Terdapat beberapa penelitian terdahulu yang mengaplikasikan algoritma Kruskal maupun pendekatan MST dalam perencanaan rute transportasi[10][14]–[16], distribusi barang[17], [18], serta pengelolaan sampah perkotaan[19].



Penerapan algoritma Kruskal pada perencanaan rute bus, distribusi logistik hingga jaringan transportasi kota menunjukkan bahwa algoritma ini mampu menghasilkan solusi rute yang efisien dengan kompleksitas komputasi yang rendah[20], [21]. Studi yang menerapkan pendekatan MST/Kruskal untuk menyelesaikan masalah transportasi dan distribusi barang tersebut relevan secara metodologis dengan permasalahan pengumpulan sampah. Studi yang mengaplikasikan algoritma MST untuk menyusun rute pengumpulan dan penjadwalan truk sampah, termasuk studi implementasi Kruskal pada kasus kota/kawasan lokal yang menunjukkan manfaat praktis dalam mengurangi jarak dan waktu tempuh. Studi lain yang membandingkan kinerja algoritma Kruskal dengan algoritma MST lainnya seperti Prim menemukan bahwa Kruskal memiliki keunggulan pada kasus graf tertentu[22]. Selain menggunakan algoritma optimasi, terdapat pula implementasi cerdas menggunakan *Machine Learning*[8], [23]. Implementasi sistem cerdas berbasis algoritma optimasi secara umum terbukti mampu mengurangi jarak pengumpulan rata-rata sebesar 21,51%[24].

Sebagian besar implementasi sistem cerdas berbasis algoritma optimasi sebelumnya masih berfokus pada analisis teoritis, simulasi statis, atau implementasi algoritma tanpa dukungan antarmuka interaktif yang mudah digunakan oleh praktisi lapangan. Implementasi sistem optimasi rute yang bersifat aplikatif dan dilengkapi dengan visualisasi grafis masih relatif terbatas, khususnya pada studi kasus pengangkutan sampah di wilayah perkotaan Indonesia. Keberadaan antarmuka visual sangat penting untuk membantu pengelola kebersihan dalam memahami struktur jaringan dan mengambil keputusan operasional secara cepat dan akurat. Hal ini sejalan dengan dengan program *Green Sister City* antara Surabaya dan Kitakyushu[25] yang menitikberatkan pada penguatan infrastruktur[26]. Penguatan infrastruktur hilir harus didukung oleh sistem logistik hulu yang presisi agar aliran limbah menuju TPA Benowo berjalan tanpa hambatan.

Penelitian ini bertujuan untuk mengembangkan aplikasi penerapan algoritma Kruskal untuk menentukan rute MST pada jaringan TPS di wilayah Surabaya Timur berbasis *Graphical User Interface* (GUI) menggunakan bahasa pemrograman Java. Aplikasi ini dirancang untuk memungkinkan pengguna memasukkan data lokasi dan jarak secara fleksibel serta menampilkan hasil optimasi rute dalam bentuk visual grafis. Jumlah TPS yang terbatas pada enam lokasi dipilih sebagai studi kasus untuk memudahkan pemodelan awal, sehingga hasil penelitian ini memiliki keterbatasan dalam generalisasi namun tetap relevan untuk menunjukkan potensi optimasi rute. Penelitian ini diharapkan dapat memberikan kontribusi praktis berupa alat bantu pengambilan keputusan logistik serta kontribusi akademik dalam penerapan algoritma graf pada manajemen sampah perkotaan.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini menggunakan model pengembangan *Waterfall*[27][28] yang menyediakan alur hidup perangkat lunak secara sekuensial[29]. Tahapan utama yang diterapkan meliputi: (1) analisis kebutuhan, (2) desain sistem, (3) pengkodean, (4) pengujian, dan (5) implementasi. Model *Waterfall* dipilih karena spesifikasi sistem optimasi rute relatif jelas dan stabil (input: data lokasi; proses: algoritma Kruskal; output: rute optimal), sehingga pendekatan sekuensial memudahkan verifikasi tiap tahapan sebelum melanjutkan ke tahap berikutnya.

### 2.2 Pengumpulan Data

Data yang dikumpulkan meliputi koordinat geografis dan jarak jalan aktual (*road distance*) antar enam TPS utama di wilayah Surabaya Timur. Pengukuran jarak dilakukan menggunakan aplikasi pemetaan digital dan diverifikasi melalui pengecekan lapangan sederhana bila memungkinkan. Data kemudian dimodelkan sebagai graf berbobot  $G = (V, E)$ , di mana  $V$  merepresentasikan simpul (TPS) dan  $E$  merepresentasikan sisi dengan bobot berupa jarak jalan aktual.

### 2.3 Desain dan Implementasi Algoritma

Aplikasi dikembangkan menggunakan Java pada Eclipse IDE dengan antarmuka GUI. Fitur utama aplikasi meliputi: input data lokasi/jarak, pembentukan graf berbobot, eksekusi algoritma Kruskal untuk menentukan MST, serta visualisasi grafis hasil rute. Implementasi GUI dirancang agar pengguna dapat memasukkan atau mengimpor data jarak secara fleksibel dan melihat hasil optimasi secara interaktif.

Langkah implementasi algoritma Kruskal:

- Identifikasi seluruh sisi dan bobot graf berdasarkan jarak antar TPS.
- Pengurutan sisi berdasarkan bobot terkecil (*sorting*).
- Penerapan struktur data Union-Find untuk deteksi dan pencegahan siklus.
- Iterasi penambahan sisi hingga seluruh simpul terhubung (jumlah sisi =  $|V|-1$ ).

### 2.4 Objek Penelitian

Objek penelitian terdiri dari enam TPS di wilayah Surabaya Timur yang dipilih berdasarkan kriteria aksesibilitas, representativitas kawasan pemukiman, dan ketersediaan data koordinat. Setiap objek dicatat atribut berikut: nama TPS dan jarak jalan ke TPS lain (meter/km). Data ini memberikan gambaran ruang lingkup studi yang bersifat studi kasus skala kecil dan memungkinkan pengujian fungsional aplikasi serta validasi awal algoritma. Pemilihan enam TPS sebagai studi kasus dimaksudkan untuk memfasilitasi pemodelan awal dan validasi implementasi aplikasi dalam kondisi terkontrol; oleh karena itu hasil yang diperoleh bersifat indikatif dan perlu diuji ulang pada jaringan TPS yang lebih besar untuk generalisasi.

## 2.5 Pengujian dan Evaluasi Sistem

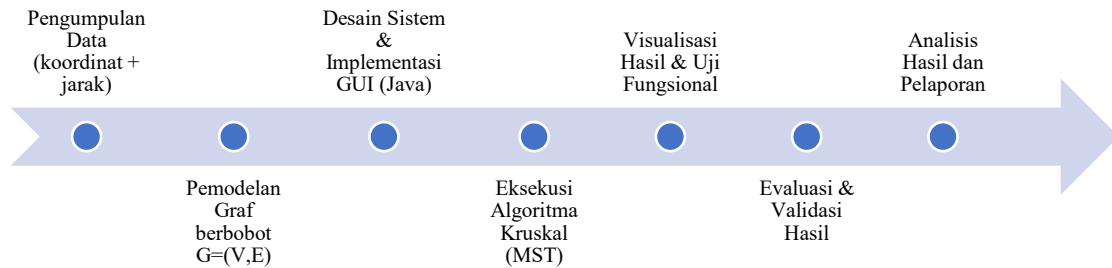
Pengujian sistem dilakukan hanya pada uji fungsional (verifikasi bahwa fitur input, pemrosesan Kruskal, dan visualisasi bekerja sesuai spesifikasi). Evaluasi meliputi: total jarak MST yang dihasilkan dan hasil visualisasi.

## 2.6 Validasi dan Analisis Hasil

Validasi hasil optimasi dilakukan dengan membandingkan total jarak MST yang dihasilkan aplikasi dengan perhitungan manual/alternatif sederhana.

## 2.7 Diagram Alur Penelitian

Diagram tahapan penelitian yang menggambarkan langkah-langkah utama penelitian dapat dilihat pada Gambar 1.



Gambar 1. Diagram Tahapan Penelitian

## 3. HASIL DAN PEMBAHASAN

### 3.1 Inisialisasi Simpul dan Bobot Graf

Penelitian ini memodelkan keenam lokasi TPS di Surabaya Timur sebagai simpul (*node*) dan jarak antar lokasi diukur dan direpresentasikan sebagai bobot sisi (*edge*) pada graf. Tabel 1 menyajikan inisialisasi simpul (V1–V6) beserta nama TPS yang menjadi titik pengamatan dalam studi ini. Keenam TPS dipilih untuk mewakili sebaran spasial di Surabaya Timur sehingga model graf dapat mencerminkan kondisi operasional pengumpulan sampah lokal. Pemilihan lokasi mempertimbangkan akses jalan utama dan konsentrasi timbulan sampah sehingga hasil optimasi rute relevan untuk praktik operasional.

Tabel 1. Daftar TPS di Surabaya Timur

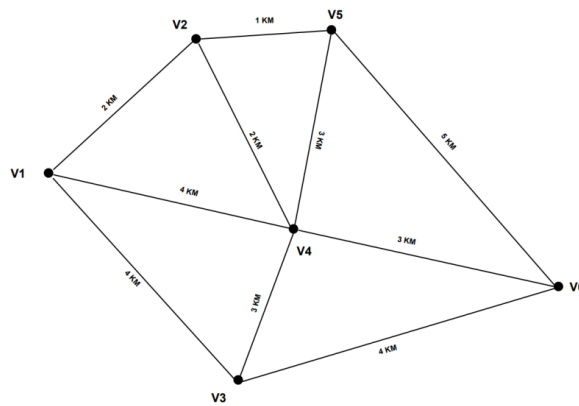
Inisialisasi Simpul (Vi)	Nama TPS Surabaya Timur
V1	TPS Rungkut Alang Alang
V2	TPS Kendal Sari
V3	TPS Purimas
V4	TPS Medokan Ayu
V5	TPS Wonorejo
V6	TPS 3R Gunung Raya

Gambar 2 memperlihatkan peta lokasi keenam TPS sehingga pembaca dapat memahami sebaran spasial simpul yang dimodelkan. Dari peta terlihat bahwa TPS tersebar di area permukiman dan koridor jalan utama; pola sebaran ini memengaruhi konektivitas graf dan pilihan jalur yang mungkin optimal.

Gambar 3 menampilkan graf komplit yang menghubungkan seluruh simpul dengan sisi-sisi yang mewakili jarak jalan aktual antar TPS. Graf komplit ini menjadi basis untuk proses pengurutan sisi pada algoritma Kruskal; setiap sisi pada graf mewakili alternatif rute yang dapat dipertimbangkan dalam pembentukan MST.



Gambar 2. Lokasi Geografis TPS di Wilayah Surabaya Timur



Gambar 3. Model Graf Awal Jaringan TPS

### 3.2 Bobot Sisi pada Graf Jaringan TPS

Tabel 2 memuat nilai bobot (dalam kilometer) untuk setiap pasangan simpul yang diukur dari jarak jalan aktual. Nilai bobot berkisar dari 1 km hingga 5 km dan menentukan prioritas pemilihan sisi oleh algoritma Kruskal. Perbedaan bobot ini mencerminkan variasi jarak antar TPS yang akan memengaruhi total jarak MST dan efisiensi rute operasional.

Tabel 2. Bobot (Jarak) Sisi pada Graf Jaringan TPS

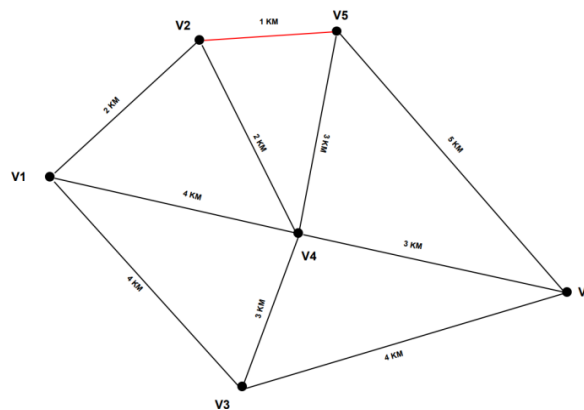
Sisi	Bobot (km)
V1 – V2	2
V2 – V4	2
V1 – V4	4
V1 – V3	4
V3 – V4	3
V3 – V6	4
V4 – V6	3
V4 - V5	3
V2 – V5	1
V5 – V6	5

### 3.3 Analisis Pembentukan MST (Langkah demi Langkah)

Proses algoritma Kruskal dimulai dengan mengurutkan seluruh sisi dari bobot terkecil. Tahapan pembentukan MST dijelaskan melalui proses iteratif algoritma Kruskal, ditambah dengan rujukan ke gambar langkah-langkah untuk memudahkan verifikasi visual.

a. Langkah 1 - Gambar 4

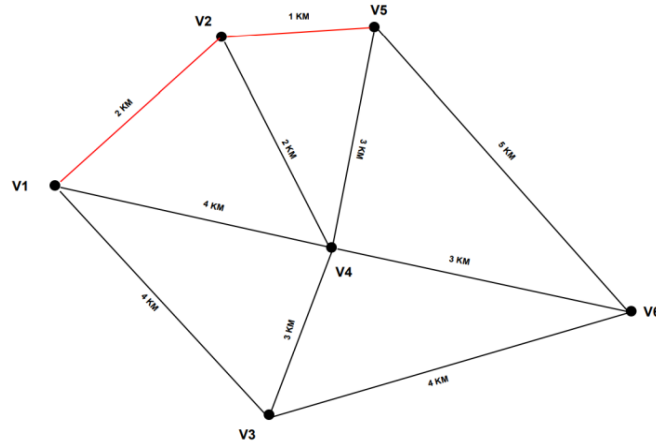
Gambar 4 menggambarkan kondisi graf setelah pemilihan sisi pertama oleh algoritma Kruskal. Algoritma memilih sisi (V2, V5) dengan bobot 1 km sebagai sisi pertama karena merupakan jarak terpendek; penambahan sisi ini menghubungkan dua simpul awal tanpa membentuk siklus.



Gambar 4. Graf Langkah 1

b. Langkah 2 - Gambar 5

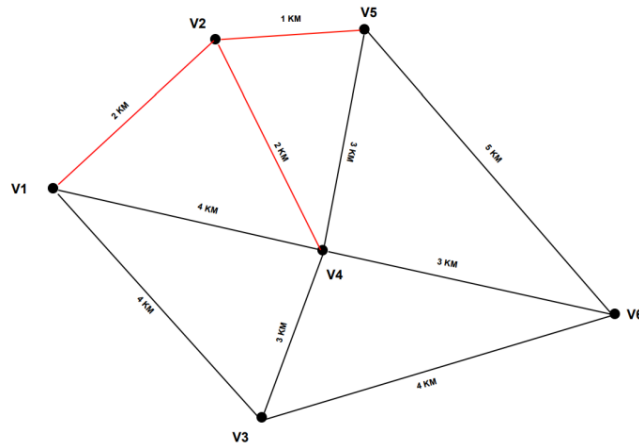
Gambar 5 menunjukkan graf setelah penambahan sisi kedua. Sisi (V1, V2) dengan bobot 2 km ditambahkan untuk memperluas komponen terhubung, mengikuti urutan bobot terkecil dan tetap menghindari siklus.



Gambar 5. Graf Langkah 2

c. Langkah 3 - Gambar 6

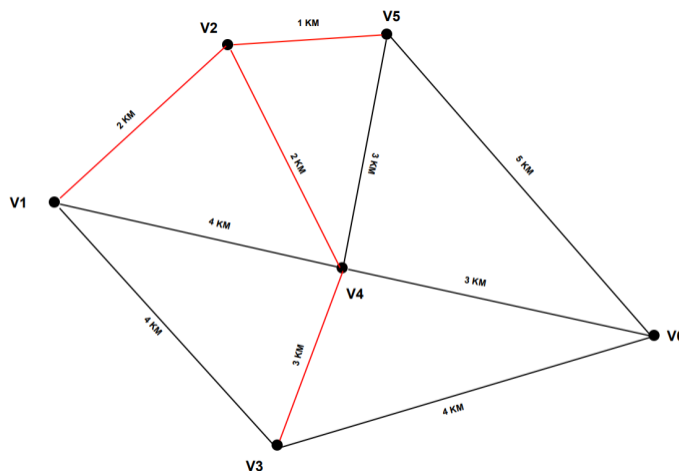
Gambar 6 memperlihatkan kondisi graf setelah penambahan sisi ketiga. Sisi (V2, V4) dengan bobot 2 km dimasukkan karena tidak membentuk siklus dan membantu menghubungkan simpul-simpul yang tersisa.



Gambar 6. Graf Langkah 3

d. Langkah 4 - Gambar 7

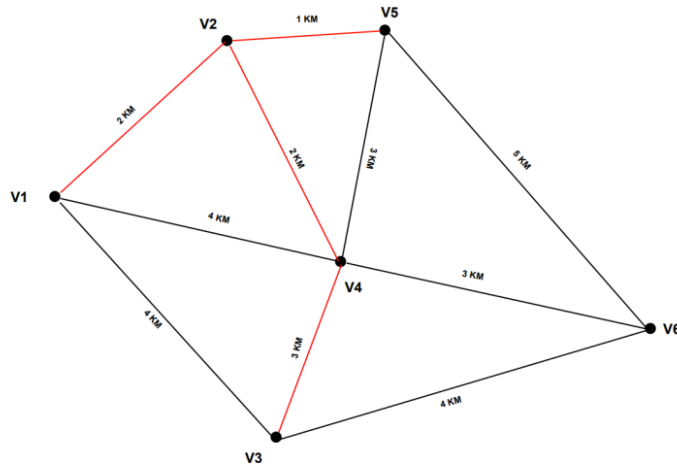
Gambar 7 menampilkan graf setelah penambahan sisi keempat. Sisi (V3, V4) dengan bobot 3 km dipilih untuk menghubungkan V3 ke komponen utama sehingga mengurangi jumlah simpul yang belum terhubung.



Gambar 7. Graf Langkah 4

e. Langkah 5 — Gambar 8

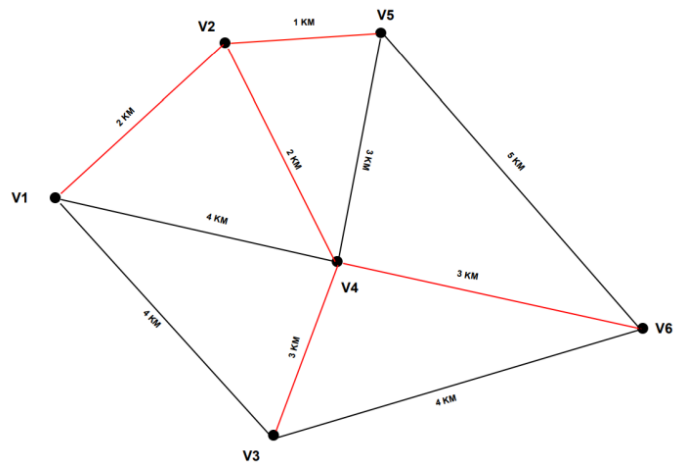
Gambar 8 menunjukkan contoh sisi yang ditolak karena akan membentuk siklus. Sisi (V4, V5) dengan bobot 3 km ditolak karena penambahannya akan membentuk siklus dengan jalur yang sudah ada (V4–V2–V5); penolakan ini menjaga sifat MST tanpa siklus.



Gambar 8. Graf Langkah 5

f. Langkah 6 - Gambar 9

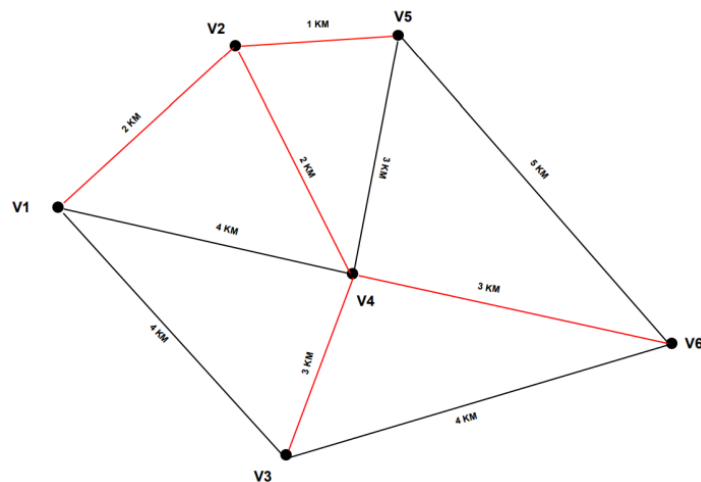
Gambar 9 memperlihatkan graf setelah penambahan sisi yang menghubungkan simpul terakhir. Sisi (V4, V6) dengan bobot 3 km dipilih untuk menyelesaikan konektivitas seluruh simpul; setelah langkah ini semua simpul terhubung dalam satu komponen.



Gambar 9. Graf Langkah 6

g. Langkah 7 - Gambar 10

Gambar 10 memperlihatkan kondisi graf setelah algoritma mempertimbangkan sisi berikutnya dengan bobot sedang. Sisi (V1, V3) = 4 km. Meskipun memiliki jarak yang lebih panjang, sisi ini tidak dapat ditambahkan karena penambahannya akan membentuk siklus dengan sisi-sisi yang sudah ada; oleh karena itu Kruskal menolak sisi ini untuk menjaga sifat pohon tanpa siklus.

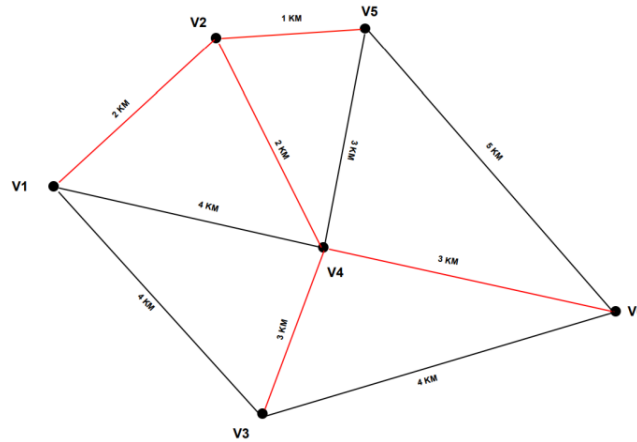


Gambar 10. Graf Langkah 7



h. Langkah 8 - Gambar 11

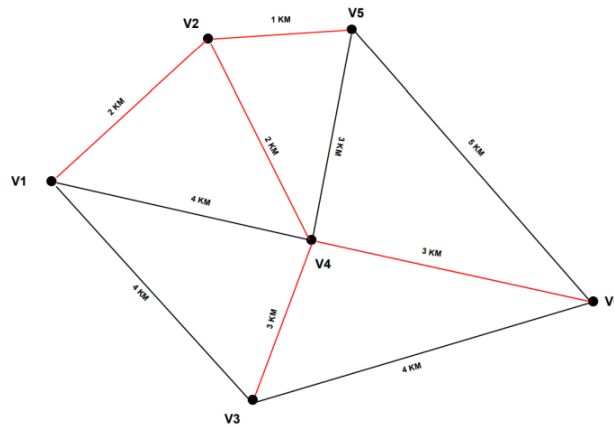
Gambar 11 menampilkan graf pada tahap verifikasi setelah beberapa sisi ditolak. Sisi  $(V1, V4) = 4$  km. Sama seperti sebelumnya, sisi ini tidak dapat ditambahkan karena memasukkannya ke dalam graf akan menghasilkan siklus; keputusan penolakan ini memastikan total bobot tetap minimal tanpa redundansi jalur.



Gambar 11. Graf Langkah 8

i. Langkah 9 - Gambar 12

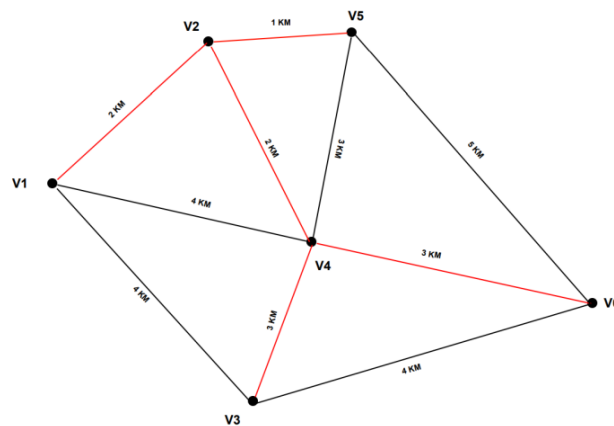
Gambar 12 menggambarkan kondisi graf saat algoritma mengevaluasi sisi dengan bobot menengah-tinggi. Sisi  $(V3, V6) = 4$  km. Sisi ini tidak dapat ditambahkan karena keberadaannya akan membentuk siklus dengan jalur yang sudah terpilih; oleh sebab itu Kruskal melewati sisi ini dan melanjutkan ke sisi berikutnya.



Gambar 12. Graf Langkah 9

j. Langkah 10 - Gambar 13

Gambar 13 menunjukkan graf setelah penolakan beberapa sisi yang berpotensi membentuk siklus. Sisi  $(V5, V6) = 5$  km. Sisi ini tidak dapat ditambahkan karena penambahannya akan membentuk siklus pada struktur graf saat ini; penolakan terhadap sisi ini mendorong pemilihan sisi alternatif yang mempertahankan konektivitas tanpa siklus.



Gambar 13. Graf Langkah 10



Gambar 14 menyajikan struktur graf akhir setelah seluruh iterasi Kruskal selesai dan semua sisi yang menyebabkan siklus ditolak. Setelah menolak sisi-sisi yang membentuk siklus pada langkah sebelumnya, algoritma membentuk struktur MST akhir yang menghubungkan seluruh simpul tanpa siklus. Sisi-sisi yang termasuk dalam MST adalah:

$$(V2, V5) = 1 \text{ km}$$

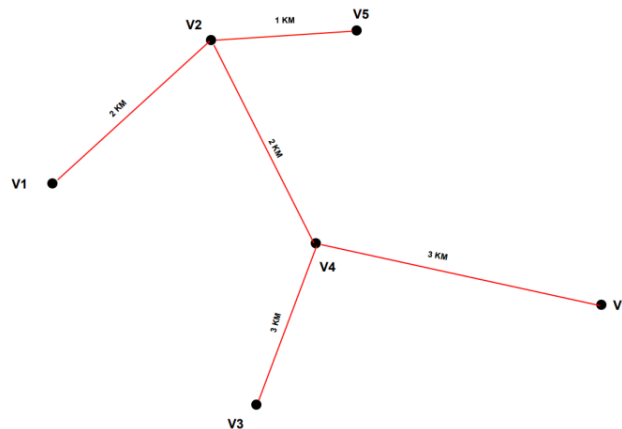
$$(V1, V2) = 2 \text{ km}$$

$$(V2, V4) = 2 \text{ km}$$

$$(V3, V4) = 3 \text{ km}$$

$$(V4, V6) = 3 \text{ km.}$$

Total jarak minimum yang diperoleh untuk menghubungkan keenam TPS tersebut adalah 11 km. Struktur pada Gambar 14 menegaskan bahwa penolakan terhadap sisi berjarak lebih panjang atau yang membentuk siklus merupakan langkah krusial untuk mencapai solusi rute optimal. Total jarak minimum yang diperoleh adalah 11 km, yang menunjukkan penghematan signifikan dibandingkan rute konvensional yang tidak terstruktur.

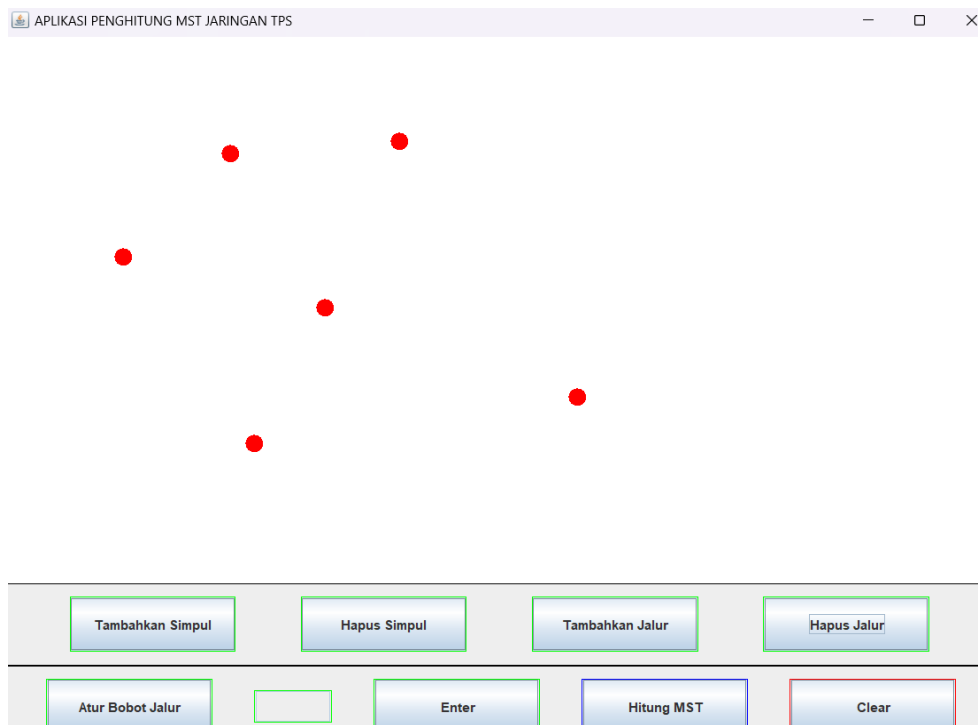


**Gambar 14.** Struktur Rute Optimal (MST) yang Menghubungkan Seluruh TPS

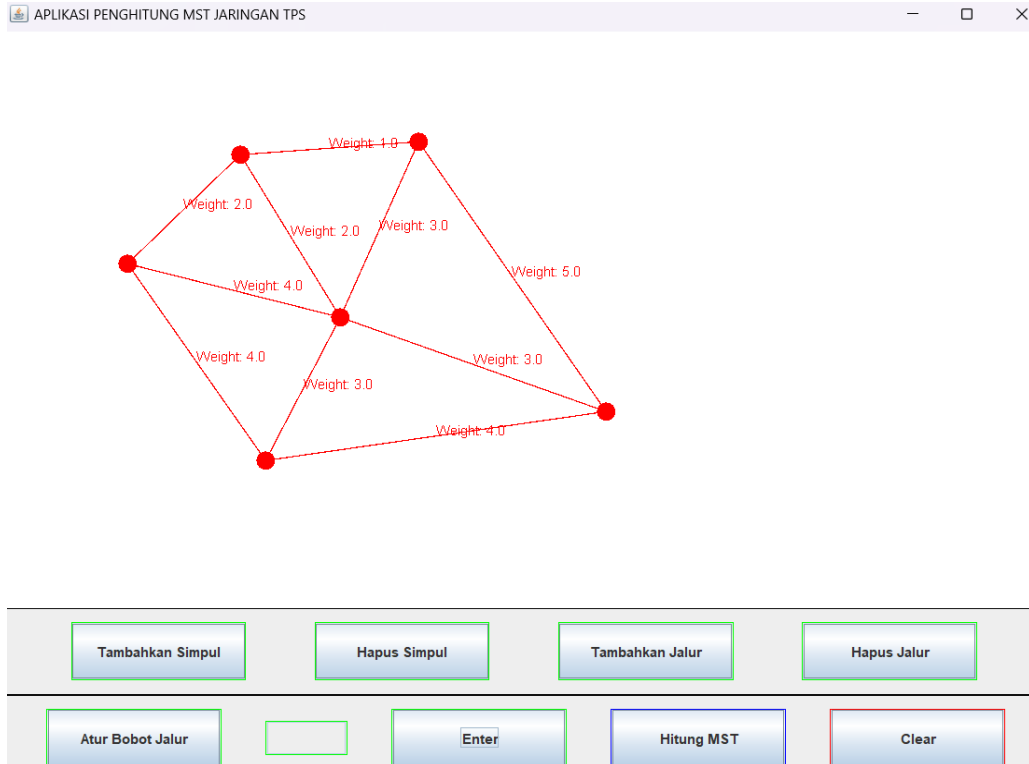
### 3.4 Implementasi dan Pengujian pada Program Java

Gambar 15 memperlihatkan tampilan GUI saat pengguna menambahkan simpul baru ke dalam graf. Antarmuka memungkinkan operator lapangan memasukkan lokasi TPS secara dinamis sehingga model dapat diperbarui tanpa perlu mengubah kode sumber.

Gambar 16 menunjukkan tampilan pengisian bobot jarak antar simpul melalui GUI. Fitur ini memudahkan verifikasi dan koreksi bobot jarak sebelum perhitungan MST dijalankan, sehingga data input lebih akurat dan keputusan operasional lebih dapat diandalkan.



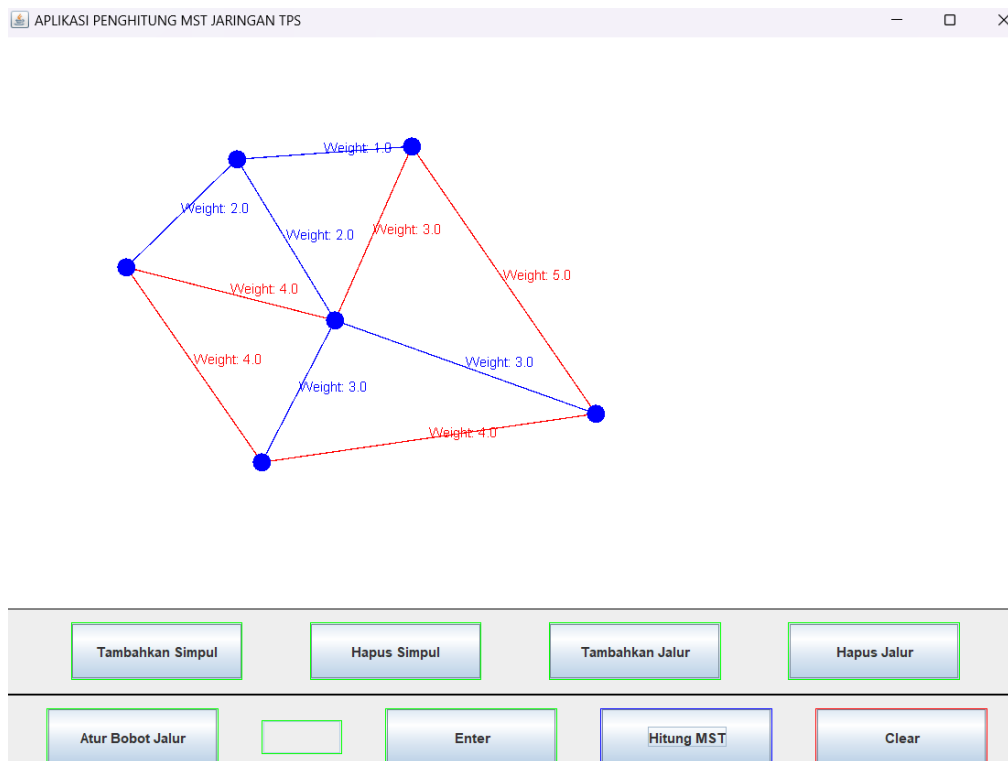
**Gambar 15.** Penambahan Simpul Melalui Antarmuka Aplikasi



**Gambar 16.** Penentuan Jalur dan Bobot Melalui Antarmuka Aplikasi

Gambar 16 menampilkan hasil perhitungan MST yang dihasilkan oleh aplikasi setelah tombol “Hitung MST” ditekan. Hasil pada Gambar 16 konsisten dengan perhitungan manual (total jarak 11 km) dan menunjukkan bahwa aplikasi dapat mengeksekusi algoritma Kruskal dengan waktu komputasi yang singkat, sehingga layak dipertimbangkan sebagai alat bantu pengambilan keputusan operasional.

Berdasarkan hasil di atas, penerapan MST melalui algoritma Kruskal dan antarmuka GUI dapat menyederhanakan perencanaan rute pengangkutan sampah, mengurangi jarak tempuh, dan menurunkan frekuensi perjalanan berulang. Hal ini relevan mengingat bahwa rute yang tidak terstruktur berkontribusi pada peningkatan konsumsi bahan bakar dan emisi karbon, sehingga optimasi rute menjadi strategi penting untuk menekan biaya operasional dan dampak lingkungan.



**Gambar 17.** Hasil Perhitungan MST di Aplikasi



#### 4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan algoritma Kruskal dalam sebuah aplikasi berbasis Java GUI untuk menentukan rute MST pada jaringan enam TPS di wilayah Surabaya Timur. Dengan pemodelan graf berbobot berdasarkan jarak jalan aktual antar lokasi, algoritma Kruskal menghasilkan struktur MST dengan total jarak minimum sebesar 11 km. Hasil ini menunjukkan potensi pengurangan rute tidak efisien dan penyederhanaan perencanaan jalur pengangkutan sampah. Aplikasi yang dikembangkan menyediakan antarmuka grafis yang memungkinkan input data lokasi dan jarak secara fleksibel serta menampilkan visualisasi rute optimal secara intuitif, sehingga berfungsi sebagai alat bantu operasional bagi pengelola kebersihan dalam merencanakan rute pengangkutan. Meskipun demikian, penelitian ini memiliki keterbatasan penting, yaitu cakupan studi yang hanya melibatkan enam TPS sebagai studi kasus skala kecil dan penggunaan bobot jarak yang bersifat statis tanpa mempertimbangkan variabilitas lalu lintas atau kapasitas armada; oleh karena itu hasil yang diperoleh bersifat indikatif dan perlu diuji ulang pada jaringan TPS yang lebih besar dan dinamis. Untuk penelitian lanjutan disarankan memasukkan data lalu lintas waktu nyata, mempertimbangkan kendala kapasitas dan waktu layanan, serta membandingkan kinerja Kruskal dengan metode optimasi rute lain untuk mengevaluasi robustitas dan skalabilitas solusi.

#### REFERENCES

- [1] M. H. Arifin, "Produksi Sampah di Surabaya Mencapai 1.800 Ton per Hari, Bisa Berkurang karena Masyarakat Masif Memilah," *Suara Surabaya*, 2024.
- [2] M. F. Miftahadi, I. Rachman, dan T. Matsumoto, "Optimizing Indonesian municipal solid waste collection scenarios: Integration of multi-objective search simulation and social cost-benefit analysis," *J. Mater. Cycles Waste Manag.*, vol. 26, no. 3, hal. 1569–1587, 2024, doi: 10.1007/s10163-024-01910-0.
- [3] B. C. Koloway, "Produksi Sampah di Surabaya Capai 1.800 Ton per Hari, Pemkot Gelontorkan Rp462 Miliar per Tahun," *TribunJatim*, 2025.
- [4] M. F. Jatmiko, Y. P. Wijianto, dan F. Z. Abidin, "Optimalisasi Rute Suroboyo Bus untuk Menjangkau Area Permukiman Baru," *KERN J. Ilm. Tek. Sipil*, vol. 10, no. 2, hal. 101–110, 2025, doi: 10.33005/kern.v10i2.80.
- [5] M. N. B. A. Yasin dan N. A. Pratomoatmojo, "Analisis Fenomena Densifikasi Perkotaan pada Wilayah Surabaya Timur dengan Metode Point Pattern Analysis," *J. Tek. ITS*, vol. 10, no. 1, hal. C26–C32, 2021.
- [6] BPS Kota Surabaya, "Kota Surabaya dalam Angka," 2025.
- [7] R. A. Mahmudah dan W. Herumurti, "Analisis Sistem Pengangkutan Sampah di Wilayah Surabaya Utara," *J. Tek. ITS*, vol. 5, no. 2, hal. D103–D108, 2016.
- [8] M. Chaerul, M. Puturuho, dan I. Artika, "Optimasi Rute Pengangkutan Sampah dengan Menggunakan Metode Nearest Neighbour (Studi Kasus: Kabupaten Manokwari, Papua Barat)," *J. Wil. dan Lingkung.*, vol. 10, no. April, hal. 55–68, 2022, doi: 10.14710/jwl.10.1.55-68.
- [9] I. D. Herrera-Granda, J. Cadena-Echeverría, J. C. León-Jácome, E. P. Herrera-Granda, D. Chavez Garcia, dan A. Rosales, "A Heuristic Procedure for Improving the Routing of Urban Waste Collection Vehicles Using ArcGIS," *Sustain.*, vol. 16, no. 13, 2024, doi: 10.3390/su16135660.
- [10] N. P. Akpan dan I. A. Iwok, "A Minimum Spanning Tree Approach of Solving a Transportation Problem," *Int. J. Math. Stat. Invent.*, vol. 5, no. 3, hal. 8–17, 2017.
- [11] A. Hosseini, "Uncertainty-Driven Stability Analysis of Minimum Spanning Tree Under Multiple Risk Variations," *Mathematics*, vol. 13, hal. 1–20, 2025, doi: <https://doi.org/10.3390/math13193100>.
- [12] A. Levitin, *Introduction to the Design & Analysis of Algorithms*, Third. in Always learning. Pearson, 2012.
- [13] W. C. Ekasaputra dan M. A. I. Pakereng, "Visualisasi Algoritma Kruskal dan Prim dalam Mencari Rute Terpendek Berbasis Unity 3D," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 7, no. 3, hal. 458–464, 2023, doi: 10.35870/jtik.v7i3.866.
- [14] A. Maulana, "Penerapan Kruskal Minimum Spanning Tree Pada Optimasi Rute Wisata Bojong Kabupaten Tegal," *J. Rekayasa Inform.*, vol. 2, no. 1, hal. 36–45, 2025.
- [15] Y. P. W. Rizkha Mardhatillah, Yulanda Rahmadiyah, "Implementasi Algoritma Kruskal dalam Menentukan Rute Terdekat di Fakultas Universitas Jambi Kampus Pinang Masak," *Multi Prox. J. Stat. Univ. Jambi*, vol. 1, no. 2, hal. 71–81, 2022, doi: [doi.org/10.22437/multiproximity.v1i2.17937](https://doi.org/10.22437/multiproximity.v1i2.17937).
- [16] S. Khoiriyah, N. Febriyanti, A. Oktavia, G. F. Tineza, dan F. Huda, "Implementasi Algoritma Kruskal Berbasis Budaya Lokal dalam Menentukan Rute Terpendek Napak Tilas Racana UMPRI," *J. Pendidik. Tambusai*, vol. 9, no. 3, hal. 40385–40393, 2025.
- [17] N. Arif Sudibyo, T. Purwanto, dan D. Rahmadi, "Minimum Spanning Tree Pada Distribusi Bahan Naskah USBN SD/MI Di Kabupaten Sragen," *Riemann Res. Math. Math. Educ.*, vol. 2, no. 2, hal. 64–69, 2020.
- [18] E. Sartika, A. Lusiani, I. Suhartini, dan N. Nuryati, "Algoritma Kruskal Untuk Menyelesaikan Masalah Optimasi Distribusi Jaringan Listrik di PLN UP3 Cimahi," *Epsil. J. Mat. Murni Dan Terap. (Epsilon J. Pure Appl. Math.)*, vol. 18, no. 2, hal. 192, 2024, doi: 10.20527/epsilon.v18i2.13256.
- [19] M. C. Ryadiani, N. Ardianingsih, dan R. Matheus, "Implementasi Algoritma Kruskal Untuk Optimasi Pengangkutan Sampah," in *Seminar Nasional Komputer dan Sistem Intelijen 2012*, Indonesia, 2012.
- [20] Erniyati dan P. Citra, "The implementation of the Kruskal algorithm for the search for the shortest path to the location of a building store in the city of Bogor," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 621, no. 1, hal. 2–11, 2019, doi: 10.1088/1757-899X/621/1/012010.
- [21] J. S. Kumar dan S. E.N, "An application of Spanning Tree Algorithm to Municipal Solid Waste Management," *Int. J. Math. Trends Technol.*, vol. 51, no. 4, hal. 303–306, 2017, doi: 10.14445/22315373/ijmtt-v51p539.
- [22] P. O. Ayegba, A. E. Okeyinka, M. Adebisi, E. O. Asani, J. A. Ayoola, dan G. C. Ben, "Comparative Performance Analysis of Kruskal and Prim MST Algorithms," *Int. J. Eng. Res. Technol.*, vol. 13, no. 12, hal. 5386–5391, 2020, [Daring]. Tersedia pada: [http://www.irphouse.com/ijert20/ijertv13n12\\_181.pdf](http://www.irphouse.com/ijert20/ijertv13n12_181.pdf)



- [23] N. T. S. Saptadi dan V. P. Taga, “Penerapan Algoritma K-Means Dalam Mengatasi Permasalahan Sampah (Studi Kasus: Pola Perilaku Masyarakat Dalam Membuang Sampah),” *Temat. J. Penelit. Tek. Inform. dan Sist. Inf.*, vol. 10, no. 1, hal. 1–14, 2022, doi: 10.56963/tematika.v10i1.134.
- [24] R. R. Maciel, A. D. de Souza, R. M. A. Almeida, dan J. P. R. R. Leite, “The Impact of IoT-Enabled Routing Optimization on Waste Collection Distance: A Systematic Review and Meta-Analysis,” *Logistics*, vol. 9, no. 4, hal. 1–27, 2025, doi: 10.3390/logistics9040161.
- [25] Z. A. Pandana dan P. Firdaus, “Kerjasama Sister City Antara Pemerintah Kota Surabaya dan Kota Kitakyushu Sebagai Upaya Pengelolaan Sampah Tahun 2012-2023,” *Glob. Focus*, vol. 04, no. 01, hal. 22–43, 2024, doi: <https://doi.org/10.21776/ub.jgf.2024.004.01.2>.
- [26] D. S. Kusuma dan S. Wibawani, “Strategi Pengelolaan Sampah di Super Depo Sutorejo Kota Surabaya,” *NeoRespublica J. Ilmu Pemerintah.*, vol. 5, no. 2, hal. 929–941, 2024, doi: 10.52423/neores.v5i2.276.
- [27] I. Sommerville, *Software Engineering*, 10 ed. Pearson, 2016.
- [28] R. S. Pressman dan B. R. Maxim, *Software Engineering: A Practitioners Approach*, 9th ed. McGraw-Hill Education, 2019.
- [29] J. L. Whitten dan L. D. Bentley, *Systems Analysis and Design Methods*. McGraw-Hill/Irwin, 2004.