INDEXIA: Informatic and Computational Intelligent Journal
Vol.07, No.01, Mei 2025, hlm. 25-33
P-ISSN: 2657-0424
E-ISSN: 2657-0432

RANCANGAN JARINGAN ACCESS POINT MENGGUNAKAN ALGORITMA KRUSKAL

Yohanes E. Putra[1)], Yosefina F. Riti[2,*)]
Ilmu Informatika, Fakultas Teknik Universitas Katolik Darma Cendika
Jl. H. Dr. Ir. Soekarno 201 Surabaya, Jawa Timur, 60117
e-mail: yohanes233406008@student.ukdc.ac.id[1)], yosefina.riti@ukdc.ac.id[2)]

(Naskah masuk : 21 Januari 2025 Diterima untuk diterbitkan : 02 Mei 2025)

ABSTRAK

Penempatan access point yang optimal dan efisien menjadi aspek penting pada sebuah instansi. Merancang jaringan dengan baik dapat meningkatkan produktivitas. Permasalahan yang terjadi pada jurnal ini yaitu penempatan access point yang kurang optimal mengakibatkan jaringan yang sulit dijangkau oleh beberapa orang di Universitas Katolik Darma Cendika. Tujuan dari penelitian ini yaitu untuk membuat rancangan jaringan yang lebih optimal agar akses jaringan pada ruangan-ruangan di Universitas Katolik Darma Cendika dapat lebih mudah untuk dijangkau. Menggunakan metode minimum spanning tree dan algoritma Kruskal dapat mengatasi permasalahan penempatan access point yang kurang optimal. Metode Minimum Spanning Tree merupakan metode untuk menemukan Kumpulan sisi graf yang menghubungkan semua simpul dalam graf tanpa membentuk siklus. Sedangkan algoritma Kruskal merupakan algoritma yang digunakan untuk menemukan jalur terpendek antara dua simpul dalam graf. Solusi yang diberikan ini adalah salah satu implementasi graf pada aspek kehidupan. Hasil dari penelitian ini yaitu rancangan access point yang lebih optimal di Universitas Katolik Darma Cendika.

***Kata Kunci**: Teori Graf, Algoritma Kruskal, Access Point, Minimum Spanning Tree*.

ABSTRACT

An agency's optimal and efficient placement of access points is an important aspect. Designing a network properly can increase productivity. The problem in this journal is that the less than optimal placement of access points results in a network that is difficult to reach some people at Darma Cendika Catholic University. The purpose of this research is to create a more optimal network design so that network access to rooms at Darma Cendika Catholic University can be more easily reached. Using the minimum spanning tree method and the Kruskal algorithm can overcome the problem of less than optimal access point placement. The Minimum Spanning Tree method is a method for finding a set of graph edges that connect all vertices in a graph without forming a cycle. The Kruskal algorithm is an algorithm used to find the shortest path between two vertices in a graph. The solution provided is one of the implementations of graphs in aspects of life. This research results in a more optimal access point design at Darma Cendika Catholic University.

***Keywords**: Graph Theory, Kruskal Algorithm, Access Point, Minimum Spanning Tree*.

I. PENDAHULUAN

Graf merupakan sekumpulan sebuah objek yang memungkinkan untuk saling terhubung serta dapat merepresentasikan permasalahan. Pada konsep graf, objek dapat digambarkan sebagai titik/simpul dan hubungan antara objek dapat digambarkan sebagai sisi. Kedua hal tersebut memiliki himpunannya masing-masing yaitu himpunan simpul dan himpunan sisi. Pada berbagai aspek kehidupan, konsep dari graf dapat diimplementasikan [1]. Salah satu penerapan konsep graf adalah optimalisasi jaringan internet. Faktor finansial dapat dipengaruhi dalam perancangan jaringan yang kurang efisien. Maka penting bagi sebuah instansi untuk merancang jaringan yang optimal untuk menunjang aktivitas di dalamnya [2].

Terdapat beberapa penelitian terdahulu yang telah melakukan riset dengan topik optimasi jaringan. Pada jurnal yang ditulis oleh Qurrota A'yuni Ar Ruhimat, berhasil meminimalkan panjang kabel pada jaringan intranet DISKOMINFO Bondowoso dengan menerapkan Algoritma Kruskal [2]. Selanjutnya jurnal yang ditulis oleh Ilma Puteri, berhasil membuat rancangan jaringan yang lebih efisien dan lebih

optimal pada lintasan kabel internet di Universitas Andalas [3]. Selanjutnya terdapat penelitian terkait tentang optimasi kabel internet yang dilakukan oleh Deddy Rahmadi, peneliti menghadapi permasalahan yaitu penumpukan kabel serta pemasangan kabel yang kurang efisien dan terstruktur di Kabupaten Sleman. Peneliti berhasil membuktikan bahwa penggunaan metode *Minimum Spanning Tree* ini sangat berpengaruh pada optimasi penggunaan kabel internet [4]. Penelitian terkait yang membahas rancangan jaringan *failover* dengan *router* mikrotik dalam meningkatkan ketersediaan jaringan pada Fakultas Teknik Universitas Pancasila menyatakan bahwa perancangan jaringan yang efisien dapat membantu pertukaran data yang sedang terjadi [5]. Berikutnya terdapat penelitian dengan algoritma yang berbeda berguna untuk membandingkan Algoritma Kruskal dan Prim. Penelitian tersebut membahas rencana pemasangan kabel *fiber optic* di ITERA dengan menggunakan Algoritma Prim. Hasil yang dari penelitian tersebut yaitu memperoleh *Minimum Spanning Tree* dengan jarak 16.503 meter. Berdasarkan penelitian tersebut peneliti menyimpulkan bahwa Algoritma Prim dapat digunakan sebagai metode untuk menemukan jaringan yang optimal pada jaringan [6]. Selanjutnya terdapat penelitian yang menggunakan Algoritma Kruskal dalam perencanaan pembangunan jaringan listrik ditulis oleh Fitri Annisa. Permasalahan yang terjadi pada jurnal tersebut yaitu meminimalkan biaya pembangunan. Hasil dari penelitian tersebut yaitu mengurangi penggunaan kabel listrik sepanjang 1.253 meter [7]. Selanjutnya terdapat penelitian yang ditulis oleh Jerry membahas optimasi jaringan komputer dengan menggunakan Algoritma Kruskal. Permasalahan yang dihadapi pada penelitian tersebut yaitu optimasi jaringan komputer agar sumber daya yang digunakan dapat lebih kecil. Hasil dari penelitian tersebut yaitu penggunaan kabel yang menghubungkan antar komputer menjadi lebih kecil dengan menggunakan Algoritma Kruskal [8]. Selanjutnya terdapat penelitian yang membahas topik yang hampir sama yaitu *Integrating Cost-231 Multi wall Propagation and Adaptive Data Rate Method for Access Point Placement* yang ditulis oleh Mukti. Jurnal tersebut memiliki masalah optimasi penempatan *Access Point* di lingkungan *indoor* dengan banyak hambatan, algoritma yang digunakan *Cost-231 Multiwall, ADR*. Hasil dari penelitian tersebut yaitu rekomendasi penempatan *Access Point* untuk sinyal yang kuat dan stabil di berbagai lantai gedung [9]. Selanjutnya penelitian yang ditulis oleh ByoungHo Kim memiliki masalah penempatan PMU yang kurang optimal. PMU (*Phasor Measurement Unit*) adalah teknologi utama yang secara langsung dapat meningkatkan kinerja pemantauan, kontrol, dan estimasi status dalam sistem tenaga. Algoritma yang digunakan adalah Kruskal. Hasil dari penelitian tersebut yaitu meminimalkan jumlah PMU dengan Observabilitas jaringan yang tinggi [10]. Selanjutnya penelitian yang ditulis oleh Tan memiliki masalah *Clustering Access Point* dan alokasi daya untuk efisiensi energi dalam Sistem *Massive* MIMO. *Massive* MIMO adalah antena yang sangat besar yang bisa digunakan untuk menangkap sinyal yang dengan cepat dan dalam jumlah yang banyak. Algoritma yang digunakan adalah *Algoritma Clustering*. Hasil dari jurnal tersebut yaitu peningkatan efisiensi daya dan cakupan jaringan di area padat pengguna [11]. Berikutnya terdapat penelitian yang menggunakan Algoritma Kruskal yang dikombinasikan dengan K-means untuk meningkatkan ketersedian *Wireless Sensor Network*. Pada penelitian tersebut penulis memiliki masalah dalam meningkatkan ketersediaan *Wireless Sensor Network* melalui pengelompokan *Access Point* dan *Minimum Spanning Tree*. WSN (*Wireless Sensor Network*) merupakan jaringan nirkabel tanpa infrastruktur. Algoritma yang digunakan yaitu Algoritma Kruskal dan K-means. Hasil dari penelitian tersebut yaitu peningkatan signifikan dalam ketersediaan jaringan dan penggunaan daya lebih efisien [12]. Selanjutnya terdapat penelitian yang membahas penempatan titik *Access Point* di jaringan *small-cell* dengan fleksibilitas infrastruktur parsial. Penelitian tersebut menggunakan UAV-AP untuk membantu jaringan tetap dalam menangani permintaan kapasitas puncak. Secara berurutan terdapat 2 algoritma, Algoritma Inter-AP Lloyd untuk menentukan penempatan *Access Point* secara optimal pada jaringan fleksibel dan Algoritma HAPPA untuk jaringan hibrida. Hasil simulasi pada penelitian tersebut menunjukkan peningkatan yang lebih optimal menggunakan Algoritma HAPPA [13].

Selanjutnya terdapat penelitian terdahulu yang secara khusus menggunakan Algoritma Kruskal dengan topik yang berbeda dapat dijadikan sebagai referensi pada penelitian ini. Contohnya penelitian menggunakan Algoritma Kruskal yang ditulis oleh Ulandari memiliki masalah dalam mencari rute terdekat. Hasil dari penelitian tersebut yaitu didapatkan rute terpendek yang menghubungkan semua titik dengan efisiensi biaya yang baik dan diimplementasikan pada dataset rute jaringannya [14]. Selanjutnya terdapat penelitian yang menggunakan Algoritma Kruskal dengan topik optimasi jaringan pipa PDAM (Perusahaan Daerah Air Minum). Permasalahan yang terjadi pada penelitian tersebut yaitu

INDEXIA: Informatic and Computational Intelligent Journal
Vol.07, No.01, Mei 2025, hlm. 25-33
P-ISSN: 2657-0424
E-ISSN: 2657-0432

debit air menuju konsumen yang kecil sedangkan debit yang dialirkan dari reservoir cukup. Hasilnya peneliti berhasil mengurangi penggunaan pipa sepanjang 10.610 meter dengan menggunakan Algoritma Kruskal [15].

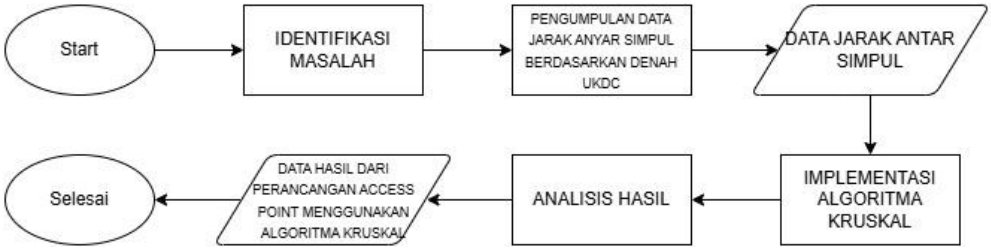
Permasalahan yang dihadapi dalam penelitian ini adalah jaringan pada *Access Point* yang sulit dijangkau di beberapa lokasi. Salah satu penyebab utama dari permasalahan tersebut adalah desain jaringan yang kurang optimal, sehingga menghasilkan jaringan yang sulit dijangkau. Selain itu, desain jaringan yang kurang optimal juga dapat menyebabkan biaya pembelian perangkat menjadi lebih tinggi dibandingkan jika menggunakan desain yang lebih efisien. Untuk mengatasi hal tersebut, penelitian ini menggunakan metode *Minimum Spanning Tree* dengan algoritma Kruskal yang merupakan bagian dari teori graf dalam perancangan jaringan *Access Point* di Universitas Katolik Darma Cendika. Teori graf adalah bidang matematika yang digunakan untuk memodelkan dan mengoptimalkan berbagai sistem jaringan, seperti jaringan komputer, transportasi, dan komunikasi nirkabel [16]. Objek penelitian adalah jalur yang optimal dan efisien dari ruangan PUSTIK ke berbagai ruangan di Universitas Katolik Darma Cendika.

Pemilihan Algoritma Kruskal didasarkan pada karakteristik algoritma ini yang sangat efektif dalam menangani graf dengan banyak simpul dan sisi, terutama graf tak berarah dan berbobot. Algoritma Kruskal bekerja dengan cara memilih sisi-sisi yang memiliki bobot terkecil secara berurutan tanpa memperhatikan simpul awal, sehingga cocok digunakan untuk mendesain jaringan yang mengutamakan penghematan biaya. Selain itu, algoritma ini memiliki fleksibilitas yang tinggi dalam menangani berbagai macam struktur graf, sehingga lebih mudah diimplementasikan untuk kasus-kasus perancangan jaringan yang membutuhkan bobot minimum.

Penelitian ini bertujuan untuk mengimplementasikan teori graf dalam aspek kehidupan, yaitu dalam merancang jaringan access point yang optimal dengan menggunakan metode *Minimum Spanning Tree* dan Algoritma Kruskal. Dengan pendekatan ini diharapkan dapat menemukan bobot minimum antar simpul yang merepresentasikan ruangan-ruangan di Universitas Katolik Darma Cendika, sehingga menghasilkan jaringan yang lebih efisien dan terjangkau.

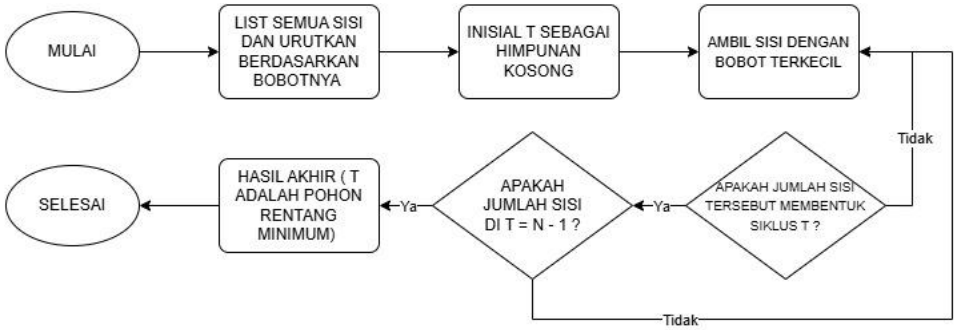
II. METODE PENELITIAN

Penelitian ini menggunakan metode *Minimum Spanning Tree* (MST) dengan Algoritma Kruskal untuk merancang jaringan *access point* yang optimal di Universitas Katolik Darma Cendika. Adapun tahapan dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Tahapan penelitian dimulai dengan identifikasi masalah, yaitu mengkaji area kampus yang memiliki jangkauan jaringan kurang optimal serta kebutuhan akses jaringan di setiap ruangan. Selanjutnya, dilakukan pengumpulan data berupa denah ruangan, jarak antar ruangan atau titik potensial pemasangan *access point*, serta jumlah dan spesifikasi perangkat yang akan digunakan. Setelah pengumpulan data selesai, maka akan terbentuk data jarak antar simpul yang merepresentasikan denah ruangan pada Universitas Katolik Darma Cendika. Berikutnya adalah implementasi algoritma Kruskal yaitu algoritma untuk menemukan *Minimum Spanning Tree* (MST) dalam graf, MST adalah subgraf yang menghubungkan semua *node* dengan jumlah bobot *edge* minimum tanpa adanya siklus. Algoritma ini bekerja dengan menambahkan *edge* dengan bobot terkecil secara bertahap ke MST, selama penambahan tersebut tidak membentuk siklus, sehingga sangat cocok untuk berbagai aplikasi optimasi jaringan [17].



Gambar 2. Diagram Alur Algoritma Kruskal

Graf berbobot adalah graf di mana setiap sisi (*edge*) memiliki nilai atau bobot yang mewakili ukuran seperti jarak, biaya, waktu, atau kapasitas. Nilai bobot ini menentukan nilai hubungan antar titik (*node*) dalam graf, sehingga graf berbobot digunakan untuk menyelesaikan masalah optimasi yang mempertimbangkan nilai biaya minimum atau maksimum antar titik [18]. Misalnya, pada penelitian ini graf berbobot diwakilkan pada jarak antar ruangan.

Dalam teori graf, metode *Minimum Spanning Tree* (MST) adalah cara untuk menghubungkan semua titik dalam graf berbobot dengan biaya total minimum tanpa membentuk siklus. MST membentuk subgraf dari himpunan titik dan sisi yang menghubungkan setiap titik tanpa membuat jalur berulang. Metode ini sering digunakan untuk optimasi jaringan, seperti dalam desain jaringan komputer atau telekomunikasi, jaringan listrik, dan sistem transportasi, karena memungkinkan desain jaringan yang mencakup semua titik dengan biaya terkecil [19]. *Access Point* (AP) adalah perangkat yang memungkinkan perangkat nirkabel terhubung ke jaringan kabel dan biasanya berfungsi dalam jaringan Wi-Fi untuk meningkatkan cakupan sinyal dan kapasitas jaringan. Penempatan AP yang optimal sangat penting untuk memaksimalkan cakupan sinyal, mengurangi interferensi, dan meningkatkan efisiensi jaringan di area tertentu [20].

Setelah data terkumpul, penelitian dilanjutkan dengan penyusunan graf berbobot. Setiap ruangan atau titik pemasangan *access point* direpresentasikan sebagai simpul (*node*), sedangkan hubungan antar simpul direpresentasikan sebagai sisi (*edge*) dengan bobot yang menggambarkan jarak atau parameter lain yang relevan. Graf berbobot ini menjadi dasar untuk penerapan Algoritma Kruskal. Tahapan berikutnya adalah penerapan Algoritma Kruskal. Proses ini dilakukan dengan mengurutkan sisi graf berdasarkan bobot terkecil dan secara bertahap menambahkan sisi ke MST, dengan memastikan tidak terbentuk siklus. Proses dilanjutkan hingga semua simpul terhubung dalam graf yang membentuk jalur optimal untuk penempatan *access point*.

Hasil dari penerapan Algoritma Kruskal akan dianalisis hasilnya, di mana hasil rancangan dievaluasi berdasarkan total bobot graf dan efisiensi konektivitas. Penelitian ini menghasilkan rancangan jaringan *access point* yang lebih optimal dengan cakupan dan efisiensi yang ditingkatkan. Semua tahapan penelitian didokumentasikan secara sistematis untuk memastikan validitas dan produktivitas hasil penelitian.

III. HASIL DAN PEMBAHASAN

Setelah mengetahui garis besar permasalahan yang telah dijabarkan pada bab pendahuluan, selanjutnya melakukan pengumpulan data terkait jarak antar simpul berdasarkan denah pada Universitas Katolik Darma Cendika.

Setelah mendapatkan data jarak antar simpul, berikutnya melakukan implementasi Algoritma Kruskal untuk mengetahui bagaimana rancangan jaringan *access point* yang optimal. Pada penelitian ini akan dilakukan implementasi Algoritma Kruskal dengan 2 cara yaitu perhitungan manual dan perhitungan dengan menggunakan *code python*.

A. Implementasi Algoritma Kruskal dengan perhitungan manual

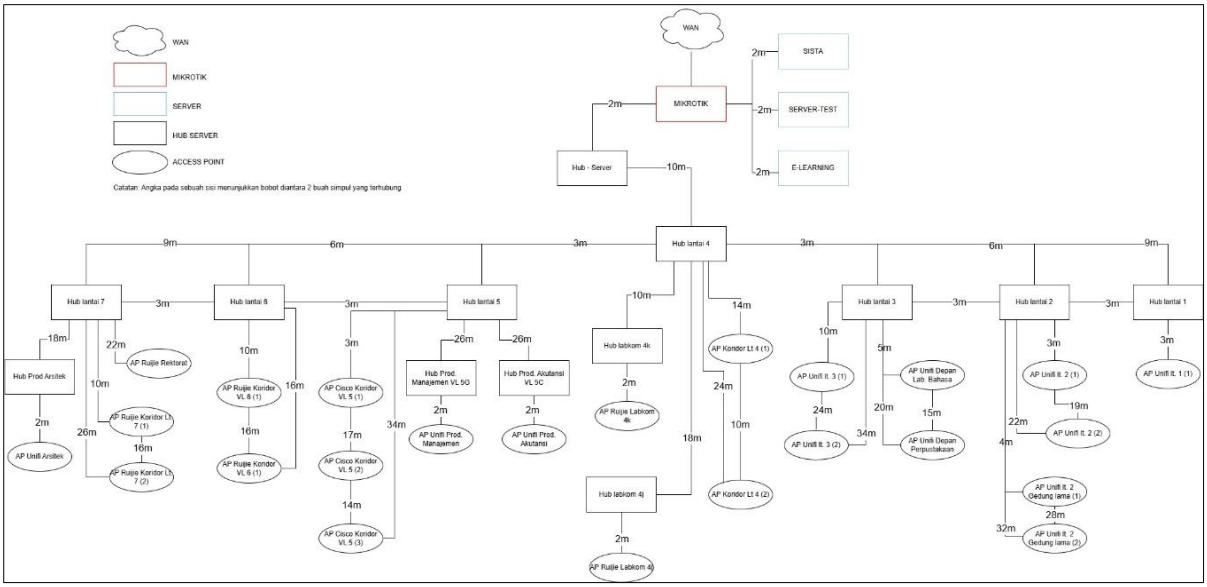
Sebelum melakukan perhitungan manual, langkah yang utama yaitu menjabarkan bagaimana cara kerja Algoritma Kruskal.

1. Melakukan pengumpulan data semua sisi pada sebuah graf dan melakukan pengurutan berdasarkan bobotnya dari yang terkecil hingga ke terbesar
2. Selanjutnya inisialisasi T sebagai himpunan kosong untuk menyimpan sisi-sisi dalam mendapatkan pohon rentang minimum
3. Ambil sisi dengan bobot terkecil yang belum diproses atau belum dipilih sebelumnya
4. Jika sisi tersebut membentuk siklus dalam T, abaikan dan kembali lagi ke langkah 3 yaitu ambil sisi dengan bobot terkecil yang belum diproses.

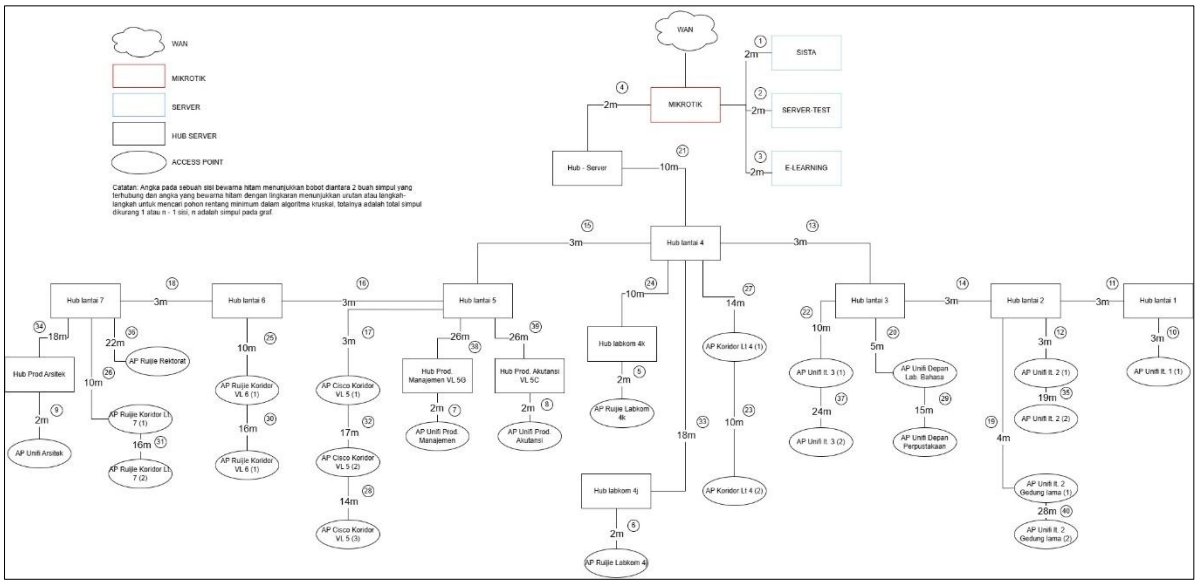
INDEXIA: Informatic and Computational Intelligent Journal
Vol.07, No.01, Mei 2025, hlm. 25-33
P-ISSN: 2657-0424
E-ISSN: 2657-0432

5. Jika sisi tersebut tidak membentuk siklus tambahkan sisi tersebut ke dalam T sebagai himpunan kosong
6. Jika jumlah sisi di T tidak sama dengan n – 1 dan n sebagai total simpul pada sebuah graf, maka kembali pada langkah 3 yaitu ambil sisi dengan bobot terkecil yang belum diproses.
7. Jika jumlah sisi di T sama dengan n -1 dan n adalah total simpul pada sebuah graf, maka proses selesai
8. Selanjutnya di dalam himpunan T akan terbentuk pohon rentang minimum

Sesuai dengan cara kerja pada Algoritma Kruskal, selanjutnya melakukan inisialisasi T sebagai himpunan kosong dan ambil sisi terkecil hingga total sisi tersebut n – 1 dimana n adalah total simpul dan tidak membentuk siklus di dalam himpunan T.



Gambar 3. Topologi Access Point Universitas Katolik Darma Cendika Dalam Bentuk Simpul Dan Sisi



Gambar 4. Hasil Implementasi Menggunakan Kruskal Dengan Perhitungan Manual

Tabel 1. Data Jarak Antar Simpul Yang Telah Diurutkan

Simpul asal	Simpul tujuan	Sisi	Bobot
Mikrotik	SISTA	Mikrotik – SISTA	2
Mikrotik	SERVER TEST	Mikrotik – SERVER TEST	2
Mikrotik	ELEARNING	Mikrotik – ELEARNING	2
Mikrotik	Hub Server	Mikrotik – Hub Server	2
Hub LABKOM 4K	AP Ruijie LABKOM 4K	Hub LABKOM 4K – AP Ruijie LABKOM 4K	2
Hub LABKOM 4J	AP Ruijie LABKOM 4J	Hub LABKOM 4J - AP Ruijie LABKOM 4J	2
Hub Prodi Manajemen VL 5G	AP Unifi Prodi Manajemen	Hub Prodi Manajemen VL 5G - AP Unifi Prodi Manajemen	2
Hub Prodi Akuntansi VL 5C	AP Unifi Prodi Akuntansi	Hub Prodi Akuntansi VL 5C - AP Unifi Prodi Akuntansi	2
Hub Prodi Arsitek	AP Unifi Prodi Arsitek	Hub Prodi Arsitek - AP Unifi Prodi Arsitek	2
Hub Lantai 1	AP Unifi Lantai 1(1)	Hub Lantai 1 - AP Unifi Lantai 1(1)	3
Hub Lantai 2	Hub Lantai 1	Hub Lantai 2 - Hub Lantai 1	3
Hub Lantai 2	AP Unifi Lantai 2 (1)	Hub Lantai 2 - AP Unifi Lantai 2 (1)	3
Hub Lantai 4	Hub Lantai 3	Hub Lantai 4 - Hub Lantai 3	3
Hub Lantai 3	Hub Lantai 2	Hub Lantai 3 - Hub Lantai 2	3
Hub Lantai 4	Hub Lantai 5	Hub Lantai 4 - Hub Lantai 5	3
Hub Lantai 5	Hub Lantai 6	Hub Lantai 5 - Hub Lantai 6	3
Hub Lantai 5	AP Cisco Koridor VL 5(1)	Hub Lantai 5 - AP Cisco Koridor VL 5(1)	3
Hub Lantai 6	Hub Lantai 7	Hub Lantai 6 - Hub Lantai 7	3
Hub Lantai 2	AP Unifi Lantai 2 Gedung Lama (1)	Hub Lantai 2 - AP Unifi Lantai 2 Gedung Lama (1)	4
Hub Lantai 3	AP Unifi Depan Lab. Bahasa	Hub Lantai 3 - AP Unifi Depan Lab. Bahasa	5
Hub Lantai 4	Hub Lantai 2	Hub Lantai 4 - Hub Lantai 2	6
Hub Lantai 4	Hub Lantai 6	Hub Lantai 4 - Hub Lantai 6	6
Hub Lantai 4	Hub Lantai 1	Hub Lantai 4 - Hub Lantai 1	9
Hub Lantai 4	Hub Lantai 7	Hub Lantai 4 - Hub Lantai 7	9
Hub Server	Hub Lantai 4	Hub Server - Hub Lantai 4	10
Hub Lantai 3	AP Unifi Lantai 3(1)	Hub Lantai 3 - AP Unifi Lantai 3(1)	10
AP Koridor Lantai 4(1)	AP Koridor Lantai 4(2)	AP Koridor Lantai 4(1) - AP Koridor Lantai 4(2)	10
Hub Lantai 4	Hub LABKOM 4K	Hub Lantai 4 - Hub LABKOM 4K	10
Hub Lantai 6	AP Ruijie Koridor VL 6(1)	Hub Lantai 6 - AP Ruijie Koridor VL 6(1)	10
Hub Lantai 7	AP Ruijie Koridor Lantai 7(1)	Hub Lantai 7 - AP Ruijie Koridor Lantai 7(1)	10
Hub Lantai 4	AP Ruijie Koridor Lantai 4(1)	Hub Lantai 4 - AP Ruijie Koridor Lantai 4(1)	14
AP Cisco Koridor VL 5(2)	AP Cisco Koridor VL5 (3)	AP Cisco Koridor VL 5(2) - AP Cisco Koridor VL5 (3)	14
AP Unifi Depan Lab. Bahasa	AP Unifi Depan Perpustakaan	AP Unifi Depan Lab. Bahasa - AP Unifi Depan Perpustakaan	15
AP Ruijie Koridor VL 6(1)	AP Ruijie Koridor VL 6(2)	AP Ruijie Koridor VL 6(1) - AP Ruijie Koridor VL 6(2)	16
AP Ruijie Koridor Lantai 7(1)	AP Ruijie Koridor Lantai 7(2)	AP Ruijie Koridor Lantai 7(1) - AP Ruijie Koridor Lantai 7(2)	16
AP Cisco Koridor VL 5(1)	AP Cisco Koridor VL5 (2)	AP Cisco Koridor VL 5(1) - AP Cisco Koridor VL5 (2)	17
Hub Lantai 4	Hub LABKOM 4J	Hub Lantai 4 - Hub LABKOM 4J	18
Hub Lantai 7	Hub Prodi Arsitek	Hub Lantai 7 - Hub Prodi Arsitek	18
AP Unifi Lantai 2(1)	AP Unifi Lantai 2(2)	AP Unifi Lantai 2(1) - AP Unifi Lantai 2(2)	19
Hub Lantai 3	AP Unifi Depan Perpustakaan	Hub Lantai 3 - AP Unifi Depan Perpustakaan	20
Hub Lantai 5	AP Cisco Koridor VL 5(2)	Hub Lantai 5 - AP1 Cisco Koridor VL 5(2)	20
Hub Lantai 2	AP Unifi Lantai 2(2)	Hub Lantai 2 - AP Unifi Lantai 2(2)	22
Hub Lantai 7	AP Ruijie Rektorat	Hub Lantai 7 - AP Ruijie Rektorat	22
AP Unifi Lantai 3(1)	AP Unifi Lantai 3(2)	AP Unifi Lantai 3(1) - AP Unifi Lantai 3(2)	24
Hub Lantai 4	AP Koridor Lantai 4 (2)	Hub Lantai 4 - AP Koridor Lantai 4 (2)	24
Hub Lantai 5	Hub Prodi Manajemen VL 5G	Hub Lantai 5 - Hub Prodi Manajemen VL 5G	26
Hub Lantai 5	Hub Prodi Akuntansi VL 5C	Hub Lantai 5 - Hub Prodi Akuntansi VL 5C	26
Hub Lantai 6	AP Ruijie Koridor VL 6(2)	Hub Lantai 6 - AP Ruijie Koridor VL 6(2)	26
Hub Lantai 7	AP Ruijie Koridor Lantai 7(2)	Hub Lantai 7 - AP Ruijie Koridor Lantai 7(2)	26
AP Unifi Lantai 2 Gedung Lama(1)	AP Unifi Lantai 2 Gedung Lama(2)	AP Unifi Lantai 2 Gedung Lama(1) - AP Unifi Lantai 2 Gedung Lama(2)	28
Hub Lantai 2	AP Unifi Lantai 2 Gedung Lama(2)	Hub Lantai 2 - AP Unifi Lantai 2 Gedung Lama(2)	32
Hub Lantai 3	AP Unifi Lantai 3(2)	Hub Lantai 3 - AP Unifi Lantai 3(2)	34
Hub Lantai 5	AP Cisco Koridor VL5(3)	Hub Lantai 5 - AP Cisco Koridor VL5(3)	34

B. Implementasi Algoritma Kruskal dengan menggunakan code dan bahasa pemrograman python

Pada tahap ini cara kerja Algoritma Kruskal sama seperti perhitungan manual, yang membedakan adalah mengaplikasikannya dalam sebuah baris kode. Dalam penelitian ini versi *python* yang digunakan yaitu versi 3.10.11 dan spesifikasi *device* yang digunakan untuk implementasi kode yaitu.

1. CPU Intel Gen 12 Core i5-12500H
2. Memory 16 GB
3. SSD 512 GB
4. GPU NVIDIA Geforce GTX 1650

Melakukan penjabar pada versi python dan spesifikasi device berguna sebagai pembanding jika ingin mencoba melakukan penelitian dengan langkah yang sama pada penelitian ini. Setelah menjabarkan spesifikasi, langkah selanjutnya adalah mengimplementasikan sesuai cara kerja Algoritma Kruskal, cara kerja Algoritma Kruskal yaitu menginisialisasi T sebagai himpunan kosong dan melakukan pengurutan sisi pada sebuah graf dari yang terkecil hingga terbesar. Berikutnya ambil sisi terkecil dan tambahkan ke dalam himpunan T hingga total di dalam himpunan T sama dengan n – 1 yang dimana n adalah total simpul pada sebuah graf dan sisi di dalam himpunan T tidak membentuk siklus. Berikut kode yang digunakan untuk mengimplementasikan Algoritma Kruskal dalam bahasa pemrograman *Python.*

INDEXIA: Informatic and Computational Intelligent Journal
Vol.07, No.01, Mei 2025, hlm. 25-33
P-ISSN: 2657-0424
E-ISSN: 2657-0432

```python
# Graph class to represent the graph
class Graph:
    def __init__(self, vertices):
        self.V = vertices  # Number of vertices
        self.edges = []    # List to store all edges
        self.node_map = {} # Dictionary to map node names to integers

    # Function to add an edge
    def add_edge(self, u, v, w):
        if u not in self.node_map:
            self.node_map[u] = len(self.node_map)
        if v not in self.node_map:
            self.node_map[v] = len(self.node_map)
        self.edges.append([self.node_map[u], self.node_map[v], w])

    # Function to find the parent of a node in the disjoint-set
    def find(self, parent, i):
        if parent[i] == i:
            return i
        return self.find(parent, parent[i])

    # Function to perform union of two subsets
    def union(self, parent, rank, x, y):
        root_x = self.find(parent, x)
        root_y = self.find(parent, y)
        if rank[root_x] < rank[root_y]:
            parent[root_x] = root_y
        elif rank[root_x] > rank[root_y]:
            parent[root_y] = root_x
        else:
            parent[root_y] = root_x
            rank[root_x] += 1

    # Kruskal's algorithm to find MST
    def kruskal(self):
        # Sort edges by weight
        self.edges.sort(key=lambda item: item[2])

        parent = []
        rank = []
        mst = []  # Resulting MST

        # Initialize parent and rank
        for node in range(self.V):
            parent.append(node)
            rank.append(0)

        # Process edges to find MST
        for u, v, w in self.edges:
            root_u = self.find(parent, u)
            root_v = self.find(parent, v)

            # If including this edge doesn't form a cycle, include it in the MST
            if root_u != root_v:
                mst.append([u, v, w])
                self.union(parent, rank, root_u, root_v)

        return mst
```

Gambar 5. Baris Kode Algoritma Kruskal Dalam Bahasa Pemrograman Python

Selanjutnya agar hasilnya dapat memberikan rancangan jaringan yang optimal, berikanlah data yang sesuai dengan studi kasus penelitian ini yaitu topologi yang terdapat bobot antar 2 buah simpul, simpul tersebut mewakilkan Lokasi jaringan pada denah jaringan Universitas Katolik Darma Cendika.

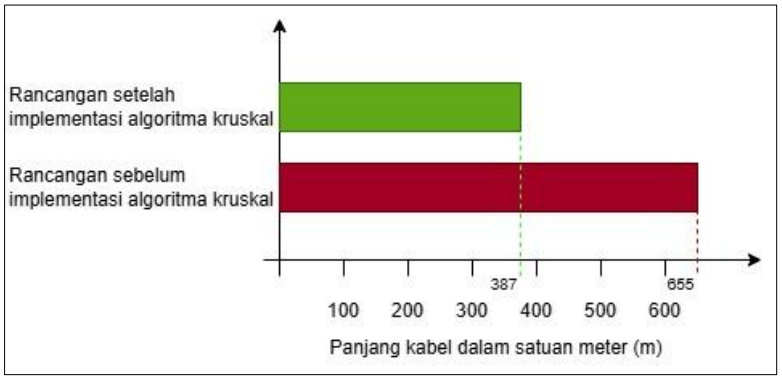
```python
# Collect unique nodes and create graph
for u, v, w in edges:
    nodes.add(u)
    nodes.add(v)

g = Graph(len(nodes))

# Add edges to the graph
for u, v, w in edges:
    g.add_edge(u, v, w)

# Find the MST using Kruskal's algorithm
mst = g.kruskal()

result = []

# Display the results
print("Edges in the Minimum Spanning Tree:")
for u, v, w in mst:
    # Reverse lookup to get node names from indices
    u_name = list(g.node_map.keys())[list(g.node_map.values()).index(u)]
    v_name = list(g.node_map.keys())[list(g.node_map.values()).index(v)]
    result.append(f"{u_name} -- {v_name} dengan bobot {w} meter")

for i in range(len(result)):
    print(f"Langkah {i + 1}: {result[i]}")
```

Gambar 6. Menambahkan Data Sesuai Dengan Studi Kasus Pada Kode Algoritma Kruskal

Implementasi ini memberikan gambaran bahwa Algoritma Kruskal dapat secara efisien menentukan jalur optimal yang menghubungkan seluruh simpul dengan bobot yang paling rendah, hal ini dibuktikan pohon rentang minimum pada sebuah himpunan yaitu T. Tidak hanya itu, jika melihat pada gambar yang dipaparkan, terdapat sisi yang digambarkan sebagai kabel perantara antara *switch* satu dengan yang lainnya membuat jarak semakin pendek, contohnya pada Hub lantai 7 menuju ke *Access Point* lantai 7 koridor 1 dan *Access Point* lantai 7 koridor 1 menuju *Access Point* lantai 7 koridor 2, sebelumnya memiliki total jarak 36 meter sekarang menjadi 26 meter. Jika menghitung bobot berdasarkan data keseluruhan diketahui dengan menggunakan Algoritma Kruskal dapat menghemat kabel yang digambarkan sebagai sisi pada gambar sejumlah 266 meter, dengan detail sebelum menggunakan Algoritma Kruskal yaitu 655 meter dan sesudah menggunakan Algoritma Kruskal menjadi 387 meter. Hal ini menguntungkan dalam segi finansial pada jaringan *Access Point* untuk kebutuhan infrastruktur Informasi Teknologi (IT) Universitas Katolik Darma Cendika khususnya infrastruktur *Access Point*.

```
Langkah 1: Mikrotik -- SISTA dengan bobot 2 meter
Langkah 2: Mikrotik -- SERVERTEST dengan bobot 2 meter
Langkah 3: Mikrotik -- ELEARNING dengan bobot 2 meter
Langkah 4: Mikrotik -- Hub Server dengan bobot 2 meter
Langkah 5: Hub LABKOM 4K -- AP Ruijie LABKOM 4K dengan bobot 2 meter
Langkah 6: Hub LABKOM 4J -- AP Ruijie LABKOM 4J dengan bobot 2 meter
Langkah 7: Hub Prodi Manajemen VL 5G -- AP Unifi Prodi Manajemen dengan bobot 2 meter
Langkah 8: Hub Prodi Akuntasi VL 5C -- AP Unifi Prodi Akuntasi dengan bobot 2 meter
Langkah 9: Hub Prodi Arsitek -- AP Unifi Prodi Arsitek dengan bobot 2 meter
Langkah 10: Hub Lantai 1 -- AP Unifi Lantai 1 (1) dengan bobot 3 meter
Langkah 11: Hub Lantai 2 -- Hub Lantai 1 dengan bobot 3 meter
Langkah 12: Hub Lantai 2 -- AP Unifi Lantai 2 (1) dengan bobot 3 meter
Langkah 13: Hub Lantai 4 -- Hub Lantai 3 dengan bobot 3 meter
Langkah 14: Hub Lantai 3 -- Hub Lantai 2 dengan bobot 3 meter
Langkah 15: Hub Lantai 4 -- Hub Lantai 5 dengan bobot 3 meter
Langkah 16: Hub Lantai 5 -- Hub Lantai 6 dengan bobot 3 meter
Langkah 17: Hub Lantai 5 -- AP Cisco Koridor VL 5 (1) dengan bobot 3 meter
Langkah 18: Hub Lantai 6 -- Hub Lantai 7 dengan bobot 3 meter
Langkah 19: Hub Lantai 2 -- AP Unifi Lantai 2 Gedung Lama (1) dengan bobot 4 meter
Langkah 20: Hub Lantai 3 -- AP Unifi Depan Lab. Bahasa dengan bobot 5 meter
Langkah 21: Hub Server -- Hub Lantai 4 dengan bobot 10 meter
Langkah 22: Hub Lantai 3 -- AP Unifi Lantai 3 (1) dengan bobot 10 meter
Langkah 23: AP Koridor Lantai 4 (1) -- AP Koridor Lantai 4 (2) dengan bobot 10 meter
Langkah 24: Hub Lantai 4 -- Hub LABKOM 4K dengan bobot 10 meter
Langkah 25: Hub Lantai 6 -- AP Ruijie Koridor VL 6 (1) dengan bobot 10 meter
Langkah 26: Hub Lantai 7 -- AP Ruijie Koridor Lantai 7 (1) dengan bobot 10 meter
Langkah 27: Hub Lantai 4 -- AP Koridor Lantai 4 (1) dengan bobot 14 meter
Langkah 28: AP Cisco Koridor VL 5 (2) -- AP Cisco Koridor VL 5 (3) dengan bobot 14 meter
Langkah 29: AP Unifi Depan Lab. Bahasa -- AP Unifi Depan Perpustakaan dengan bobot 15 meter
Langkah 30: AP Ruijie Koridor VL 6 (1) -- AP Ruijie Koridor VL 6 (2) dengan bobot 16 meter
Langkah 31: AP Ruijie Koridor Lantai 7 (1) -- AP Ruijie Koridor Lantai 7 (2) dengan bobot 16 meter
Langkah 32: AP Cisco Koridor VL 5 (1) -- AP Cisco Koridor VL 5 (2) dengan bobot 17 meter
Langkah 33: Hub Lantai 4 -- Hub LABKOM 4J dengan bobot 18 meter
Langkah 34: Hub Lantai 7 -- Hub Prodi Arsitek dengan bobot 18 meter
Langkah 35: AP Unifi Lantai 2 (1) -- AP Unifi Lantai 2 (2) dengan bobot 19 meter
Langkah 36: Hub Lantai 7 -- AP Ruijie Rektorat dengan bobot 22 meter
Langkah 37: AP Unifi Lantai 3 (1) -- AP Unifi Lantai 3 (2) dengan bobot 24 meter
Langkah 38: Hub Lantai 5 -- Hub Prodi Manajemen VL 5G dengan bobot 26 meter
Langkah 39: Hub Lantai 5 -- Hub Prodi Akuntasi VL 5C dengan bobot 26 meter
Langkah 40: AP Unifi Lantai 2 Gedung Lama (1) -- AP Unifi Lantai 2 Gedung Lama (2) dengan bobot 28 meter
```

Gambar 7. Hasil Implementasi Algoritma Kruskal Menggunakan Kode Python



Gambar 8. Diagram Untuk Panjang Kabel Sebelum Dan Sesudah Implementasi Algoritma Kruskal

IV. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan, jaringan *access point* Universitas Katolik Darma Cendika didesain secara optimal dengan menggunakan Algoritma Kruskal secara manual dan kode Python. Berdasarkan penelitian ini, algoritma Kruskal bekerja dengan baik untuk menemukan pohon rentang minimum atau minimum spanning tree yang menghubungkan semua node dengan bobot terkecil tanpa membentuk siklus. Penelitian ini menghasilkan desain jaringan yang efektif untuk kebutuhan infrastruktur (Informasi Teknologi) IT dalam segi finansial, khususnya access point dengan total bobot yang minimal yaitu 387 meter menggunakan data topologi jaringan dan jarak antar node. Implementasi ini menunjukkan bahwa algoritma Kruskal dapat diadaptasi untuk menyelesaikan masalah optimasi jaringan secara efisien melalui berbagai metode, baik manual maupun otomatis.

DAFTAR PUSTAKA

[1] H. Hendra and Y. F. Riti, “Perbandingan Algoritma Dijkstra Dan Floyd-Warshall Dalam Menentukan Rute Terpendek Stasiun Gubeng Menuju Wisata Surabaya,” *JIKA J. Inform.*, vol. 6, no. 3, p. 297, Oct. 2022, doi: 10.31000/jika.v6i3.6528.

[2] Q. A. A. Ruhimat, S. Slamin, and A. Malinda, “Efektivitas Algoritma Kruskal dalam Mengoptimalkan Jalur Terpendek pada Jaringan Intranet,” *JSN J. Sains Nat.*, vol. 2, no. 3, Art. no. 3, Jun. 2024, doi: 10.35746/jsn.v2i3.546.

[3] I. Puteri, M. Syafwan, and A. I. Baqi, “Penerapan Algoritma Prim Untuk Menentukan Lintasan Terpendek Jaringan Kabel Internet Di Universitas Andalas,” *J. Mat. UNAND*, vol. 10, no. 4, p. 476, Oct. 2021, doi: 10.25077/jmu.10.4.476-488.2021.

[4] D. Rahmadi, N. P. Maharani, M. R. Syifa, S. A. Sama, and G. F. Ardiansyah, “Optimasi Pemasangan Kabel Internet Antar Daerah Kabupaten Sleman Menggunakan Minimum Spanning Tree,” *J. Math. Theory Appl.*, vol. 2, no. 2, Art. no. 2, Apr. 2024, doi: 10.32938/j-math22202424.

[5] H. Zidan, Z. J. Arnecia, L. Oktaviani, G. Anisa, and B. R. Arsad, "Implementasi Failover Router MikroTik Untuk Meningkatkan Ketersediaan Jaringan Pada Fakultas Teknik Universitas Pancasila," *J. Inform. Adv. Comput. JIAC*, vol. 5, no. 1, Art. no. 1, May 2024.
[6] D. Suhika, T. Muliawati, and H. Ruwandar, "Optimalisasi Rencana Pemasangan Kabel Fiber Optic Di Itera Dengan Algoritma Prim," *AKSIOMA J. Program Studi Pendidik. Mat.*, vol. 9, no. 1, p. 86, Mar. 2020, doi: 10.24127/ajpm.v9i1.2597.
[7] F. Annisa and F. Muliani, "Penerapan Algoritma Kruskal Dalam Sisrem Jaringan Listrik Di Kecamatan Langsa Baro," *J. GAMMA-PI*, vol. 2, no. 02, pp. 5–9, Oct. 2020.
[8] F. Mahardika, "Penerapan Teori Graf Pada Jaringan Komputer Dengan Algoritma Kruskal," *J. Inform. J. Pengemb. IT*, vol. 4, no. 1, pp. 48–53, Jan. 2019, doi: 10.30591/jpit.v4i1.1032.
[9] F. S. Mukti *et al.*, "Integrating Cost-231 Multiwall Propagation and Adaptive Data Rate Method for Access Point Placement Recommendation," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 4, 2021, doi: 10.14569/IJACSA.2021.0120494.
[10] B.-H. Kim and H. Kim, "PMU Optimal Placement Algorithm Using Topological Observability Analysis," *J. Electr. Eng. Technol.*, vol. 16, no. 6, pp. 2909–2916, Nov. 2021, doi: 10.1007/s42835-021-00822-5.
[11] F. Tan, Q. Deng, and Q. Liu, "Energy-efficient access point clustering and power allocation in cell-free massive MIMO networks: a hierarchical deep reinforcement learning approach," *EURASIP J. Adv. Signal Process.*, vol. 2024, no. 1, p. 18, Jan. 2024, doi: 10.1186/s13634-024-01111-9.
[12] W. Jlassi, R. Haddad, R. Bouallegue, and R. Shubair, "A Combination of Kruskal and K-means Algorithms for Network Lifetime Extension in Wireless Sensor Networks," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, Harbin City, China: IEEE, Jun. 2021, pp. 658–663. doi: 10.1109/IWCMC51323.2021.9498594.
[13] G. R. Gopal, B. D. Rao, and G. P. Villardi, "Access Point Placement for Hybrid UAV-Terrestrial Small-Cell Networks," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1826–1841, 2021, doi: 10.1109/OJCOMS.2021.3100060.
[14] N. M. A. Ulandari, A. Amrullah, J. Junaidi, and S. Subarinah, "Implementasi Algoritma Kruskal Dalam Menentukan Rute Terdekat Pada Tempat Pariwisata di Daerah Lombok Tengah," *Griya J. Math. Educ. Appl.*, vol. 1, no. 4, pp. 578–589, Dec. 2021, doi: 10.29303/griya.v1i4.117.
[15] Y. M. Situmorang and A. Mansyur, "Pengoptimalan Jaringan Pipa Primer PDAM Tirtanadi Cabang Tuasan Dengan Menggunakan Algoritma Kruskal," *J. Ris. RUMPUN Mat. DAN ILMU Pengetah. ALAM*, vol. 2, no. 2, pp. 221–237, Oct. 2023, doi: 10.55606/jurrimipa.v2i2.1613.
[16] K. Rusek, J. Suarez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2260–2270, Oct. 2020, doi: 10.1109/JSAC.2020.3000405.
[17] Z.-H. Fu, S.-B. Chen, Y.-F. Ming, Y.-Q. Chen, and X.-J. Lai, "Dynamically Reconstructing Minimum Spanning Trees After Swapping Pairwise Vertices," *IEEE Access*, vol. 7, pp. 16351–16363, 2019, doi: 10.1109/ACCESS.2019.2894829.
[18] L. Cui, Z. Hao, Y. Jiao, H. Fei, and X. Yun, "VulDetector: Detecting Vulnerabilities Using Weighted Feature Graph Comparison," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2004–2017, 2021, doi: 10.1109/TIFS.2020.3047756.
[19] E. Cakir and Z. Ulukan, "An Intuitionistic Fuzzy MCDM Approach Adapted to Minimum Spanning Tree Algorithm for Spreading Content on Social Media," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, NV, USA: IEEE, Jan. 2021, pp. 0174–0179. doi: 10.1109/CCWC51732.2021.9375942.
[20] K. Farkas, Á. Huszák, and G. Gódor, "Optimization of Wi-Fi Access Point Placement for Indoor Localization," vol. 1, no. 1, 2013.