

APLIKASI BERBASIS DESKTOP UNTUK PENYELESAIAN GRAPH DENGAN ALGORITMA KRUSKAL DAN ALGORITMA PRIM

Siti Alvi Sholikhatin¹, Adi Budi Prasetyo², Ade Nurhopipah³

Informatika, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto^{1,2,3}
Purwanegara, Purwokerto, Indonesia

e-mail: sitialvi@amikompurwokerto.ac.id¹, adibudip81@gmail.com²,
ade_nurhopipah@amikompurwokerto.ac.id³

Received : Juli, 2020

Accepted : September, 2020

Published : Oktober, 2020

Abstract

A graph has several algorithms in its solution, including the Kruskal algorithm and Prim algorithm, both of which are greedy algorithms for determining the minimum spanning tree. Completion of graphs is useful in various fields of life, so an accurate graph calculation is important. Making an application to solve a graph, especially the Kruskal algorithm and Prim algorithm aims to facilitate the work of the graph so as to produce an accurate final result. The flow of research carried out are: a background review of research, study of literature and relevant literature, application design, building desktop-based applications, as well as implementation and application tests.

The level of technological readiness or TKT in this research is based on self-assessment which is at level 7, meaning the prototype demonstration system in the actual environment, with details of the TKT indicators as follows: TKT indicator 1 is met, TKT indicator 2 is met, TKT indicator 3 is not met, TKT indicator 4, TKT indicator 5 are met, TKT indicator 6 are met, TKT indicator 7 is met, TKT indicator 8 and 9 are not met. The application that has been built is useful for completing a graph with the Kruskal algorithm and Prim algorithm. This research was conducted to answer the crucial needs of a weighted graph settlement application.

Keywords: *Kruskal, Prim, Graph, Algorithm, Desktop Application*

Abstrak

Suatu graf memiliki beberapa algoritma dalam penyelesaiannya, diantaranya adalah algoritma Kruskal dan algoritma Prim, keduanya merupakan algoritma *greedy* untuk menentukan *minimum spanning tree*. Penyelesaian graf bermanfaat di berbagai bidang kehidupan, sehingga perhitungan graf yang akurat menjadi penting. Pembuatan aplikasi untuk menyelesaikan suatu graf terutama algoritma Kruskal dan algoritma Prim bertujuan untuk memudahkan pengerjaan graf sehingga menghasilkan hasil akhir yang akurat. Alur penelitian yang dilakukan yaitu: tinjauan latar belakang penelitian, studi pustaka dan literatur yang relevan, desain aplikasi, membangun aplikasi berbasis *desktop*, serta implementasi dan tes aplikasi.

Tingkat Kesiapterapan Teknologi atau TKT pada penelitian ini berdasarkan *self-assessment* yaitu berada pada tingkat 7, artinya demonstrasi prototipe sistem dalam lingkungan sebenarnya, dengan rincian TKT indikator sebagai berikut: TKT indikator 1 terpenuhi, TKT indikator 2 terpenuhi, TKT indikator 3 tidak terpenuhi, TKT indikator 4, TKT indikator 5 terpenuhi, TKT indikator 6 terpenuhi, TKT indikator 7 terpenuhi, TKT indikator 8 dan 9 tidak terpenuhi. Aplikasi yang telah dibangun ini bermanfaat untuk menyelesaikan suatu graf dengan algoritma Kruskal dan algoritma Prim. Penelitian ini dilakukan untuk menjawab kebutuhan yang cukup krusial akan sebuah aplikasi penyelesaian graf berbobot.

Kata Kunci: *Kruskal, Prim, Graf, Algoritma, Aplikasi Desktop*

1. PENDAHULUAN

Graf adalah struktur yang terdiri dari simpul atau *vertices* $V = \{v_1, v_2, \dots\}$ dan tepi atau *edges* $E = \{e_1, e_2, \dots\}$ dimana setiap *edge* atau tepi memiliki dua ujung, yang merupakan *verticle* atau simpul [1]. Graf digunakan untuk menggambarkan hubungan antara titik atau simpul dengan tepi atau garis yang saling berhubungan. Jurnal tentang graf pertama kali ditulis oleh matematikawan Swiss Leonhard Euler (1707-1783) dengan karya yang terkenal yaitu *Königsberg Bridge Problem*, dimana kota Königsberg (sekarang bernama Kaliningrad) yang berada di Prussia Timur merupakan wilayah yang terdiri dari dua pulau dan dikelilingi oleh sungai, dimana antarkota dihubungkan dengan tujuh jembatan. Permasalahan bermula ketika seseorang harus melewati ketujuh jembatan tersebut dan hanya bisa melewatinya sekali. Euler memecahkan masalah tersebut (dengan tidak ditemukannya solusi) dan menemukan fondasi teori graf [2]. Graf memiliki beberapa algoritma dalam penyelesaiannya. Salah satunya adalah algoritma Kruskal dan algoritma Prim. Algoritma Kruskal ditulis oleh Joseph Kruskal pada tahun 1956 dalam sebuah prosiding. Pada algoritma Kruskal, semua tepi dipersingkat dengan urutan *non-decreasing* dan selanjutnya dipilih tepi atau *edge* paling pendek sehingga menjadi *minimum spanning tree* [3]. Algoritma Prim atau *Jarnik-Prim algorithm* dimulai dengan satu titik kemudian tumbuh dengan menambahkan satu tepi hingga semua *vertices* atau simpul membentuk sebuah pohon [4]. Algoritma Kruskal dan algoritma Prim merupakan algoritma paling umum untuk menentukan *Minimum Spanning Tree* (MST) dalam suatu graf [5]. MST merepresentasikan sub-graf yang tidak terarah, dimana setiap node yang berhubungan memiliki total bobot yang minimal. Kedua algoritma tersebut, Kruskal dan Prim, memanfaatkan graf berbobot yang tidak terarah sehingga bisa menghasilkan solusi optimal untuk menyelesaikan permasalahan MST [6].

Graf dan algoritmanya digunakan di berbagai bidang kehidupan dan bermanfaat untuk menyelesaikan masalah terutama untuk optimasi pemanfaatan sumber daya. Misalnya, dalam membuat sebuah desain jalur perpipaan yang efektif dengan risiko yang paling kecil, graf berperan penting dalam perhitungannya. Seperti diungkapkan [7] bahwa banyak masalah

yang dapat disimpulkan sebagai persoalan yang berhubungan dengan himpunan, yang mana logika tersebut seringkali dapat digambarkan dengan graf. Bahkan teori graf juga bisa digunakan sebagai perhitungan untuk mengatur ventilasi sebuah jaringan [8]. Seperti pada penelitian yang dilakukan oleh [9] yang memanfaatkan perhitungan dengan algoritma Prim untuk menentukan pemasangan kabel *fiber optic* sehingga penggunaan kabel bisa efektif.

Penyelesaian graf memerlukan ketelitian yang tinggi untuk meminimalisir kesalahan, sehingga dibutuhkan aplikasi yang dapat membantu dalam pengerjaan agar didapat hasil yang akurat. Aplikasi yang kami bangun merupakan aplikasi *desktop* dengan menggunakan bahasa pemrograman C#, dan dapat menyelesaikan graf dengan algoritma Kruskal dan algoritma Prim dengan cepat dan tepat. Kontribusi yang dihasilkan dari riset ini adalah terciptanya aplikasi berbasis *desktop* yang memudahkan penyelesaian masalah yang menggunakan algoritma Kruskal dan algoritma Prim dengan akurat.

2. METODE PENELITIAN

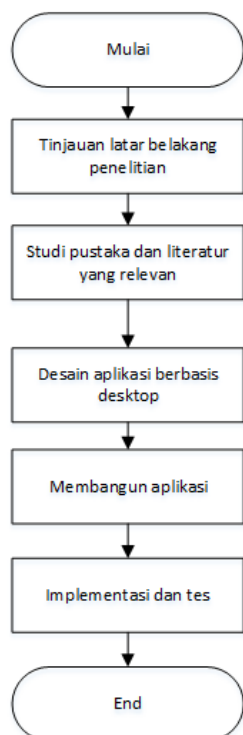
Menurut [4] Algoritma Kruskal adalah algoritma *greedy* untuk menentukan *minimum spanning tree*, dimulai dengan pohon V dimana setiap pohon adalah sebuah node tunggal. Sedangkan algoritma Prim adalah algoritma yang dimulai dengan node tunggal kemudian tumbuh menjadi sebuah pohon. Algoritma Prim dan algoritma Kruskal adalah dua algoritma klasik yang secara umum paling banyak digunakan untuk menentukan *minimum spanning tree* (MST). *Minimum Spanning Tree* dari sebuah graf berbobot adalah sebuah pohon rentang dimana jumlah dari bobot semua garis adalah yang paling kecil dari semua kemungkinan rentang pohon pada graf tersebut [3].

Penyelesaian dasar pada sebuah algoritma Kruskal adalah sebagai berikut: diketahui graf terhubung dengan sudut $G = \{V, E\}$, dan graf tidak terhubung tanpa garis $T = \{V, \emptyset\}$. Setiap sudut pada graf G adalah komponen yang saling terhubung, tambahkan nilai pada setiap garis pada graf G , kemudian hilangkan garis dengan nilai terendah. Jika dua sudut dari garis tersebut jatuh pada komponen terhubung yang berbeda, pilihlah salah satu ke graf T , atau abaikan dan pilihlah garis dengan nilai minimum dari garis yang tersisa. Ulangi langkah

tersebut sampai semua komponen terhubung menjadi satu dan semua sudut berada di komponen terhubung yang sama [10].

Perbedaan prinsip antara algoritma Prim dan Kruskal adalah jika pada algoritma Prim, sisi yang dimasukkan ke dalam pohon (*tree*) harus bersisian dengan sebuah simpul pada *tree* tersebut. Namun pada algoritma Kruskal, sisi yang dipilih tidak perlu bersisian dengan sebuah simpul, namun harus dicatat bahwa penambahan sisi tersebut harus tidak membentuk sebuah *cycle*. Dengan kalimat sederhana, algoritma Kruskal menggunakan *edge*, sedangkan algoritma Prim menggunakan *vertex* (sudut) yang terhubung.

Metode penelitian dijabarkan pada alur penelitian yang dapat dilihat di Gambar 1.



Gambar 1. Alur penelitian

Deskripsi Gambar 1.

Tinjauan latar belakang penelitian

Pada kegiatan ini, peneliti menentukan tujuan dan rumusan penelitian yang akan dilakukan. Kemudian, dari tujuan dan rumusan masalah penelitian tersebut disusun menjadi latar belakang penelitian. Tujuan dari dibangunnya aplikasi berbasis desktop yang mampu menyelesaikan graf dengan algoritma Prim dan algoritma Kruskal ini adalah untuk membantu peneliti khususnya, dan rekan-rekan sebidang

ilmu atau peneliti lain yang membutuhkan dalam penyelesaian permasalahan graf.

Studi pustaka dan literatur yang relevan

Studi pustaka dan literatur yang relevan dilakukan untuk mengetahui posisi peneliti dan penelitian yang sedang disusun, termasuk mengkaji penelitian sebelumnya dengan tema yang sama.

Desain aplikasi berbasis desktop

Desain aplikasi berbasis *desktop* terlebih dahulu harus menguasai cara penyelesaian graf algoritma Kruskal dan Prim, sehingga bisa mengimplementasikan ke dalam *coding* program pada bahasa pemrograman C#.

Membangun aplikasi

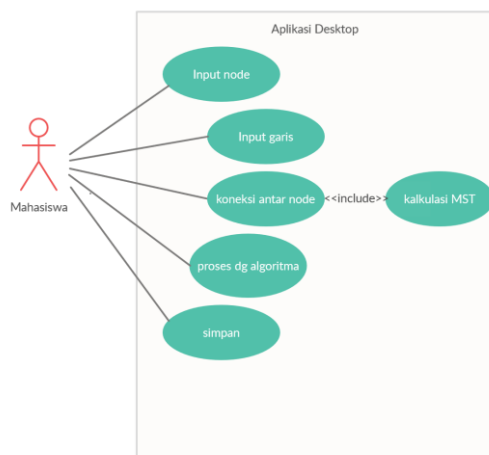
Membangun aplikasi sesuai dengan desain dan kebutuhan. Aplikasi terdiri dari dua kotak dialog yang berbeda untuk masing-masing penyelesaian algoritma.

Implementasi dan tes

Pada tahap ini, dilakukan tes dan uji kelayakan aplikasi, apakah sudah sesuai dengan aturan algoritma yang ada dan menghasilkan perhitungan yang akurat.

3. HASIL DAN PEMBAHASAN

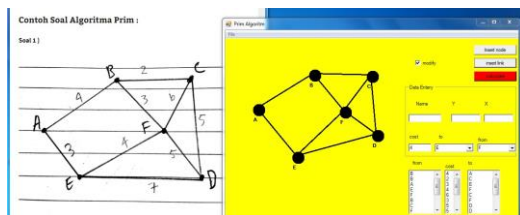
Mempelajari teori graf di berbagai bidang ilmu terutama di bidang teknologi dan informatika, penting untuk dipahami secara menyeluruh karena teori graf berkaitan dengan pemahaman ilmu dasar matematika. Teori graf mendorong kemampuan analisis serta pemahaman algoritma, dimana kemampuan dasar ini merupakan salah satu kebutuhan primer untuk membangun dan merancang sebuah sistem atau teknologi. Selain itu, peran teori graf juga cukup krusial antara lain di bidang teknologi jaringan, menentukan *travelling salesman problem*, bahkan analisis *handwriting recognition* serta *image segmentation* [11]. Berangkat dari kebutuhan tersebut, maka dirancang sebuah aplikasi berbasis *desktop* yang mudah digunakan untuk membantu mahasiswa dalam memahami teori graf secara optimal. Aplikasi ini juga memudahkan pengajar dalam memvisualisasikan dan mensimulasikan bentuk-bentuk dan penyelesaian graf sehingga lebih mudah dipahami. *Use case* aplikasi bisa dilihat pada Gambar 2.



Gambar 2. Use case aplikasi graf

Penjelasan dari *use case* aplikasi graf ini yaitu, aplikasi dapat melakukan operasi-operasi sebagai berikut: dapat menginputkan node, menginputkan garis, mengkoneksikan antar-node, melakukan perhitungan graf dengan algoritma Prim atau algoritma Kruskal, dapat mengetahui nilai Minimum Spanning Tree (MST), serta menyimpan hasil perhitungan yang telah dilakukan.

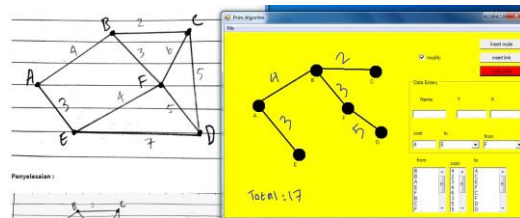
Desain aplikasi ini dibangun dengan menggunakan bahasa pemrograman C#, *user interface* diciptakan dengan tampilan yang sederhana untuk memudahkan penggunaan dan pemahaman pengguna. Tampilan antarmuka untuk penyelesaian algoritma Prim menggunakan aplikasi dan secara manual adalah sebagai berikut:



Gambar 3. Contoh soal pada graf dengan algoritma Prim

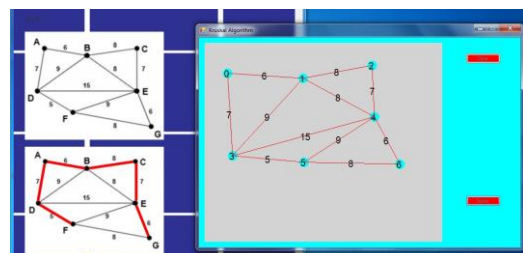
Dalam pembuatan bentuk graf pada aplikasi desktop diatas dapat dilakukan dengan langkah awal membuat node, dengan cara mengisikan *Data Entry* antara lain *Name* (node), tata letak (Y, X), tekan tombol *insert node* untuk mengimport *data entry* yang kita inputkan ke dalam *picture box* (penggambaran). Langkah kedua yaitu mengkoneksikan antara node satu dengan node yang lainnya dengan cara menginputkan *data entry Cost* (bobot), titik awal dan titik tujuan (to, from), menekan

tombol *insert link* untuk mengimportkan dan menghubungkan antar node yang sudah dibuat sebelumnya. Kemudian ketika tombol "Calculate" diklik, maka akan menghasilkan perhitungan sesuai dengan aturan algoritma Prim.



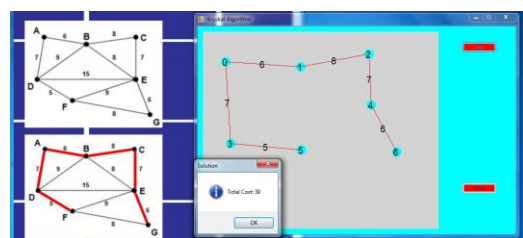
Gambar 4. Hasil perhitungan graf dengan algoritma Prim

Berikutnya, untuk perhitungan draf dengan algoritma Kruskal, tampilan antarmuka awal dan contoh soal bisa dilihat pada Gambar 5. Penambahan titik atau node, menggunakan *mouse on click* dengan cara memilih letak awal untuk pembuatan titik. Kemudian pilih titik awal dan tujuan yang akan di koneksi dengan cara menekan tombol CTRL pada keyboard dan klik titik yang akan dikoneksikan, kemudian diberikan *cost* (bobot).



Gambar 5. Contoh soal pada graf dengan algoritma Kruskal

Ketika tombol "Solve" ditekan, maka aplikasi akan menampilkan hasil perhitungan seperti pada Gambar 6.



Gambar 6. Hasil perhitungan graf dengan algoritma Kruskal

Dapat dilihat pada kedua aplikasi berbasis *desktop* tersebut, perhitungan menggunakan aplikasi sesuai dengan perhitungan manual,

dimana perhitungan manual cenderung membutuhkan waktu relatif lebih lama dalam penyelesaiannya. Aplikasi ini mampu memberikan visualisasi penyelesaian sebuah graf sehingga bisa lebih mudah dipahami dan pengajar mempunyai media untuk membantu dalam menjelaskan teori.

4. KESIMPULAN

Pembuatan aplikasi berbasis *desktop* untuk penyelesaian graf dengan algoritma Prim dan algoritma Kruskal bertujuan untuk membantu memahami graf dengan lebih mudah. Penelitian ini dilakukan berdasarkan kajian dan observasi di kelas yang mempelajari Teori Graf, dimana masih ditemukan berbagai kendala dalam memahami cara-cara penyelesaian teori graf yang cenderung bersifat abstrak dan sulit. Aplikasi ini merupakan bentuk visualisasi yang diharapkan mampu membantu mahasiswa dalam memahami penyelesaian teori graf. Selain itu, diharapkan juga dapat membantu pengajar dalam mendeskripsikan teori graf secara lebih interaktif dan menyenangkan.

Penelitian ini tentu masih terdapat banyak kekurangan, baik dari segi pengembangan aplikasi maupun penulisan laporan. Saran dan masukan dari *peer reviewer* dan rekan sejawat sangat diharapkan untuk meningkatkan kualitas tulisan kami agar menjadi lebih baik.

PERNYATAAN PENGHARGAAN

Ucapan terima kasih penulis sampaikan yang setulusnya kepada Universitas Amikom Purwokerto serta kepada Lembaga Penelitian dan Pengabdian Masyarakat Universitas Amikom Purwokerto yang telah membantu baik moril maupun materiil sehingga penelitian ini bisa kami selesaikan tepat waktu.

DAFTAR PUSTAKA

[1] A. Mamun and S. Rajasekaran, "An efficient Minimum Spanning Tree algorithm," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, 2016, pp. 1047–1052, doi: 10.1109/ISCC.2016.7543874.

[2] W. K. Chen, *Applied Theory Graph*. London: North-Holland Publishing Company, 2012.

[3] H. Li, Q. Xia, and Y. Wang, "Research and Improvement of Kruskal

Algorithm," *J. Comput. Commun.*, vol. 05, no. 12, pp. 63–69, 2017, doi: 10.4236/jcc.2017.512007.

[4] M. Sam and Yuliani, "PENERAPAN ALGORITMA PRIM UNTUK MEMBANGUN POHON MERENTANG MINIMUM (MINIMUM SPANNING TREE) DALAM ENGOPTIMALAN JARINGAN TRANSMISI NASIONAL PROVINSI SULAWESI SELATAN," *J. Din.*, vol. 4, no. 2, pp. 246–260, 2016, doi: 10.3929/ethz-b-000238666.

[5] R. Moharam and E. Morsy, "Genetic algorithms to balanced tree structures in graphs," *Swarm Evol. Comput.*, vol. 32, pp. 132–139, 2017, doi: <https://doi.org/10.1016/j.swevo.2016.06.005>.

[6] B. Munier, M. Aleem, M. A. Islam, M. A. Iqbal, and W. Mehmood, "A Fast Implementation of Minimum Spanning Tree Method and Applying it to Kruskal's and Prim's Algorithms," *Sukkur IBA J. Comput. Math. Sci.*, vol. 1, no. 1, p. 58, 2017, doi: 10.30537/sjcms.v1i1.8.

[7] N. Patel and K. M. Patel, "A survey on: enhancement of minimum spanning tree," *J. Res. Appl.*, vol. 5, no. 1, pp. 6–10, 2015.

[8] B. Wang, Q. Liu, and W. Wu, "Prim Algorithm Based on Heap for Finding Minimal Spanning Tree of Ventilation Network BT - 2015 International conference on Applied Science and Engineering Innovation," May 2015, pp. 1381–1386, doi: <https://doi.org/10.2991/asei-15.2015.273>.

[9] D. Suhika, T. Muliawati, and H. Ruwandar, "OPTIMALISASI RENCANA PEMASANGAN KABEL FIBER OPTIC DI ITERA DENGAN ALGORITMA PRIM," vol. 9, no. 1, pp. 86–92, 2020.

[10] S. Even, *Graph Algorithms 2nd Edition*, 2nd ed. Cambridge University Press, 2012.

[11] I. Rafid, "Performance evaluation for Kruskal's and Prim's Algorithm in Minimum Spanning Tree using Networkx Package and Matplotlib to visualizing the MST Result," no. May, 2019.