



Analisis Algoritma Greedy Untuk Mewarnai Graf

Hazel Pernanda Putra¹, Sisilia Sylviani^{1*}, Fahmi Candra Permana²

¹ Departemen Matematika FMIPA Universitas Padjadjaran

² Program Studi Pendidikan Multimedia, UPI Kampus di Cibiru

* Corresponding Author

Article Information

Article History:

Submitted: 01 08 2024

Accepted: 05 08 2024

Published: 06 30 2024

Key Words:

graph theory

graph coloring

greedy algorithm

DOI:10.33369/diophantine.v3i1.32261

Abstract

Graph coloring is one of the problems in graph theory. In graph theory, a greedy algorithm is a common algorithm in solving a problem. Greedy algorithms can be used for graph coloring; however, not all of the graph coloring problems that this algorithm uses can optimum outcomes globally. The selection of node sequencing used during the search also affects the success of the greedy algorithm in dealing with graph coloring cases. However, the use of greedy algorithms still helps reduce the number of colors used in non-directed graph coloring. The results show that the optimization of this algorithm is highly dependent on the order of selected nodes, which should be considered based on the number of degrees.

1. PENDAHULUAN

Algoritma Greedy telah menjadi fokus perhatian utama dalam dunia ilmu komputer, dikenal karena pendekatan yang sederhana namun efisien dalam menyelesaikan berbagai permasalahan optimasi (Ammar, 2019; Elektronik Sistem Informasi dan Komputer et al., 2021; Lakutu et al., 2023). Dalam konteks ini, pewarnaan graf merupakan suatu permasalahan klasik yang melibatkan penentuan warna pada setiap simpul graf sedemikian rupa sehingga simpul yang terhubung tidak memiliki warna yang sama. Pewarnaan graf memiliki relevansi yang luas dalam berbagai bidang, mulai dari jaringan komunikasi hingga penjadwalan tugas. Penggunaan algoritma Greedy dalam konteks ini memberikan solusi yang relatif cepat dan mudah dipahami. Penggunaan algoritma tersebut juga membuat keputusan yang diharapkan dapat menghasilkan solusi global optimal, dan penerapannya pada pewarnaan graf menciptakan strategi yang menarik dan efisien.

Dalam eksplorasi konsep ini, artikel ini akan membahas prinsip dasar algoritma Greedy, menerapkannya pada permasalahan pewarnaan graf. Diharapkan bahwa pemahaman yang mendalam tentang algoritma Greedy dalam konteks pewarnaan graf akan memberikan wawasan yang berharga bagi pembaca untuk memahami dan menerapkan strategi algoritmik yang efisien dalam menyelesaikan permasalahan serupa. Melalui analisis ini, kita dapat membangun pemahaman yang lebih baik tentang potensi serta batasan dari pendekatan ini dalam konteks pewarnaan graf.

Algoritma Greedy merupakan salah satu algoritma yang paling banyak digunakan di bidang ilmu matematika dan ilmu komputer. Penerapan algoritma ini dapat ditemukan pada teori graf, di mana algoritma ini dapat digunakan untuk kompresi data, menghitung lintasan terpendek, penjadwalan tugas, hingga pewarnaan graf. Pewarnaan graf sangat berguna dalam kehidupan sehari-hari. Salah satunya untuk mewarnai wilayah peta dengan warna paling sedikit, dengan batasan setiap wilayah di sekitarnya akan diberi warna yang berbeda. Setiap simpul menunjukkan wilayah tertentu pada peta. Sebuah garis menghubungkan suatu wilayah dengan wilayah lain.

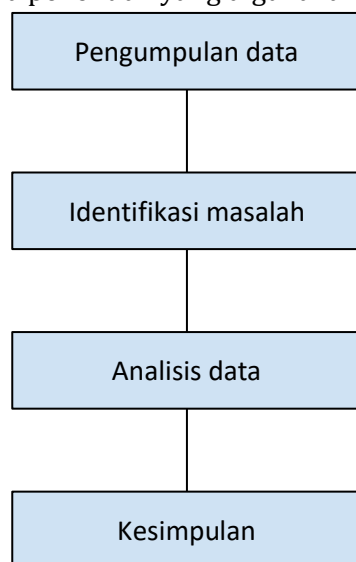
Pewarnaan graf adalah masalah pewarnaan warna M pada graf yang tidak berarah dengan menggunakan sebanyak mungkin warna M sehingga simpul atau garis tetangga tidak mempunyai warna yang sama (Fran et al., 2023; Oddang, 2015). Pewarnaan graf adalah jenis pelabelan graf dalam teori graf. Pelabelan tersebut

berkaitan dengan elemen warna yang memiliki pembatas tersendiri pada graf. Bilangan kromatik merupakan jumlah minimum warna yang digunakan untuk mewarnai suatu graf. Algoritma Greedy merupakan metode yang paling sering digunakan dalam masalah optimasi (maksimasi dan minimasi). Algoritma greedy mencari subset S dari himpunan C . Himpunan S harus memenuhi beberapa syarat, seperti menunjukkan solusi, dan dioptimasi oleh fungsi objektif. Algoritma ini mempertimbangkan simpul-simpul dalam graf sebagai urutan lalu mewarnai semua simpul dengan warna pertama yang dapat digunakan.

Penelitian ini mendapat urgensi yang tinggi karena pewarnaan graf memiliki aplikasi luas dalam berbagai bidang, termasuk optimasi jaringan, penjadwalan, dan alokasi sumber daya. Algoritma greedy menjadi fokus utama dalam penelitian ini karena dianggap sebagai pendekatan yang sederhana namun efisien untuk menyelesaikan masalah pewarnaan graf. Sebagai kontribusi signifikan, penelitian ini memperdalam pemahaman terhadap algoritma greedy, menggali lebih lanjut aspek-aspek kritis seperti pemilihan warna, urutan pemrosesan simpul, dan strategi pemilihan simpul berdasarkan tingkat kepentingan. Seiring dengan itu, riset sebelumnya telah memberikan fondasi untuk pengembangan algoritma ini, namun belum secara menyeluruh mempertimbangkan semua faktor yang dapat mempengaruhi kinerja algoritma greedy dalam konteks pewarnaan graf. Oleh karena itu, kontribusi riset ini terletak pada analisis mendalam terhadap faktor-faktor tersebut dan pengujian kinerja yang komprehensif, sehingga memberikan wawasan baru dan solusi yang lebih baik untuk meningkatkan efektivitas algoritma greedy dalam menangani pewarnaan graf pada berbagai skenario.

2. METODE

Pada Penelitian ini dilakukan beberapa peninjauan terhadap penelitian terdahulu yang memiliki keterkaitan dengan topik yang dibahas. Identifikasi masalah dalam penelitian ini adalah apakah algoritma greedy dapat digunakan untuk menyelesaikan permasalahan pewarnaan graf, dan jika demikian, bagaimana algoritma greedy dapat digunakan dalam pewarnaan graf. Penelitian ini dilakukan menggunakan metode studi pustaka, dengan mempelajari jurnal-jurnal yang berkaitan dengan, teori graf, pewarnaan graf, dan algoritma greedy. Berikut adalah kerangka penelitian yang digunakan.



Gambar 1. Kerangka Penelitian

Penelitian ini mengikuti alur penelitian yang terstruktur, dimulai dengan pengumpulan data sebagai langkah pertama. Data yang diperoleh melibatkan graf-graf yang memerlukan proses pewarnaan, dan data tersebut mencakup informasi tentang simpul-simpul, tepian, serta keterhubungan antar simpul. Setelah data terkumpul, langkah berikutnya adalah identifikasi masalah. Tahap selanjutnya analisis data yang kemudian dilanjutkan dengan pengambilan kesimpulan. Penelitian ini diharapkan memberikan kontribusi terhadap

pemahaman algoritma greedy dalam pewarnaan graf, memberikan wawasan lebih lanjut dan arah untuk pengembangan selanjutnya.

3. HASIL DAN PEMBAHASAN

Algoritma Greedy

Algoritma greedy men-set simpul dengan settingan tertentu v_i dengan $i = 1, 2, 3, \dots, n$ dan mengisi value v_i dengan warna yang berbeda dengan semua tetangga v_i di antara $d_1, \dots, v_{i-1}$. Warna baru dapat dimasukkan ke dalam simpul yang masih di tahap proses. Pewarnaan menggunakan algoritma greedy tidak selalu optimum dalam artian memiliki warna dengan jumlah minimum. Semua bergantung pada pengurutan; beberapa pengurutan dapat membuat pewarnaan dengan hasil yang optimal. Selain itu, algoritma pewarnaan ini juga dapat menghasilkan solusi yang tidak bagus, seperti pada graf crown. Algoritma greedy menghasilkan solusi warna maksimum yaitu sebesar derajat terbesar dari simpul + 1 dengan mempertimbangkan jumlah derajat.

$$\max_i \min\{d(x_i + 1, i)\}$$

1. Hitung semua derajat simpul pada graf.
2. Buat himpunan simpul belum berwarna dengan setiap simpul pada graf memiliki urutan derajat yang tidak naik. Simpul dengan derajat tertinggi adalah elemen pertama dalam himpunan.
3. Mulai himpunan solusi dengan himpunan kosong.
4. Pilih simpul yang akan diisi warnanya dalam himpunan simpul belum berwarna menggunakan selection function.
5. Hapus simpul yang dipilih dari simpul belum berwarna dan ubah warna simpul yang dipilih menjadi warna current.
6. Input simpul ke himpunan solusi.
7. Periksa semua simpul dalam himpunan simpul belum berwarna:
 - a. Simpul yang memenuhi syarat akan di input ke himpunan solusi (tidak bertetangga dengan elemen himpunan solusi),
 - b. Simpul yang memenuhi syarat akan dihapus dari himpunan simpul belum berwarna,
 - c. Warnai simpul yang memenuhi syarat dengan warna current.

Tambah indeks warna current, dan proses berakhir ketika seluruh simpul telah diwarnai. Kembali ke langkah tiga jika belum semua simpul terwarnai.

```

Given G = (V,E):
ComputeSemuaDerajat(v) for all v in V
Set BelumTerwarnai = V sorted in decreasing order of Derajat(V)
Set warnaSekarang = 0
set WarnaDenganSekarang = {}

while there are BelumTerwarnai nodes:
  set A = first element of BelumTerwarnai
  remove A from BelumTerwarnai
  set Color(A) = warnaSekarang
  set WarnaDenganSekarang = {A}

  for each v in BelumTerwarnai:
    if v is not adjacent to anything in WarnaDenganSekarang:
      set Color(v) = warnaSekarang
      add v to WarnaDenganSekarang
      remove v from BelumTerwarnai
    end if
  end for
  warnaSekarang = warnaSekarang + 1
end while

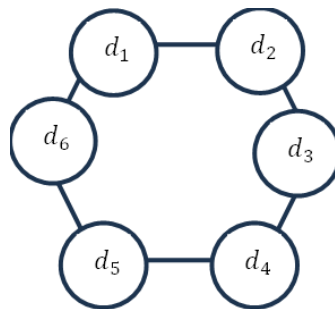
```

Gambar 2. Algoritma Greedy

Gambar algoritma greedy yang disajikan mengilustrasikan langkah-langkah eksekusi dalam menyelesaikan masalah pewarnaan graf. Pertama, algoritma memulai dengan inisialisasi, yaitu menetapkan warna pertama pada simpul awal. Selanjutnya, algoritma memilih simpul berikutnya untuk diproses berdasarkan kriteria greedy, yang mungkin melibatkan pemilihan simpul dengan derajat tertinggi atau strategi pemilihan lainnya. Setelah simpul terpilih, algoritma memilih warna untuk simpul tersebut, dengan memperhatikan warna-warna yang telah digunakan pada tetangga-tetangganya. Proses ini berulang hingga semua simpul pada graf telah diwarnai. Dalam setiap iterasi, algoritma memastikan bahwa pemilihan warna dilakukan dengan optimal, mencoba meminimalkan jumlah warna yang digunakan. Gambar algoritma ini memberikan pandangan visual yang jelas tentang langkah-langkahnya, memudahkan pemahaman dan implementasi algoritma greedy dalam konteks pewarnaan graf.

Pengujian

Kasus ke-1



Gambar 3. Graf dengan simpul belum terwarnai

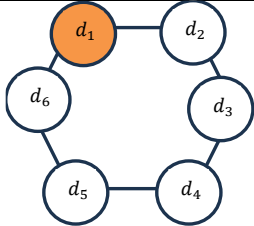
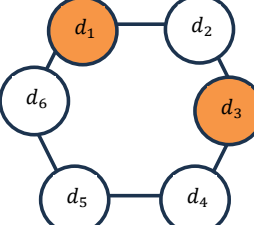
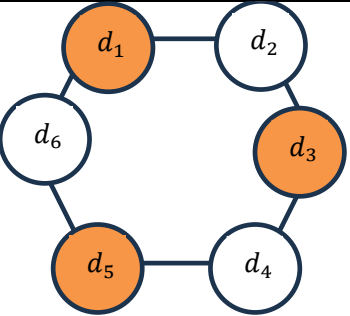
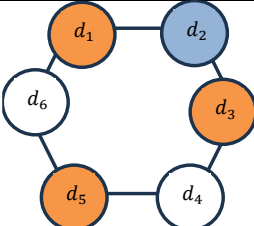
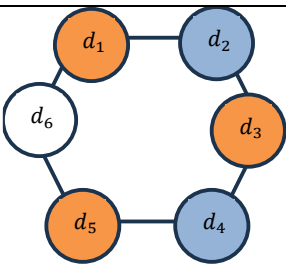
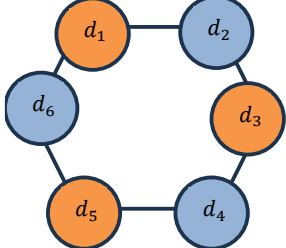
Simpul d_1 berderajat 2, simpul d_2 berderajat 2, simpul d_3 berderajat 2, simpul d_4 berderajat 2, simpul d_5 berderajat 2, simpul d_6 berderajat 2,

BelumTerwarnai : {d1, d2, d3, d4, d5, d6}

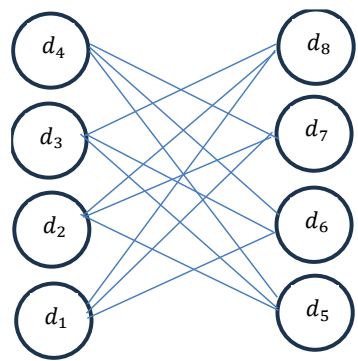
warnaSekarang = 0

himpSolusiTahap_1 : {}

Tabel 1. Proses Pewarnaan Graf pada Kasus 1

$A = d_1$ BelumTerwarnai : {d2, d3, d4, d5, d6} himpSolusiTahap_1 : {d1} Warna(d1) = 0	
$v = d_3$ BelumTerwarnai : {d2, d4, d5, d6} himpSolusiTahap_1: {d1, d3} Warna(d3) = 0	
$v = d_5$ BelumTerwarnai : {d2, d4, d6} himpSolusiTahap_1: {d1, d3, d5} Warna(d5) = 0 warnaSekarang = 1 himpSolusiTahap_2 : {}	
$A = d_2$ BelumTerwarnai : {d4, d6} himpSolusiTahap_2 : {d2} Warna(d2) = 1	
$v = d_4$ BelumTerwarnai = {d6} himpSolusiTahap_2 : {d2, d4} Warna(d4) = 1	
$v = d_6$ BelumTerwarnai : {} himpSolusiTahap_2 : {d2, d4, d6} Warna(d6) = 1	

Kasus ke-2



Gambar 4. Graf dengan simpul belum terwarnai

Simpul d1 berderajat 3, simpul d2 berderajat 3, simpul d3 berderajat 3, simpul d4 berderajat 3, simpul d5 berderajat 3, simpul d6 berderajat 3, simpul d7 berderajat 3, simpul d8 berderajat 3.

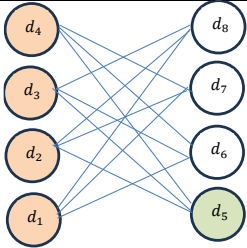
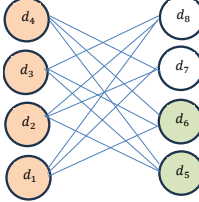
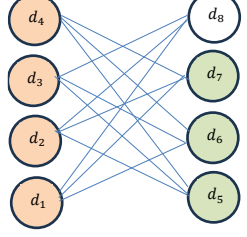
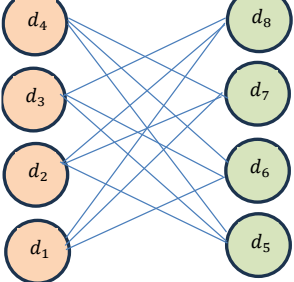
BelumTerwarnai : {d1, d2, d3, d4, d5, d6, d7, d8}

warnaSekarang = 0

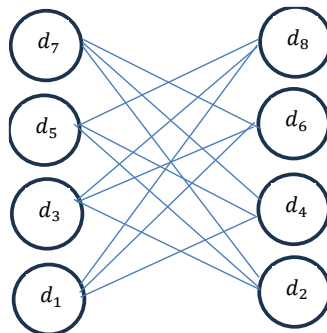
himpSolusiTahap_1 : {}

Tabel 2. Proses Pewarnaan Graf pada Kasus 2

A = d1 BelumTerwarnai : {d2, d3, d4, d5, d6, d7, d8} himpSolusiTahap_1: {d1} Warna(d1) = 0	
v = d2 BelumTerwarnai : {d3, d4, d5, d6, d7, d8} himpSolusiTahap_1 : {d1, d2} Warna(d2) = 0	
v = d3 BelumTerwarnai : { d4, d5, d6, d7, d8} himpSolusiTahap_1 : {d1, d2, d3} Warna(d3) = 0	
v = d4 BelumTerwarnai : {d5, d6, d7, d8} himpSolusiTahap_1 : {d1, d2, d3, d4} Warna(d4) = 0 Tahap 2: warnaSekarang = 0 himpSolusiTahap_2 : {}	

$A = d5$ BelumTerwarnai : {d6, d7, d8} himpSolusiTahap_2: {d5} Warna(d5) = 1	
$v = d6$ BelumTerwarnai : {d7, d8} himpSolusiTahap_2 : {d5, d6} Warna(d6) = 1	
$v = d7$ BelumTerwarnai : { d8} himpSolusiTahap_2 : { d5, d6, d7} Warna(d7) = 1	
$v = d8$ BelumTerwarnai : {} himpSolusiTahap_2 : { d5, d6, d7, d8} Warna(d8) = 1	

Kasus ke-3



Gambar 4. Graf dengan simpul belum terwarnai

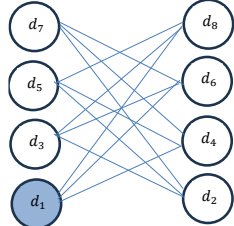
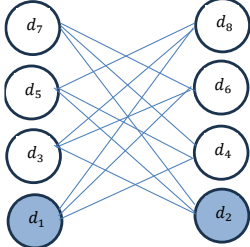
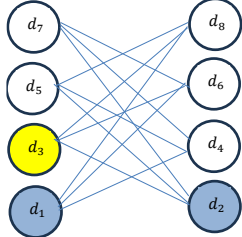
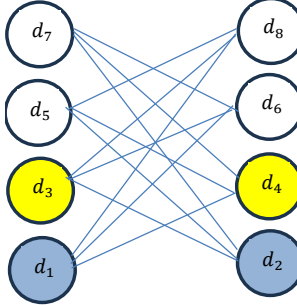
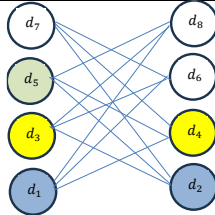
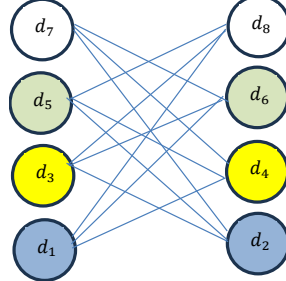
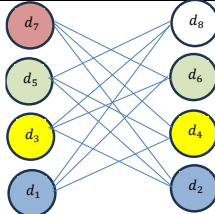
Simpul d1 berderajat 3, simpul d2 berderajat 3, simpul d3 berderajat 3, simpul d4 berderajat 3, simpul d5 berderajat 3, simpul d6 berderajat 3, simpul d7 berderajat 3, simpul d8 berderajat 3.

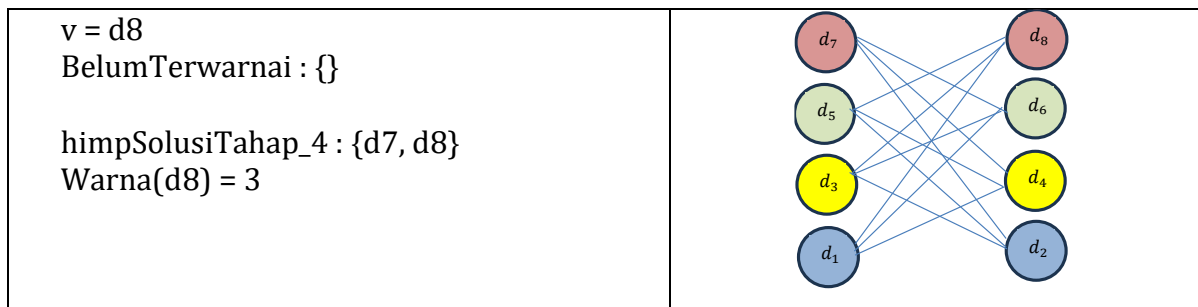
BelumTerwarnai : {d1, d2, d3, d4, d5, d6, d7, d8}

warnaSekarang = 0

himpSolusiTahap_1: {}

Tabel 3. Proses Pewarnaan Graf pada Kasus 3

$A = d_1$ BelumTerwarnai : {d2, d3, d4, d5, d6, d7, d8} himpSolusiTahap_1: {d1} Warna(d1) = 0	
$v = d_2$ BelumTerwarnai : {d3, d4, d5, d6, d7, d8} himpSolusiTahap_1 : {d1, d2} Warna(d2) = 0 warnaSekarang = 0 himpSolusiTahap_2 : {}	
$A = d_3$ BelumTerwarnai : { d4, d5, d6, d7, d8} himpSolusiTahap_2: {d3} Warna(d3) = 1	
$v = d_4$ BelumTerwarnai : { d5, d6, d7, d8} himpSolusiTahap_2 : {d3, d4} Warna(d4) = 1 warnaSekarang = 0 himpSolusiTahap_3 : {}	
$A = d_5$ BelumTerwarnai : {d6, d7, d8} himpSolusiTahap_3: {d5} Warna(d5) = 2	
$v = d_6$ BelumTerwarnai : {d7, d8} himpSolusiTahap_3 : {d5, d6} Warna(d6) = 2 warnaSekarang = 0 himpSolusiTahap_4 : {}	
$A = d_7$ BelumTerwarnai : {d8} himpSolusiTahap_4: {d7} Warna(d7) = 3	



Hasil Pengujian

Tiga kasus diatas menampilkan hasil yang berbeda, dapat dilihat bahwa algoritma greedy berhasil di kasus pertama dan kedua dan memberikan hasil yang optimal. Namun, dalam kasus ketiga, algoritma greedy memberikan hasil empat warna (jumlah maksimum warna = jumlah derajat tertinggi plus 1). Kasus 2 dan 3 memiliki perbedaan dalam cara penomoran simpul dengan derajat yang sama. Artinya, dalam mewarnai graf, algoritma greedy tidak menjamin solusi yang diharapkan. Namun, jumlah warna tidak boleh melebihi jumlah terbesar derajat simpul + 1 dalam himpunan simpul pada graf G . Optimasi algoritma greedy dalam hal mewarnai graf juga bergantung pada p .

4. SIMPULAN

Berdasarkan uraian sebelumnya, dapat disimpulkan bahwa algoritma greedy adalah salah satu pilihan yang dapat digunakan untuk menyelesaikan permasalahan pewarnaan graf. Namun perlu diperhatikan bahwa meskipun algoritma ini dapat memberikan solusi, tidak semua solusi yang dihasilkannya akan mencapai nilai optimum. Selain itu, pemilihan fungsi seleksi simpul yang digunakan memiliki peran krusial dalam kesuksesan algoritma greedy. Optimasi algoritma ini sangat tergantung pada urutan simpul yang dipilih, yang sebaiknya dipertimbangkan berdasarkan jumlah derajatnya. Untuk penelitian selanjutnya, rencananya akan difokuskan pada eksplorasi strategi pemilihan simpul yang lebih canggih dan penyesuaian parameter algoritma guna meningkatkan kemampuannya dalam mencapai solusi yang lebih mendekati nilai optimum. Selain itu, penelitian akan mempertimbangkan penggabungan algoritma greedy dengan pendekatan lainnya, seperti algoritma genetika atau pencarian lokal, untuk melihat potensi peningkatan kinerja dan efisiensi dalam menangani graf-graf yang lebih kompleks.

REFERENSI

- Ammar, M. (2019). Implementasi Algoritma Greedy Dalam Menyelesaikan Kasus Knapsack Problem Pada Jasa Pengiriman Pt Citra Van Titipan Kilat (Tiki) Kota Makassar. *Jurnal Axiomath : Jurnal Matematika Dan Aplikasinya*, 1(2), 26–32. <https://www.ejournals.umma.ac.id/index.php/axiomath/article/view/304>
- Elektronik Sistem Informasi dan Komputer, J., Grace, D., Tanciga, amar S., Nurdin, N., & Teknik Informatika STMIK Bina Mulia Palu, J. (2021). Sistem Informasi Letak Geografis Penentuan Jalur Tercepat Rumah Sakit Di Kota Palu Menggunakan Algoritma Greedy Berbasis Web. *Jurnal Elektronik Sistem Informasi Dan Komputer*, 4(2), 59–76. <https://www.jesik.web.id/index.php/jesik/article/view/87>
- Fran, F., Kusumastuti, N., Robiandi,), Program,), & Matematika, S. (2023). Bilangan Kromatik Harmonis Pada Graf Payung, Graf Parasut, Dan Graf Semi Parasut. *Jurnal Matematika Sains Dan Teknologi*, 24(1), 15–22. <https://doi.org/10.33830/JMST.V24I1.3945.2023>

- Lakutu, N. F., Katili, M. R., Mahmud, S. L., Yahya, I., & Matematika, J. (2023). Algoritma Dijkstra dan Algoritma Greedy Untuk Optimasi Rute Pengiriman Barang Pada Kantor Pos Gorontalo. *Euler : Jurnal Ilmiah Matematika, Sains Dan Teknologi*, 11(1), 55–65. <https://doi.org/10.34312/EULER.V11I1.18244>
- Oddang, R. (2015). Pembangunan Sistem Penjadwalan Kuliah Menggunakan Algoritma Pewarnaan Graf. *D'ComPutarE: Jurnal Ilmiah Information Technology*, 1(2), 50–55. <https://www.journal.uncp.ac.id/index.php/computare/article/view/163/155>