

Penerapan Algoritma Greedy pada Pewarnaan Graf beserta Aplikasinya

Muhammad Akmal Arifin - 13520037¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13520037@std.itb.ac.id

Abstract—Teori graf merupakan salah satu teori dari matematika diskrit, teori ini menggambarkan suatu relasi antar komponen yang terdapat di dalamnya, komponen ini dilambangkan dengan simpul dan relasi dilambangkan dengan sisi. Teori graf juga memiliki beberapa permasalahan yang dapat menggambarkan suatu permasalahan nyata di dunia. Salah satunya ada pewarnaan graf. Secara singkat, pewarnaan graf bertujuan untuk memberi warna pada setiap simpul atau sisi akan tetapi setiap simpul atau sisi yang bertetangga tidak boleh memiliki warna yang sama. Tujuannya adalah untuk mendapatkan jumlah warna terdikit. Salah satu penyelesaian permasalahan tersebut adalah dengan menggunakan algoritma greedy. Prinsip dari algoritma tersebut adalah pada setiap langkah yang diambil, algoritma memilih langkah optimal lokal dengan harapan langkah selanjutnya mengarah kepada solusi yang optimal.

Keywords— Algoritma Greedy, Pewarnaan Graf, Teori Graf.

I. PENDAHULUAN

Teori graf merupakan salah satu teori dari matematika diskrit, teori ini menggambarkan suatu relasi antar komponen yang terdapat di dalamnya, komponen ini dilambangkan dengan simpul (node) dan relasi dilambangkan dengan sisi (edge). Dalam kehidupan sehari-hari, teori graf dapat merepresentasikan dan menyelesaikan permasalahan yang ada di kehidupan nyata, salah satu permasalahan yang terdapat di graf adalah pewarnaan graf.

Maksud dari pewarnaan graf adalah kita diharuskan memberi warna pada masing-masing simpul, akan tetapi setiap simpul yang terhubung dengan sisi tidak boleh memiliki warna yang sama. Semakin sedikit warna yang digunakan maka semakin optimal solusi tersebut.

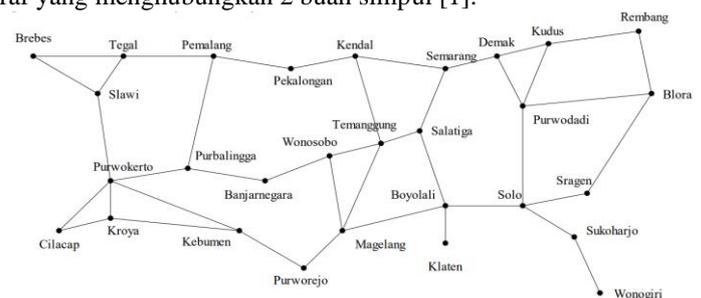
Permasalahan ini dapat menggambarkan beberapa permasalahan yang terdapat di dunia nyata, contohnya yaitu, pewarnaan negara di dalam peta, permanan sudoku dan *art gallery problem*. Untuk menyelesaikan permasalahan pewarnaan graf dapat menggunakan salah satu algoritma yang sudah cukup terkenal, yaitu algoritma greedy.

II. LANDASAN TEORI

A. Teori Graf

Secara matematis, graf didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, dengan V

merupakan himpunan tidak kosong yang berisikan simpul-simpul pada graf, dan E merupakan himpunan dari sisi-sisi pada graf yang menghubungkan 2 buah simpul [1].



Gambar 1. Graf jaringan jalan raya kota di Provinsi Jawa Tengah.

(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-BagianI.pdf>)

B. Jenis-jenis Graf

Berdasarkan bentuk dan ciri-ciri yang terdapat pada suatu graf, graf dapat dibagi menjadi beberapa jenis, yang pertama yaitu berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf:

1. Graf sederhana (*simple graph*), graf sederhana merupakan graf yang tidak mengandung gelang ataupun sisi ganda.
2. Graf tak-sederhana (*unsimple-graph*), graf tak-sederhana merupakan lawan dari graf sederhana, yaitu graf yang mengandung sisi ganda atau gelang.

Graf tak-sederhana juga dapat dibedakan lagi menjadi:

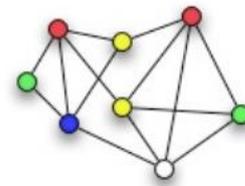
1. Graf ganda (*multi-graph*), graf yang mengandung sisi ganda.
2. Graf semu (*pseudo-graph*), graf yang mengandung sisi gelang.

C. Terminologi Graf

Terminologi atau istilah-istilah yang digunakan pada teori graf antara lain:

1. Ketetanggaan (*adjacent*), dua buah simpul saling bertetanggaan apabila kedua simpul tersebut dihubungkan oleh sebuah sisi.

2. Berisian (*incidency*), sebuah sisi dikatakan berisian dengan dua buah simpul apabila kedua simpul tersebut dihubungkan oleh sisi tersebut.
3. Simpul terpencil (*isolated vertex*) ialah simpul yang tidak mempunyai sisi yang berisian dengannya.
4. Graf kosong (*null graph* atau *empty graph*) ialah graf yang himpunan sisinya adalah himpunan kosong atau graf yang di dalamnya tidak terdapat sisi sama sekali.
5. Derajat (*degree*), derajat dari suatu simpul berarti jumlah sisi yang berisian dengan simpul tersebut.
6. Lintasan (*path*) merupakan kumpulan dua buah simpul yang menunjukkan lajur pada suatu graf, simpul pertama menunjukkan simpul asal, simpul kedua menunjukkan simpul tujuan. Lintasan biasa digunakan untuk menunjukkan langkah-langkah dari suatu simpul ke simpul yang lain pada graf.
7. Siklus (*cycle*) atau sirkuit (*circuit*) merupakan lintasan pada suatu graf dengan simpul awal yang sama dengan simpul akhir atau tujuan yang berarti lintasan tersebut kembali ke simpul yang sama.
8. Keterhubungan (*connected*), dua buah simpul dikatakan terhubung apabila terdapat lintasan yang menghubungkan kedua simpul tersebut. Apabila semua simpul dalam graf tersebut saling terhubung, maka graf tersebut merupakan graf terhubung (*connected graph*). Sedangkan, apabila terdapat simpul yang tidak terhubung dengan simpul lain yang berada pada graf tersebut, maka graf tersebut merupakan graf tak-terhubung (*disconnected graph*).
9. Upagraf (*upagraph*) dan Komplemen Upagraf. Upagraf berarti subset dari suatu subgraf. Sedangkan komplemen upagraf berarti bagian pada graf yang bukan termasuk dari upagraf tersebut.
10. Upagraf merentang (*spanning subgraph*) merupakan sebuah upagraf yang mengandung semua simpul pada graf sebelumnya.
11. Cut-set merupakan himpunan sisi pada suatu graf yang apabila sisi tersebut dihilangkan, graf tersebut akan berubah menjadi graf tak-terhubung.
12. Graf berbobot (*weighted graph*) merupakan graf yang di setiap sisinya mengandung nilai angka.



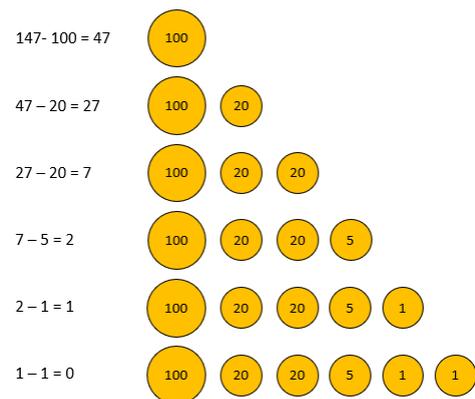
Gambar 2. Pewarnaan graf.

(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>)

E. Algoritma Greedy

Algoritma berarti langkah-langkah yang harus dilakukan untuk menyelesaikan suatu permasalahan dengan baik. Algoritma juga biasa digunakan untuk menjelaskan suatu metode efektif yang dapat diekspresikan dengan rangkaian terbatas yang bersih intruksi-intruks yang terdefinisi dengan baik [2].

Algoritma greedy merupakan salah satu algoritma yang digunakan untuk memecahkan suatu permasalahan optimasi. Permasalahan optimasi merupakan permasalahan yang dibutuhkan jawaban paling optimum untuk menyelesaikan permasalahan tersebut. Langkah-langkah algoritma greedy mencerminkan dari kata *greedy* yang dalam Bahasa Inggris artinya rakus atau tamak. Pada setiap langkah yang kita pilih, terdapat banyak pilihan yang perlu untuk dievaluasi, dan kita perlu memilih langkah yang terbaik dari setiap langkah yang ada. Prinsip greedy adalah “*take what you can get now!*” yaitu pada setiap langkah, algoritma membuat pilihan optimal lokal. Dan dari langkah-langkah tersebut diharapkan bahwa langkah-langkah selanjutnya akan membawa ke solusi yang optimal



Gambar 3. Penerapan algoritma greedy pada penukaran uang logam.

Salah satu contoh penerapan algoritma greedy adalah permasalahan meminimalkan hasil penukaran uang kertas menjadi uang logam seperti yang ditunjukkan pada gambar 3. Jika tersedia uang pecahan logam sebesar 1, 5, 20, 50, 100, maka uang kertas senilai 147 dapat ditukar dengan cara berikut.

Strategi yang digunakan dalam penyelesaian permasalahan ini adalah dengan memilih uang logam dengan nilai terbesar dan apabila uang yang tersisa tidak mencukupi, maka tukar dengan uang logam yang lebih kecil, sehingga langkah-langkah yang perlu diambil adalah sebagai berikut:

1. Pilih satu koin bernilai 100 (Total = 100)

2. Pilih satu koin bernilai 20 (Total = 100 + 20 = 120)
3. Pilih satu koin bernilai 20 (Total = 100 + 20 + 20 = 140)
4. Pilih satu koin bernilai 5 (Total = 100 + 20 + 20 + 5 = 145)
5. Pilih satu koin bernilai 1 (Total = 100 + 20 + 20 + 5 + 1 = 146)
6. Pilih satu koin bernilai 1 (Total = 100 + 20 + 20 + 5 + 1 + 1 = 147)

Solusi: Jumlah koin minimum = 6 (solusi optimal minimum).
 Algoritma greedy disusun oleh elemen-elemen berikut [3]:

1. Himpunan kandidat berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi berisi kandidat-kandidat yang terpilih menjadi solusi dari permasalahan.
3. Fungsi seleksi adalah pemilihan kandidat yang paling memungkinkan untuk mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi kelayakan adalah pemeriksaan apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.
5. Fungsi obyektif merupakan fungsi untuk memaksimalkan atau meminimumkan nilai solusi, contohnya yaitu memaksimalkan keuntungan penjualan, meminimumkan biaya produksi, dan lain-lain.

III. ANALISIS PERSOALAN

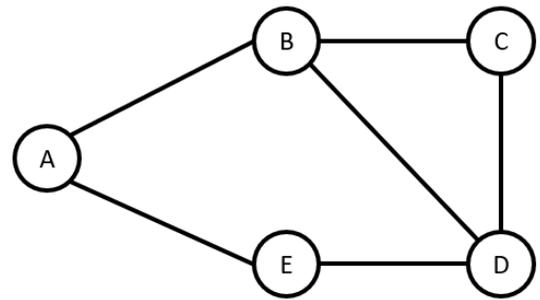
A. Pewarnaan Graf

Setelah kita memahami apa itu algoritma greedy dan bagaimana prinsip dan langkah-langkah yang digunakan. Kita perlu menerapkan algoritma greedy ini dalam penerapan pewarnaan graf. Langkah yang perlu dilakukan untuk menerapkan algoritma greedy dalam menyelesaikan permasalahan pewarnaan graf adalah:

1. Beri warna pada simpul pertama dengan warna pertama.
2. Untuk simpul-simpul yang tersisa pada graf, pilih salah satu simpul dan beri warna dengan warna yang sebelumnya telah digunakan, apabila warna tersebut telah digunakan pada simpul yang bertetangga dengan simpul tersebut, gunakan warna yang baru.

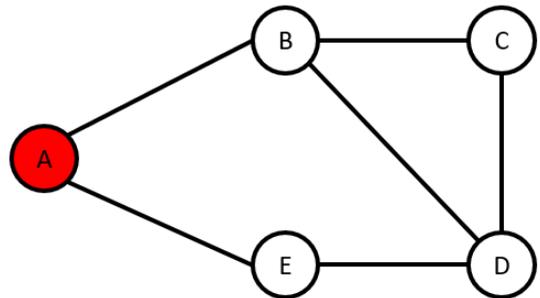
B. Contoh Penerapan

Setelah kita mengetahui langkah-langkah yang harus dilakukan untuk menyelesaikan permasalahan pewarnaan graf menggunakan greedy algorithm, kita dapat menerapkan algoritma tersebut pada sebuah graf. Langkah-langkah yang perlu dilakukan adalah sebagai berikut:



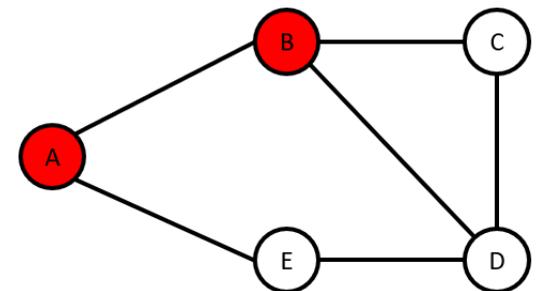
Gambar 4. Graf contoh untuk persoalan pewarnaan graf.

1. Pilih salah satu simpul pada graf, dan beri warna pertama pada simpul tersebut.



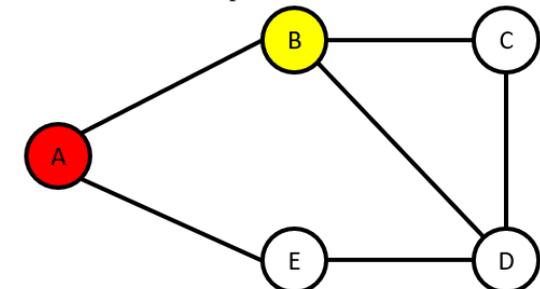
Gambar 5. Graf setelah diberi warna pertama.

2. Pilih simpul lain pada graf tersebut yang belum di beri warna, lalu beri warna dengan warna yang telah dipilih sebelumnya atau warna pertama.



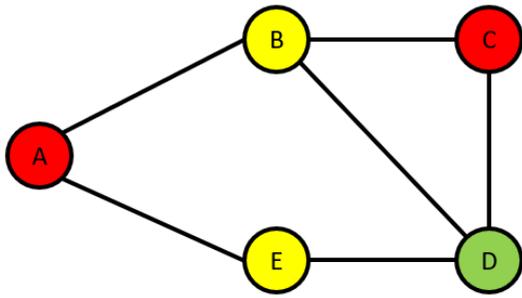
Gambar 5. Graf setelah diberi warna kedua.

3. Setelah itu kita cek apakah simpul yang bertetangga memiliki warna yang sama dengan simpul tersebut, apabila memiliki warna yang sama pilih warna selanjutnya, jika belum ada warna selanjutnya, pilih warna baru untuk simpul tersebut.



Gambar 6. Graf setelah langkah ketiga.

4. Lakukan yang sama pada simpul C, D, dan E.



Gambar 7. Graf dengan pewarnaan graf.

Setelah dilakukan penyelesaian menggunakan algoritma greedy untuk permasalahan pewarnaan graf pada graf diatas, kita peroleh bilangan kromatik dari graf tersebut adalah 3.

C/ Analisis dan Pembahasan

Perlu kita ketahui bahwa algoritma greedy dalam penyelesaian pewarnaan graf ini tidak menjamin kita akan mendapatkan solusi paling efisien atau optimal. Akan tetapi, penggunaan algoritma ini pasti tidak akan mendapatkan solusi lebih dari derajat terbesar ditambah 1 pada graf tersebut. Hal ini dikarenakan dalam pengerjaan algoritma greedy, kita memperhitungkan warna-warna yang dimiliki pada simpul-simpul yang bertetanggan dengan simpul yang ingin diberi warna, apabila setiap warna diberi angkat 1, 2, ..., maka nilai yang diperoleh tidak akan melebihi derajat terbesar ditambah 1.

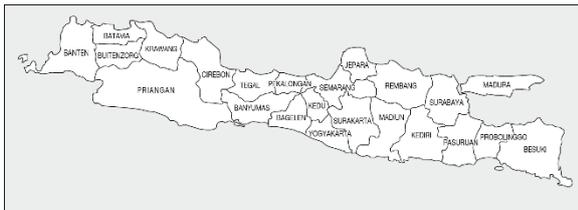
IV. APLIKASI

A. Pewarnaan Peta

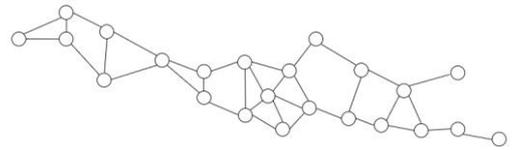
Peta merupakan gambaran permukaan bumi pada suatu bidang datar dengan skala tertentu Daerah-daerah di dunia seperti negara atau provinsi juga diberi batas-batas. Ada beberapa macam cara menyajikan peta dunia, salah satu caranya adalah pada sebuah bidang datar dan setiap daerah diberi warna untuk membedakan antara daerah yang satu dengan yang lainnya. Pewarnaan dalam peta inilah merupakan salah satu dari penerapan pewarnaan graf di kehidupan nyata.

Langkah-langkah penyelesaian permasalahan pewarnaan peta adalah:

1. Mengubah peta yang telah ada menjadi sebuah graf. Simpul pada graf ini menunjukkan daerah pada peta tersebut, simpul ini yang nantinya akan kita beri warna. Lalu sisi pada graf ini menunjukkan bahwa kedua simpul atau daerah tersebut berdempetan.

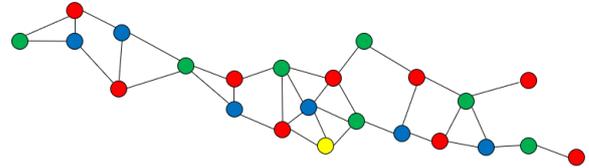


Gambar 8. Peta Pulau Jawa pada masa Penjajahan. (Sumber: <https://www.abundancethebook.com/wp-content/uploads/2019/10/peta-jawa-3.png>)



Gambar 9. Graf dari peta Pulau Jawa pada gambar 8.

2. Menerapkan algoritma greedy pada graf yang telah dibuat sebelumnya.



Gambar 10. Graf setelah diberi warna.

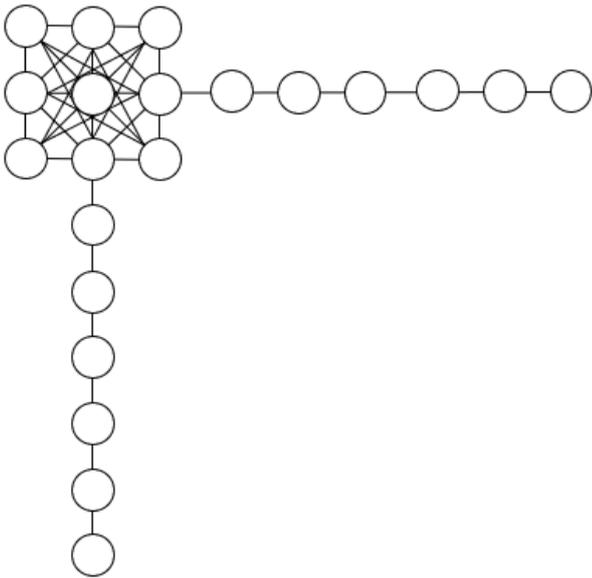
B. Sudoku

Sudoku merupakan salah satu jenis permainan teka-teki logika menggunakan angka. Permainan ini merupakan sebuah papan permainan, dengan kotak sebanyak 9x9 yang terdiri dari 9 kotak dengan 3x3. Tujuan dari permainan ini adalah untuk mengisi semua kotak yang ada dengan angka 1 hingga 9 akan tetapi di dalam satu bagian kotak (berisi 3x3 kotak) dan dibaris serta kolom yang sama tidak boleh terdapat angka yang sama.

		8		3				
	2		9				1	5
	9	4				8	6	
2		9	6	8		4	7	1
6		1			4			
5		7	3	9	1	2	8	
	1					5		
				3	2		4	8
		2	8	5				

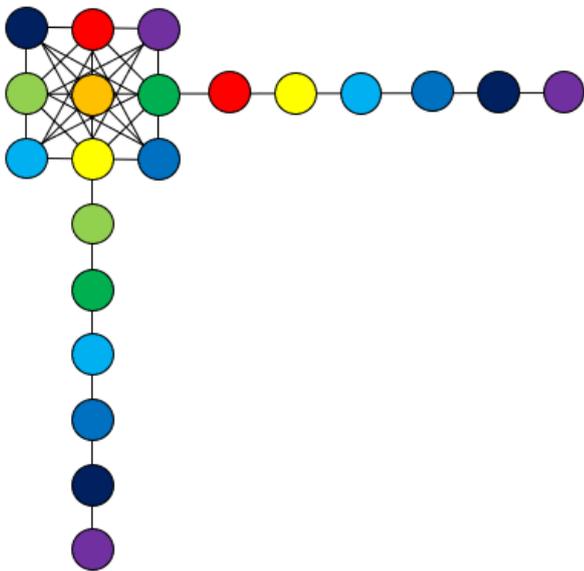
Gambar 11. Sudoku. (Sumber: <https://sudoku.com/>)

Kita menerapkan pewarnaan graf pada permainan sudoku, dengan cara merepresentasikan permainan sudoku ke dalam graf. Simpul yang ada pada graf melambangkan setiap kotak pada sudoku, sedangkan sisi yang ada pada graf melambangkan kotak-kotak yang tidak boleh memiliki angka yang sama.



Gambar 12. Graf dari permainan sudoku.

Graf diatas melambangkan salah satu bagian pada sudoku yang tidak boleh memiliki angka yang sama. Maka langkah selanjutnya adalah dengan menerapkan algoritma greedy pada graf tersebut.



Gambar 13. Graf setelah diberi warna.

C. Art Gallery Problem

Art gallery problem merupakan permasalahan dunia nyata yang terkait jumlah paling minimum penjaga yang diperlukan untuk menjaga suatu galeri. Peta dari galeri direpresentasikan dengan poligon sederhana, dan dari poligon tersebut dibentuklah graf. Simpul pada graf merepresentasikan letak penjaga, sedangkan sisi menandakan tidak ada dinding yang menghalangi antara dua simpul.

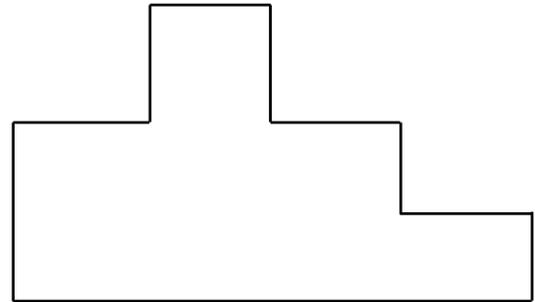
Sebelum menyelesaikan permasalahan tersebut, ada beberapa hal yang perlu didefinisikan di dalam permasalahan ini. Penjaga pada permasalahan ini tidak dapat bergerak bebas dan hanya bisa diam saja di tempat, namun ia dapat berputar dari posisinya,

sehingga dapat melihat ke seluruh sudut yang terlihat dari sisinya. Dan tentu saja, penjaga tidak dapat melihat melewati dinding yang ada pada galeri, dinding yang ada di galeri diasumsikan merupakan dinding yang tidak tembus pandang.

Sisi yang merepresentasikan penglihatan dari suatu penjaga pada permasalahan ini hanya bisa pada garis lurus dari dua simpul. Oleh karena itu, sisi pada graf tidak boleh berbelok, wajib tegak lurus.

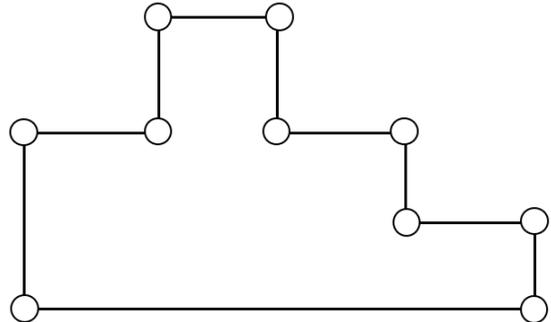
Setelah mendefinisikan penjaga dari persoalan ini, langkah-langkah yang dibutuhkan untuk menyelesaikan permasalahan ini:

1. Mengubah peta dari galeri ke dalam bentuk poligon.



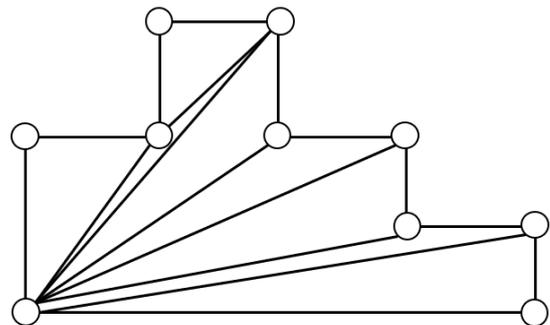
Gambar 14. Peta galeri.

2. Memilih titik-titik untuk dijadikan lokasi penjaga. Disini kita memilih sudut-sudut yang ada pada poligon, dikarenakan di sudut tersebut lah tempat yang paling efisien.



Gambar 15. Graf dari peta galeri.

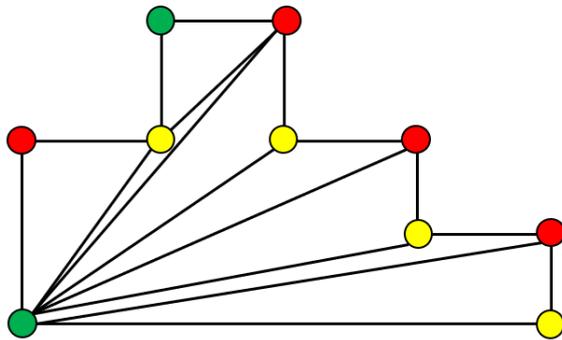
3. Menghubungkan simpul simpul yang tidak terhalang oleh tembok, disini kita membuat bentuk poligon menjadi segitiga segitiga kecil.



Gambar 16. Graf setelah diberi sisi.

4. Menerapkan pewarnaan graf pada graf yang telah dibentuk, menggunakan algoritma greedy yang

sebelumnya telah dibahas.



Gambar 17. Graf setelah diberi warna.

Berdasarkan graf yang diperoleh di atas, jumlah minimum penjaga yang dibutuhkan untuk menjaga galeri tersebut adalah dua.

V. KESIMPULAN

Teori graf memiliki potensi yang besar untuk merepresentasikan berbagai permasalahan yang ada di dunia. Salah satu permasalahan yang ada pada graf, pewarnaan graf, juga memiliki berbagai macam penerapannya di dunia nyata, seperti pewarnaan peta, permainan sudoku, dan *art gallery problem*. Tak hanya pewarnaan graf, terdapat juga permasalahan lain yang melibatkan graf, seperti mencari jalur terpendek dan graf planar.

Pada makalah ini penulis menggunakan algoritma greedy untuk menyelesaikan permasalahan pewarnaan graf, dan seperti yang telah dijelaskan sebelumnya algoritma greedy tidak selalu menghasilkan solusi paling efisien dalam menyelesaikan permasalahan ini. Oleh karena itu, masih terbuka ruang untuk mencari bagaimana algoritma terbaik dan paling efisien untuk menyelesaikan permasalahan ini.

VI. UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada kehadiran Allah SWT, Tuhan yang Maha Esa, karena berkat rahmat, hidayah serta karunia-Nya lah penulis dapat menyelesaikan makalah ini tepat pada waktunya

Tak lupa juga, penulis mengucapkan terima kasih sebesar-besarnya kepada seluruh pihak yang telah turut membantu dalam penulisan dan penyelesaian makalah ini. Terima kasih banyak kepada Bapak Dr. Ir. Rinaldi Munir, M.T. dan dosen pengajar mata kuliah IF2120 Matematika Diskrit, atas bimbingan yang telah beliau berikan kepada penulis. Terima kasih juga kepada seluruh tim pengajar mata kuliah IF2120 Matematika Diskrit, dan yang terakhir terima kasih juga kepada rekan-rekan yang selalu memberi semangat dan membantu penulis dalam penyelesaian makalah ini.

REFERENCES

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>, diakses pada 13 Desember 2021
- [2] H. Rogers, *Theory of Recursive Functions and Effective Computability*, MIT Press, Cambridge Massachusetts, 1987. diakses pada tanggal 14 Desember 2021.

- [3] 5 Graph Theory, 5.8 Graph Coloring, https://www.whitman.edu/mathematics/cgt_online/book/section05.08.html, diakses pada tanggal 13 Desember 2021.
- [4] <https://brilliant.org/wiki/guarding-a-museum/>, diakses pada tanggal 14 Desember 2021.
- [5] <https://www.geeksforgeeks.org/graph-coloring-set-2-greedy-algorithm/>, diakses pada tanggal 13 Desember 2021.
- [6] J. Ahmad. Penerapan Algoritma Greedy pada Penjadwalan Produksi Single-Stage dengan Parallel Machine di Industri Konveksi. Oktober 2015, diakses pada tanggal 14 Desember 2021.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 14 Desember 2020

Muhammad Akmal Arifin - 13520037