

# APLIKASI ALGORITMA *GREEDY* PADA PEWARNAAN PETA

Muhammad Amrimirza-13506003

Program Studi Teknik Informatika,  
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung  
Jalan Ganesha No.10 Bandung  
e-mail: if16003@students.if.itb.ac.id

## ABSTRAK

Algoritma *Greedy*, merupakan salah satu bentuk algoritma yang ada dari sekian banyak algoritma yang berkembang pada saat ini. Algoritma *Greedy* merupakan salah satu bentuk algoritma yang paling dasar dan cenderung mudah diaplikasikan pada beberapa kasus masalah. Dalam pengembangannya, algoritma *greedy* ini kemudian memiliki berbagai macam bentuk, semuanya disesuaikan agar terbentuk sebuah algoritma *greedy* yang paling efisien dalam mengerjakan suatu masalah tertentu.

Pewarnaan peta merupakan salah satu contoh masalah yang dapat diselesaikan dengan menggunakan algoritma *greedy*. Seperti yang kita ketahui, pada sebuah peta tergambar wilayah-wilayah yang dimiliki antar Negara. Dan untuk membedakan wilayah antar negara tersebut, digunakan beberapa warna yang berbeda-beda.

Maka dengan algoritma *greedy*, kita dapat melakukan pewarnaan peta dengan hanya menggunakan jenis warna seminimal mungkin tanpa adanya wilayah Negara yang saling terbaur dalam peta tersebut.

**Kata kunci:** algoritma *greedy*, graf, pewarnaan peta.

## 1. PENDAHULUAN

Pada zaman sekarang ini, manusia sudah banyak menggunakan jasa komputasi untuk menyelesaikan masalah yang dihadapinya. Hal ini dikarenakan, ketimbang manusia menyelesaikan masalahnya sendiri, akan jauh lebih efisien jika masalah tersebut diselesaikan dengan bantuan computer. Hal ini dikarenakan, cara kerja computer dalam menyelesaikan masalah relatif lebih cepat dan sangat kecil terjadinya unsur "error". Pada kesempatan ini, penulis akan membahas salah satu masalah yang dapat diselesaikan dengan teknik komputasi, yaitu pewarnaan peta.

Peta merupakan sebuah media yang digunakan untuk menggambarkan muka Bumi. Pada zaman sekarang ini, peta tidak hanya digambarkan pada media kertas saja,

melainkan pada media digital, seperti komputer. Dalam hal pebuatannya, sudah sangat lumrah pada saat ini jika peta dibuat dengan menggunakan jasa computer. Penulis akan membahas masalah yang dibahas, yaitu meliputi proses pewarnaan peta ketika pembuatan dilakukan.

Dalam masalah pewarnaan peta ini akan digunakan jenis algoritma dasar, yaitu algoritma *greedy*. Dengan memanfaatkan strategi algoritmik, diharapkan algoritma yang terbentuk merupakan algoritma yang tidak hanya menghasilkan solusi saja, melainkan sebuah algoritma yang menghasilkan solusi terbaik, dengan waktu kerja seefektif mungkin. Pada kasus ini solusi terbaik tercapai jika pada pewarnaan peta hanya digunakan jenis warna seminimal mungkin.

## 2. METODE

Dalam menyelesaikan masalah pewarnaan peta kali ini akan digunakan skema dasar algoritma *greedy*. Dan untuk mempermudah merepresentasikan peta yang akan diwarnai akan digunakan pemodelan dalam bentuk graf.

### 2.1 Dasar Teori Algoritma *Greedy*

Algoritma *Greedy* merupakan algoritma yang menghasilkan solusi melalui penyelesaian langkah perlangkah (*step by step*) dengan menerapkan 2 hal berikut pada tiap langkahnya:

1. Pilihan yang diambil merupakan pilihan terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensinya ke depan nanti, hal ini bersesuaian dengan prinsip Algoritma *Greedy* yaitu "*take what you can get now*".
2. Berharap dengan memilih pilihan terbaik saat itu (**optimum lokal/local optimum**) dapat mencapai solusi terbaik dari permasalahan yang dihadapi (**optimum global/global optimum**). Dalam algoritma *Greedy* diasumsikan bahwa optimum lokal merupakan bagian dari optimum global.

Sedangkan untuk aplikasinya algoritma *Greedy* digunakan untuk pemecahan yang memerlukan solusi

yang bersifat **optimum**, yakni solusi yang memiliki **nilai terbesar (maksimum)** atau **nilai terkecil (minimum)**.

## 2.2 Komponen Algoritma Greedy

Pada umumnya persoalan yang dipecahkan dengan algoritma *Greedy* tersusun oleh:

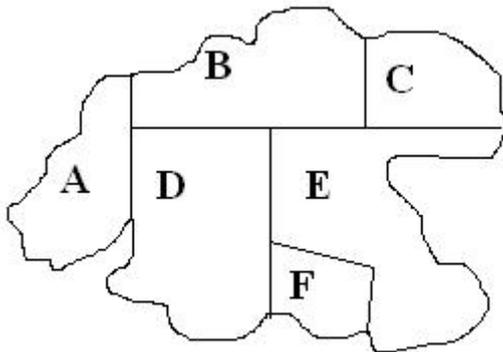
1. Himpunan Kandidat C  
Merupakan himpunan yang berisi elemen pembentuk Solusi.
2. Himpunan Solusi S  
Himpunan yang berisi elemen solusi pemecahan masalah.
3. Fungsi Seleksi  
Fungsi yang memilih kandidat yang paling memungkinkan dari Himpunan Kandidat untuk dimasukkan ke dalam himpunan solusi agar solusi optimal terbentuk. Kandidat yang sudah terpilih pada suatu langkah tidak akan dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi Kelayakan  
Fungsi yang memeriksa apakah suatu kandidat yang terpilih akan menimbulkan solusi yang layak, yaitu kandidat tersebut, bersama dengan himpunan solusi yang terpilih tidak akan melanggar kendala yang berlaku pada masalah.
5. Fungsi Obyektif  
Fungsi yang memaksimalkan atau meminimalkan nilai solusi.

## 2.3 Representasi Peta

Untuk mempermudah pemrosesan dalam komputer, peta akan direpresentasikan dengan pemodelan graf, di mana:

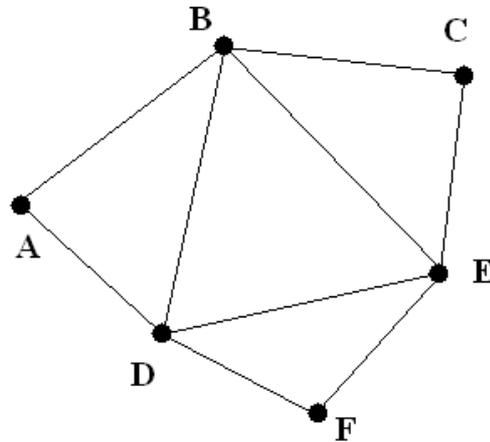
1. Tiap Negara diwakilkan dalam bentuk simpul pada graf
2. Tiap Negara yang saling berbatasan akan digambarkan dengan ketersambungan simpul melalui suatu busur.

Perhatikan ilustrasi berikut:



Gambar 1. Ilustrasi Peta

Bentuk ilustrasi ini akan diubah ke dalam bentuk graf menjadi:



Gambar 2. Bentuk Graf dari Ilustrasi Peta pada Gambar.1

Dalam pewarnaan peta, cukup dilakukan pemberian kode warna pada tiap-tiap simpul yang mewakili tiap negara.

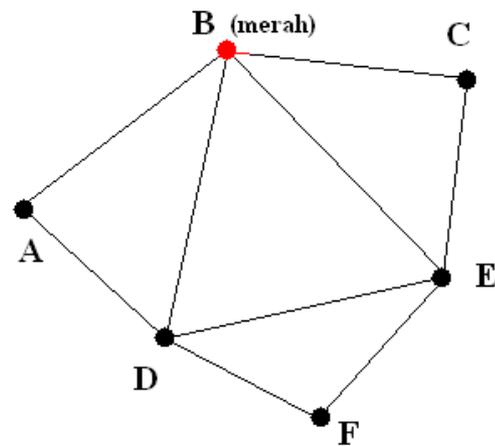
## 2.4 Rancangan Algoritma Greedy

Untuk menemukan solusi optimal dari permasalahan pewarnaan peta ini akan digunakan rancangan algoritma *greedy* yang akan dipaparkan berikut ini. Sedangkan solusi optimal yang diinginkan merupakan sebuah solusi di mana seluruh negara berhasil diwarnai dengan menggunakan warna seminimal mungkin.

Dalam pencarian solusi optimal, akan dirancang komponen dari algoritma *Greedy* sebagai berikut:

1. Himpunan Kandidat  
Dalam kasus ini himpunan kandidat merupakan seluruh jenis warna yang ada yang akan digunakan untuk mewarnai peta.
2. Himpunan Solusi  
Himpunan solusi akan diisi dengan himpunan warna yang sudah digunakan untuk mewarnai peta.
3. Fungsi Seleksi  
Pada kasus ini fungsi seleksi terbagi menjadi 2, yaitu:
  - Fungsi Seleksi simpul  
Fungsi yang menyeleksi simpul mana yang harus dikerjakan terlebih dahulu. Prioritas pengerjaan dilihat dari simpul yang memiliki jumlah busur terbanyak.
  - Fungsi seleksi warna  
Fungsi ini akan memilih warna yang akan digunakan untuk mewarnai simpul pada peta. Penyeleksian warna ini akan dibagi menjadi 2 tahap, yaitu:

- a. Jika layak warna akan diambil dari himpunan solusi, yaitu warna yang sudah dipakai sebelumnya, atau
  - b. Jika tidak satu pun warna dari himpunan solusi layak atau himpunan solusi masih kosong, akan diambil warna dari himpunan kandidat, yaitu warna yang sama sekali belum digunakan.
4. Fungsi Kelayakan  
Fungsi yang akan memeriksa apakah suatu warna layak untuk digunakan pada sebuah simpul. Pemeriksaan kelayakan dapat dilakukan dengan melihat simpul-simpul yang bertetangga dengan simpul yang akan diwarnai. Jika memiliki warna yang sama dengan simpul yang akan diwarnai.
5. Fungsi Objektif  
Dalam kasus ini objektif dari algoritma ini adalah meminimalkan jumlah jenis warna yang digunakan pada proses pewarnaan seluruh simpul.



Gambar 3. Pewarnaan Peta tahap 1

Untuk skema dari algoritma ini akan dijabarkan sebagai berikut:

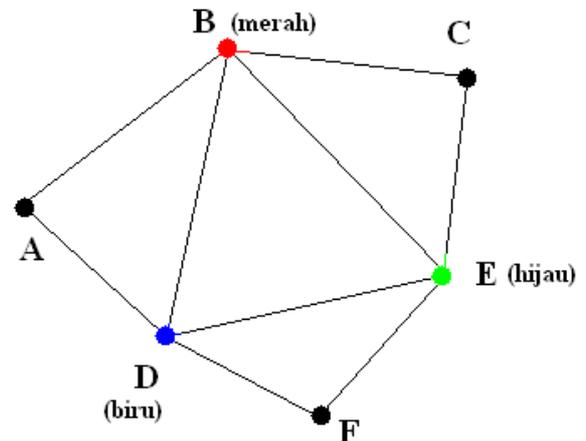
1. Inisiasi himpunan solusi dengan kosong.
2. Melakukan pemilihan simpul yang akan diisi warnanya dengan fungsi seleksi simpul.
3. Memilih kandidat warna dengan menggunakan fungsi seleksi warna. Jika warna diambil dari himpunan kandidat kurangi warna anggota himpunan kandidat dengan warna yang diambil.
4. Periksa kelayakan warna yang dipilih menggunakan fungsi kelayakan. Jika tidak layak, kembali ke langkah 2. Jika layak dimasukkan ke himpunan solusi.
5. Periksa apakah solusi sudah meliputi pewarnaan seluruh simpul. Jika sudah maka berhenti, jika belum kembali ke langkah 2.

4. Karena simpul tetangganya masih kosong maka warna merah ini otomatis dianggap layak.
5. Simpul yang dikerjakan pun lanjut ke simpul D atau E. Untuk kali ini akan diambil simpul D.
6. Ketika diseleksi warna yang akan digunakan, akan dipilih lagi warna merah (sesuai dengan fungsi seleksi). Namun, hal ini tidak cocok dengan fungsi kelayakan. Oleh karena itu, seleksi dilakukan lagi dan akan diambil warna baru dari himpunan kandidat. Dalam kasus ini warna biru.
7. Simpul pun dilanjutkan pada simpul E, simpul yang memiliki busur terbanyak.
8. Dan pada simpul E, hal yang sama dengan yang dilakukan pada simpul D pun dilakukan kembali.

## 2.5 Studi Kasus Pewarnaan Peta

Pada sub-bab ini akan dijelaskan bagaimana jalannya algoritma ketika mengerjakan sebuah kasus pewarnaan peta. Untuk contoh kasusnya akan digunakan peta pada gambar.1. Berikut akan dijelaskan tahap-tahapnya:

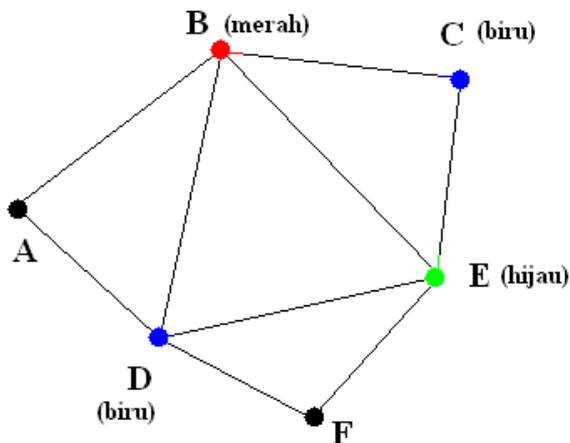
1. Melakukan Inisiasi
2. Pada tahap ini akan dipilih simpul awal yang akan dikerjakan. Berdasar seleksi simpul akan dipilih simpul B, D, atau E. Pada kali ini akan diambil simpul B.
3. Karena solusi masih kosong, maka akan dipilih warna langsung dari himpunan kandidat. Untuk saat ini akan digunakan warna merah. Maka bentuk graf akan menjadi seperti berikut:



Gambar 3. Pewarnaan Peta tahap 2

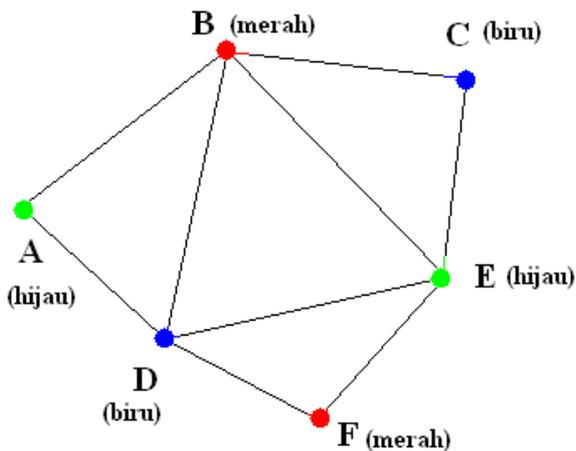
9. Dari simpul E, akan dilakukan seleksi simpul. Kali ini kandidatnya adalah simpul A, C, atau F. Kali ini akan diambil simpul C.
10. Pada simpul ini ketika diperiksa apakah ada warna yang sudah terdaftar dalam himpunan yang masih

layak digunakan, ternyata didapatkan warna biru layak untuk digunakan.



Gambar 3. Pewarnaan Peta tahap 3

11. Proses ini diulang kembali pada simpul A dan F. Sehingga akan menghasilkan bentuk akhir sebagai berikut:



Gambar 4. Hasil Akhir Pewarnaan Peta

Berdasarkan algoritma yang dibentuk kali ini didapatkan pewarnaan peta dengan menggunakan 3 warna yang berbeda.

#### IV. KESIMPULAN

1. Pemanfaatan algoritma *greedy* pada kasus pewarnaan peta dapat dikatakan cukup efektif. Hal ini dikarenakan tiap optimum lokal (pewarnaan pada satu simpul) yang dicapainya akan langsung mengarah pada optimum global (pewarnaan seluruh simpul) solusi secara keseluruhan.

2. Algoritma *Greedy* pada pewarnaan peta ini memiliki langkah yang sangat krusial pada heuristik pemilihan simpul yang akan diwarnai. Hal ini dikarenakan, jika keterurutan pemilihan simpul salah akan mengakibatkan terbentuknya solusi yang tidak optimal.

#### REFERENSI

- [1] Munir, Rinaldi, 2006, *Diktat Kuliah IF2251 Strategi Algoritmik*, Institut Teknologi Bandung.

Nama, "Judul Buku", Penerbit, Tahun terbit.