

# Penggunaan Algoritma Divide and Conquer Dalam Pewarnaan Graf

Desfrianta Salmon Barus - 13508107  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
If18107@students.itb.ac.id

**Abstrak** - Algoritma divide and conquer adalah salah satu algoritma yang seringkali digunakan dalam memecahkan persoalan-persoalan yang bersangkutan dengan jumlah data yang besar. Pada algoritma ini kumpulan data dibagi menjadi sekumpulan data yang lebih kecil dan akan dipecahkan masing-masing, setelah pemecahan untuk setiap sekumpulan kecil data dapat dilakukan, data tersebut akan digabung kembali, dan dengan penggabungan dari solusi data-data yang lebih kecil tersebut solusi pemecahan akhir dari permasalahan akan ditemukan.

Dalam masalah pewarnaan graf, algoritma divide and conquer dapat diterapkan sebagai salah satu solusi pemecahan permasalahan. Dalam pewarnaan graf aturan utama yang ada yaitu harus melakukan pewarnaan berbeda untuk setiap simpul yang bertetangga dengan suatu simpul lainnya, dengan aturan dasar tersebut sebagai dasar utama untuk berpikir algoritma divide and conquer akan disesuaikan seperlunya sehingga dapat memecahkan masalah tersebut.

Masalah utama dalam penggunaan algoritma divide and conquer dalam menyelesaikan permasalahan pewarnaan graf adalah pengelompokan bagian-bagian graf ke dalam pecahan kelompok yang lebih kecil, sehingga dengan demikian pewarnaan dapat dilakukan lebih mudah. Persoalan yang perlu untuk ditangani dalam memecahkan permasalahan pewarnaan dalam graf ini adalah menentukan jumlah warna yang optimal dalam melakukan pewarnaan graf. Dengan melakukan modifikasi seperlunya pada konsep dasar dari algoritma divide and conquer diharapkan permasalahan-permasalahan yang ada tersebut dapat diselesaikan.

**Kata Kunci** – algoritma divide and conquer, graf, simpul, tetangga.

## I. PENDAHULUAN

Graf merupakan sebuah representasi dari data dan hubungan dari setiap data tersebut. Data itu bisa merupakan data apa saja dan bersumber dari mana saja, dan dengan kemudahan yang diberikan melalui graf ini, penggunaan graf terus bertambah seiring waktu sebagai suatu bentuk representasi data yang baik. Graf itu sendiri dapat digambarkan sebagai simpul yang menggambarkan suatu entitas dari data dan garis-garis yang menunjukkan hubungan antar simpul yang satu dengan simpul lainnya.

Salah satu bagian yang menarik dari graf ini adalah pewarnaan graf yang merupakan bagian dari pemberian label dari graf, dalam kasus ini label yang digunakan

adalah warna-warna tertentu. Banyak metode yang digunakan dalam mewarnai graf, namun dalam bagian ini metode yang dibahas adalah metode pewarnaan *edge coloring*. Dalam pewarnaan graf ada beberapa hal yang seringkali menjadi permasalahan utama, yaitu dalam hal jumlah optimal warna yang dibutuhkan untuk mewarnai graf sehingga setiap bagian yang bertetangga akan memiliki warna berbeda dan proses pewarnaan graf itu sendiri, hal ini disebabkan karena dalam pewarnaan graf ada aturan dasar yang harus diperhatikan yaitu untuk setiap simpul yang bertetangga tidak boleh memiliki warna yang sama.

Untuk memecahkan masalah pewarnaan graf yang merupakan salah satu *nonpolynomial-time algorithm* banyak metode algoritma yang sering digunakan. Diantaranya adalah algoritma brute-force, algoritma greedy dan algoritma runut balik. Pada makalah ini pemecahan masalah pewarnaan graf tersebut akan dilakukan dengan algoritma divide and conquer, sebuah algoritma yang berdasar dalam pemecahan suatu masalah menjadi bagian-bagian terkecil untuk dipecahkan. Dengan algoritma ini diharapkan solusi dapat dicapai dengan kinerja yang baik pula.

## II. METODE PENYELESAIAN

Metode yang akan digunakan untuk memecahkan permasalahan pewarnaan graf adalah divide and conquer, algoritma yang berbasis pada pemecahan masalah dengan membagi permasalahan tersebut dalam struktur yang lebih kecil.

### A. Dasar Teori Algoritma Divide and Conquer

Algoritma *Divide and Conquer* adalah sebuah algoritma dengan teknik pemecahan masalah dengan membagi suatu masalah menjadi upa-masalah samapai pada ukuran terkecil, kemudian untuk setiap upa-masalah tersebut dilakukan penyelesaian secara independen masing-masingnya dan akhirnya setiap upa-masalah tersebut akan digabung kembali sehingga dapat menjadi solusi dari masalah awalnya. Algoritma ini disusun oleh

tiga proses utama yaitu *divide* (membagi masalah menjadi upa-masalah yang mirip dengan masalah awal namun ukurannya lebih kecil), *conquer* (menyelesaikan masing-masing upa-masalah), dan *combine* (menggabungkan solusi dari setiap upa-masalah).

Pemecahan masalah dengan menggunakan algoritma *divide and conquer* memberikan dua keuntungan utama, pertama, dengan metode *divide and conquer* ini masalah yang secara konseptual sulit untuk dipecahkan dapat diselesaikan dengan pendekatan yang sederhana karena ukuran masalah direduksi hingga menjadi kasus basis yang mudah untuk diselesaikan. Kedua, kompleksitas dari algoritma ini menjadi semakin kecil bersamaan dengan semakin diketahuinya solusi dari suatu permasalahan.

### B. Skema Umum Algoritma Divide and Conquer

Algoritma *divide and conquer* dicirikan oleh batas ambang dari masukan, ukuran upa-masalah, jumlah upa-masalah, dan algoritma penggabungan.

Batas ambang atau *threshold* merupakan ukuran masukan dimana permasalahan mencapai titik dimana upa-masalah tersebut tidak perlu dibagi lagi, sering juga disebut sebagai nilai basis. Ukuran upa-masalah adalah pembagian masalah dengan rasio pembagian tertentu dari permasalahan yang semula. Rasio yang umum digunakan adalah 2. Bagian tersulit dari algoritma ini adalah bagian algoritma penggabungan dari seluruh upa-masalah untuk memperoleh solusi dari masalah semula.

### C. Rancangan Algoritma Divide and Conquer

Untuk dapat memecahkan permasalahan pewarnaan graf akan digunakan algoritma *divide and conquer* yang akan dimodifikasi seperlunya.

Untuk memecahkan permasalahan pewarnaan graf, bagian yang terpenting untuk diketahui adalah mencari jumlah warna optimal yang akan digunakan untuk mewarnai masing-masing gambar dari graf tersebut. Untuk itu dibutuhkan sebuah fungsi seleksi yang mencari jumlah warna optimal. Hal lain yang penting adalah mencari *pattern* untuk memecah graf tersebut untuk itu diperlukan suatu metode khusus untuk mengecek *pattern* tersebut. Bagian penting terakhir adalah bagian penggabungan dengan suatu metode tertentu. Berikut ini adalah pemaparan lebih lanjut dari komponen-komponen algoritma yang akan digunakan.

#### 1. Himpunan Warna

Berisi kombinasi dari warna yang dapat digunakan untuk mewarnai graf dengan kombinasi tertentu.

#### 2. Himpunan Pola

Berisi pola-pola dalam graf dengan hierarki tersendiri. Bersama dengan informasi pola awal dari keseluruhan.

#### 3. Fungsi Pengakses Simpul

Fungsi ini akan mengakses simpul mana yang akan di akses.

#### 4. Fungsi Pemeriksa Warna

Fungsi ini berfungsi untuk mengecek warna yang sudah pernah diakses sebelumnya dan warna yang masih tersedia dan dapat digunakan dengan suatu aturan tertentu (seperti hierarki warna).

#### 5. Fungsi Penyeleksi Warna

Fungsi ini akan melakukan seleksi terhadap warna yang akan digunakan untuk mewarnai simpul tersebut. Fungsi ini akan mengakses himpunan warna yang ada dan mengeksekusi pewarnaan berdasarkan kesesuaian dengan fungsi pemeriksa warna.

#### 6. Fungsi Optimalisasi Warna

Fungsi ini akan menghasilkan jumlah optimal dari jenis warna yang dapat digunakan untuk mewarnai peta. Jumlah ini dapat diketahui dengan mengecek jumlah simpul yang setiap simpulnya saling berhubungan.

#### 7. Fungsi Pengecek Pola Antar simpul

Fungsi ini adalah fungsi yang akan mencari pola khusus antar simpul yang akan menjadi basis dari pemecahan masalah menjadi upa masalah, pengecekannya mirip dengan optimalisasi warna, namun karena mungkin terdapat pola yang berbeda pola-pola tersebut disimpan dengan hierarki tersendiri.

Secara skematik algoritma yang akan digunakan dapat dijabarkan sebagai berikut:

1. Lakukan pengecekan pola, dan simpan semua pola yang ada dalam himpunan pola dengan hierarki tersendiri dari yang menyangkut simpul terbanyak hingga paling sedikit sambil melakukan pemecahan (langkah *divide*). Urutan dari simpul disusun dengan aturan tertentu.
2. Untuk setiap bagian upa-masalah lakukan pewarnaan dengan fungsi penyeleksi warna yang telah dicek dengan fungsi pemeriksa warna. (langkah *conquer*)
3. Setelah setiap bagian diwarnai, seluruh upa-masalah digabung kembali seperti awalnya dengan panduan himpunan pola, dan solusi permasalahan ditemukan (langkah *combine*).

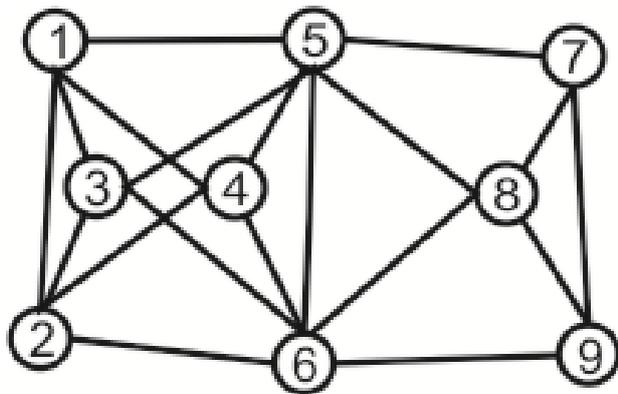
Dengan algoritma tersebut, masalah pewarnaan graf akan dapat terselesaikan.

## III. STUDI KASUS PEWARNAAN GRAF

Untuk dapat membuktikan dan memahami lebih lanjut penggunaan algoritma *divide and conquer* dalam memecahkan permasalahan pewarnaan graf, berikut ini adalah contoh kasus penerapan dari algoritma tersebut dan proses algoritma yang digunakan

### A. Contoh Graf

Graf yang akan digunakan sebagai studi kasus dalam penggunaan algoritma *divide and conquer* dalam memecahkan permasalahan pewarnaan graf adalah sebagai berikut:



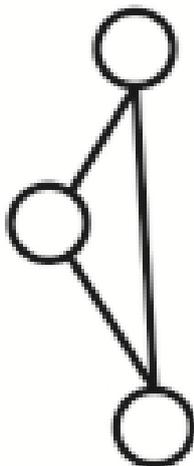
Gambar 1 Contoh permasalahan graf

Graf tersebut terdiri dari 9 simpul, dengan bagian-bagian simpul yang berhubungan dengan beberapa simpul lainnya yang memiliki pola-pola tertentu, untuk pembahasan kinerja algoritma *divide and conquer* dalam menyelesaikan masalah ini akan dijelaskan dibawah ini.

### B. Divide

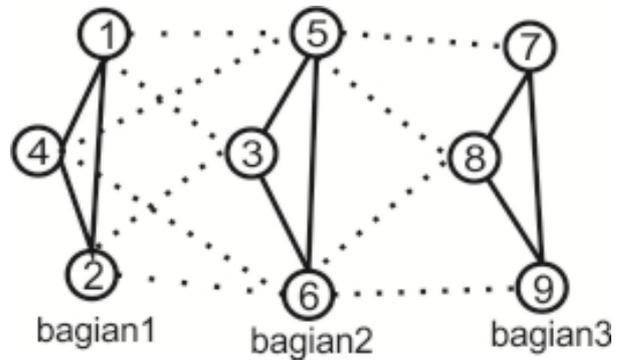
Dengan algoritma optimalisasi warna pada persoalan yang sesuai dengan algoritma diatas akan diperoleh 3 warna yang optimal untuk digunakan, pada kasus ini ketiga warna yang akan digunakan tersebut adalah merah, kuning dan biru.

Pada langkah awal pengecekan pola akan dilakukan, dari pengecekan pola tersebut akan diperoleh pola umum sebagai berikut ini:



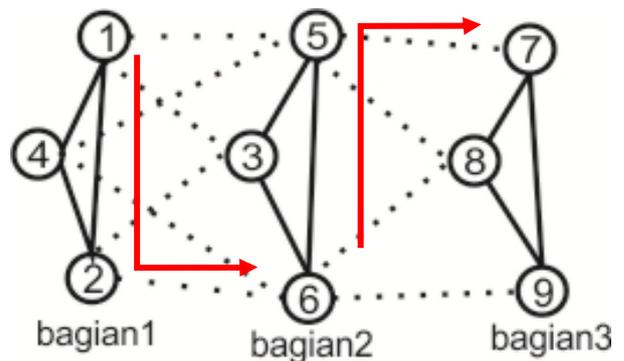
Gambar 2 Pola Graf

Pada contoh tersebut, graf yang digunakan memiliki pola yang sama persis untuk semuanya. Graf tersebut terdiri dari tiga buah pola berulang yang bila dibagi dapat membentuk susunan sebagai berikut ini.



Gambar 3 Hasil pemecahan graf

Hasil dari pemecahan graf tersebut dapat langsung diproses, namun untuk kemudahan dapat dikonversi menjadi sebuah list dengan aturan konversi dari kiri dengan arah bolak-balik dari atas ke bawah ke arah kanan sebagai gambar berikut



Gambar 4 Aturaan konversi menjadi list

Dengan konversi dengan urutan seperti di atas akan diperoleh sebuah list dengan urutan yang dapat dilihat dalam gambar berikut ini:

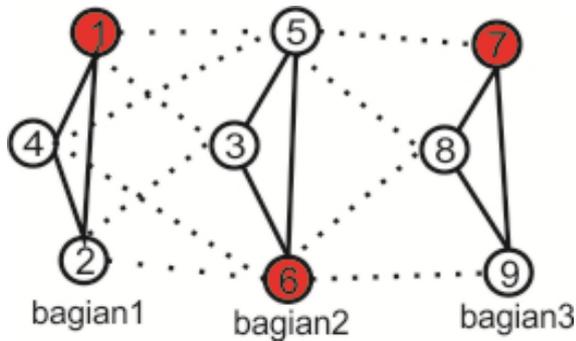


Gambar 5 Hasil konversi menjadi list

Dengan menggunakan hasil konversi list berikut sebagai suatu alternatif, proses *divide* telah selesai dilakukan menjadi 3 buah bagian terpisah.

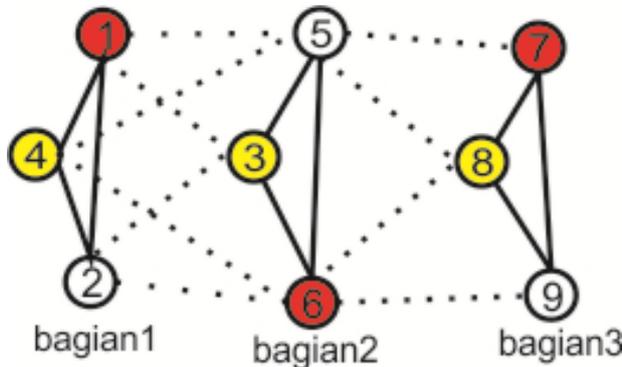
### C. Conquer

Bagian berikutnya adalah penyelesaian setiap bagian permasalahan yang telah dipecah secara independen. Pewarnaan dilakukan dengan urutan warna merah, kuning, biru. Pada awalnya setiap bagian dari yang paling kiri dalam list akan diwarnai dengan merah.



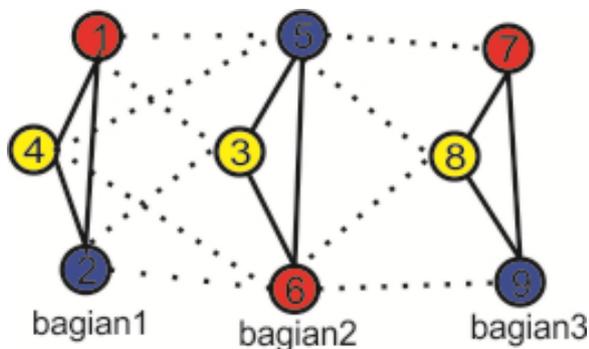
Gambar 6 Pewarnaan merah untuk setiap bagian

Setelah diberi warna merah untuk setiap bagian yang paling kanan dari setiap list tersebut, kemudian dilakukan pemberian warna kuning untuk elemen list yang selanjutnya (sesuai dengan gambar list pada gambar 5), pada graf akan terlihat sebagai berikut:



Gambar 7 Langkah kedua pewarnaan, kuning

Setelah setiap bagian memperoleh warna kuning, maka proses akan dilanjutkan, dan berdasarkan list, proses akan berlanjut pada pewarnaan biru untuk setiap elemen list berikutnya. Proses ini akan menghasilkan suatu hasil sebagai berikut:



Gambar 8 Langkah ketiga pewarnaan, biru

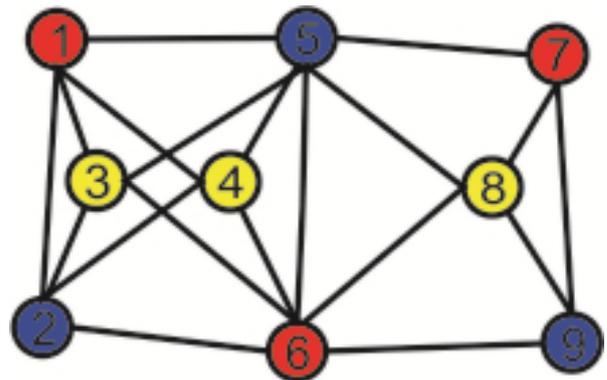
Setelah melalui setiap tahap tersebut maka, penyelesaian untuk setiap bagian kecil dari pemecahan (upa-graf) telah berhasil dilakukan, maka tahap *conquer* pun telah selesai dilakukan.

#### D. Combine

Bagian terakhir dari proses penyelesaian masalah dengan menggunakan algoritma *divide and conquer* adalah tahap *combine* atau penggabungan. Pada bagian ini bagian yang telah dipecah-pecah menjadi bagian yang lebih kecil akan digabungkan kembali menjadi suatu kesatuan dengan suatu mekanisme tertentu.

Pada tahap ini penggabungan dilakukan dengan menghubungkan kembali indeks bagian yang sebelumnya telah dilepas dan disimpan dalam himpunan pola.

Penyatuan kembali dari setiap bagian solusi permasalahan yang sebelumnya telah dipecah menjadi beberapa bagian akan menghasilkan suatu solusi permasalahan yang utuh. Yakni untuk studi kasus ini, solusi dari permasalahan graf tersebut adalah:



Gambar 9 Penyelesaian studi kasus pemecahan permasalahan pewarnaan graf dengan menggunakan algoritma *divide and conquer*

Dengan dibentuknya kembali sebuah graf utuh yang dapat dilihat diatas, dan karena seluruh aturan dasar dalam pewarnaan graf dapat dipenuhi dalam pemecahan tersebut maka dapat dikatakan dalam studi kasus ini solusi permasalahan dari pewarnaan graf dengan menggunakan algoritma *divide and conquer* dapat diperoleh.

#### V. KESIMPULAN

Dengan pemaparan yang telah dilakukan pada bagian-bagian sebelumnya, maka dapat ditarik kesimpulan mengenai algoritma *divide and conquer* khususnya dalam melakukan pemecahan persoalan menyangkut pewarnaan graf yaitu:

1. Algoritma *divide and conquer* adalah salah satu metode yang dapat digunakan dalam memecahkan persoalan berkaitan dengan graf, dan metode ini dapat dikatakan cukup efektif, karena memiliki kinerja yang baik.
2. Metode pemecahan permasalahan pewarnaan graf dengan menggunakan algoritma *divide and conquer* akan semakin cocok untuk digunakan

ketika berhadapan dengan jumlah simpul yang banyak, Karena karakteristik dari algoritma *divide and conquer* yang memecahkan permasalahan hingga mencapai tingkat basis.

## VII. UCAPAN TERIMA KASIH

Melalui bagian ini penulis ingin mengucapkan terima kasih kepada Tuhan YME atas segala berkat yang diberikan, kepada dosen Strategi Algoritma pak Rinaldi Munir atas bimbingannya dalam semester ini, kepada orangtua dan sahabat penulis yang telah banyak mendukung penulis dalam berbagai hal.

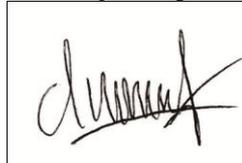
## DAFTAR PUSTAKA

- [1] Ir. Rinaldi Munir, M.T, "Diktat Kuliah IF3051 Strategi Algoritma".

## 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, 29 April 2010



Desfrianta Salmon Barus  
13508107