

# PENERAPAN PEWARNAAN GRAF DALAM PENJADWALAN

## Adventus Wijaya Lumbantobing

Program Studi Teknik Informatika, Institut Teknologi Bandung  
Jalan Ganesha 10, Bandung  
if15112@students.if.itb.ac.id

### ABSTRAK

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek dengan simpul, noktah, bulatan, titik, atau vertex, sedangkan hubungan antara objek dinyatakan dengan garis atau edge.

Pewarnaan graf adalah proses pelabelan setiap simpul dalam graf dengan label tertentu (warna) sehingga tidak ada dua simpul bertetangga yang memiliki warna yang sama.

Pewarnaan graf dapat diaplikasikan dalam berbagai bidang atau masalah. Salah satu aplikasi pewarnaan graf adalah dalam masalah penjadwalan.

Dalam penjadwalan, setiap *job* dinyatakan sebagai simpul dan sisi menggambarkan bahwa kedua *job* yang terhubung oleh sisi tersebut berjalan secara bersamaan (konflik). Tujuan dari penerapan graf adalah agar mengetahui *job* yang konflik (tidak bertetangga).

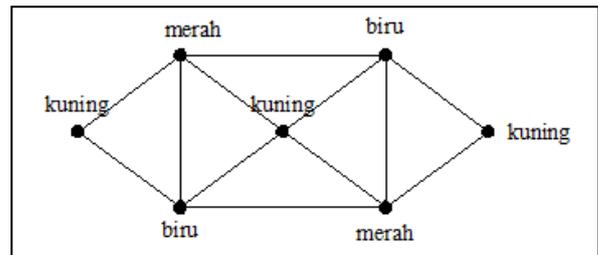
**Kata kunci:** *Pewarnaan graf, penjadwalan.*

## 1. PENDAHULUAN

Pewarnaan graf dapat menjadi suatu metode dalam memecahkan suatu permasalahan misalnya dalam penjadwalan. Terdapat berbagai kasus pewarnaan graf dalam penjadwalan, seperti *precoloring extension*, *list coloring*, *multicoloring*, *minimum sum coloring*.

Suatu graf  $G$  memiliki sebanyak  $k$  warna sehingga setiap simpul yang bertetangga tidak memiliki warna yang sama. Tujuan dari pewarnaan graf adalah untuk mencari jumlah warna minimal yang diperlukan untuk mewarnai graf tanpa adanya konflik antar simpul graf.

Jumlah warna minimum yang dapat digunakan untuk mewarnai simpul disebut bilangan kromatik graf  $G$ .



Gambar 1 Contoh pewarnaan graf dengan 3 warna.

Dalam masalah penjadwalan, penjadwalan dapat diimplementasikan dalam bentuk graf. Setiap simpul menyatakan pekerjaan (*job*) dalam jadwal. Sisi antar dua simpul menyatakan bahwa kedua buah *job* (simpul) tidak dapat dikerjakan secara bersamaan. Warna menunjukkan slot waktu yang tersedia. Setiap *job* membutuhkan satu slot waktu. Jadi dapat dituliskan: simpul  $v$  menerima *job*  $i$  jika dan hanya jika  $v$  dieksekusi dalam waktu  $i$ . Sehingga graf  $k$ -warna berarti semua *job* dapat dikerjakan dalam  $k$  waktu secara tidak simultan.

Bilangan kromatik adalah jumlah minimum warna yang dibutuhkan untuk mewarnai graf. Menentukan bilangan kromatik adalah persoalan *NP-hard* karena tidak dapat mencari solusinya secara efektif untuk graf berukuran besar.

Oleh karena itu ada dua hal yang perlu diperhatikan dalam memodelkan sistem penjadwalan, yaitu:

1. Graf mungkin dapat memiliki suatu struktur khusus sehingga memudahkan dalam pewarnaan.
2. Jika tidak dapat menemukan solusi optimum, maka dapat digunakan metode aproksimasi yang tidak memberikan solusi optimum tetapi setidaknya memberikan performansi yang lebih baik.

## 2. PENJADWALAN DENGAN METOD GRAF

Masalah yang akan dianalisa adalah implemmentasi pewarnaan graf dalam masalah penjadwalan. Ada tiga macam pewarnaan graf, yaitu pewarnaan simpul, pewarnaan sisi, dan pewarnaan wilayah (region).

Pewarnaan simpul adalah memberi warna pada simpul-simpul suatu graf sedemikian sehingga tidak ada dua simpul bertetangga mempunyai warna yang sama. Kita dapat memberikan sembarang warna pada simpul-simpul asalkan berbeda dengan simpul tetangganya.

Dalam persoalan pewarnaan graf, kita tidak hanya sekedar mewarnai simpul-simpul dengan warna berbeda dari warna simpul tetangganya saja, namun kita juga menginginkan jumlah macam warna yang digunakan sesedikit mungkin.

Tabel berikut adalah contoh penjadwalan kuliah dalam graf:

**Tabel 1 Contoh tabel penjadwalan kuliah**

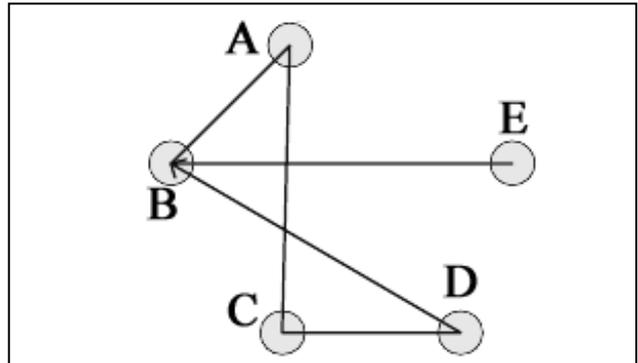
	A	B	C	D	E
1	0	1	0	0	1
2	0	1	0	1	0
3	0	0	1	1	0
4	1	1	0	0	0
5	0	1	0	1	0
6	0	0	1	1	0
7	1	0	1	0	0
8	0	0	1	1	0

Terdapat matriks lima mata kuliah dan delapan orang mahasiswa. Angka 1 pada elemen  $(i,j)$  berarti mahasiswa  $i$  memilih mata kuliah  $j$ , sedangkan angka 0 menyatakan mahasiswa tidak memilih mata kuliah  $j$ .

Berdasarkan tabel diatas, admin mata kuliah ingin menentukan jadwal ujian sedemikian sehingga semua mahasiswa dapat mengikuti ujian mata kuliah yang diambilnya tanpa bertabrakan waktunya dengan jadwal ujian kuliah lain yang juga diambilnya. Jika ada mahasiswa yang mengambil dua buah mata kuliah atau lebih, jadwal ujian mata kuliah tersebut harus pada waktu yang tidak bersamaan. Ujian dua buah mata kuliah dapat dijadwalkan pada waktu yang sama jika tidak ada mahasiswa yang sama yang mengikuti ujian dua mata kuliah itu.

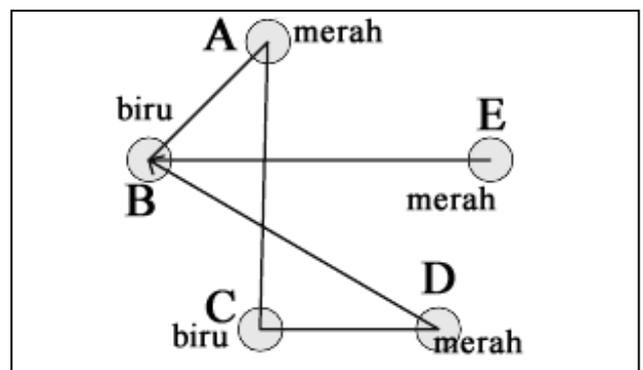
Penyelesaian persoalan menentukan jadwal ujian semua mata kuliah sama dengan menentukan bilangan kromatis graf. Kita dapat menggambarkan graf yang menyatakan penjadwalan ujian. Simpul- simpul pada graf menyatakan mata kuliah. Sisi yang menghubungkan dua buah simpul

menyatakan ada mahasiswa yang memilih kedua mata kuliah itu.



**Gambar 2 Contoh graf penjadwalan mata kuliah**

Berdasarkan graf tersebut kita menyimpulkan, bahwa apabila terdapat dua buah simpul dihubungkan oleh sisi, maka ujian kedua mata kuliah itu tidak dapat dibuat pada waktu yang sama. Warna- warna yang berbeda dapat diberikan pada simpul graf yang menunjukkan bahwa waktu ujiannya berbeda. Diinginkan jadwal ujiannya sesedikit mungkin untuk memudahkan pelaksanaannya. Jadi kita harus menentukan bilangan kromatis graf. Bilangan kromatik untuk graf jadwal ujian tersebut adalah dua. Jadi, ujian mata kuliah A, E, dan D dapat dilaksanakan bersamaan, sedangkan ujian mata kuliah B dan C dilakukan bersamaan tetapi pada waktu yang berbeda dengan mata kuliah A, E, dan D. Berikut ini graf yang telah diwarnai.



**Gambar 3 Contoh pewarnaan graf pada penjadwalan**

## 3. KLASIFIKASI PENJADWALAN

Ada tiga jenis pewarnaan dalam graf yaitu:

1. Pewarnaan simpul (*vertex colouring*), merupakan pemberian warna atau label pada setiap simpul sehingga tidak ada 2 simpul bertetangga yang memiliki warna sama

2. Pewarnaan sisi (*edge coloring*), merupakan pemberian warna pada setiap sisi pada graf sehingga sisi-sisi yang berhubungan tidak memiliki warna yang sama.

3. Pewarnaan wilayah (*region colouring*), merupakan pemberian warna pada setiap wilayah pada graf sehingga tidak ada wilayah yang bersebelahan yang memiliki warna yang sama.

Jumlah warna minimum yang dapat digunakan untuk mewarnai graf dinyatakan dengan bilangan kromatik, yang disimbolkan dengan  $\chi(G)$ .

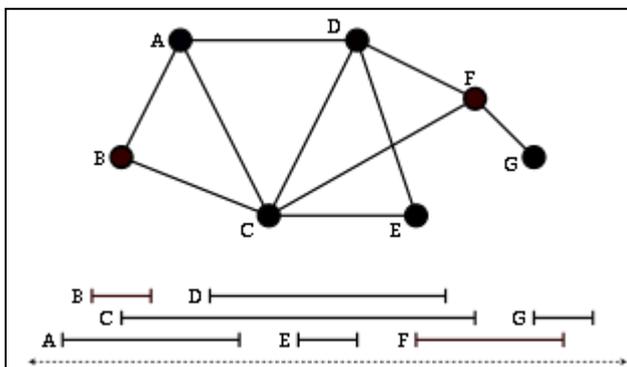
Graf yang memiliki bilangan kromatik 1 adalah graf kosong, yaitu graf yang hanya terdiri dari sebuah simpul. Sementara sembarang graf planar (graf yang dapat digambar di bidang datar tanpa ada sisi yang menyilang di atas sisi lainnya) dapat digambar hanya dengan menggunakan 4 warna.

## 2. KASUS PENJADWALAN DENGAN GRAF

Berikut adalah tipe-tipe kasus [enjadwalan yang dapat diterapkan dengan pewarnaan:

### 2.1 Penjadwalan pesawat

Misalkan terdapat  $k$  pesawat dan  $n$  jumlah penerbangan. Penerbangan ke- $i$  berada pada interval waktu  $(a_i, b_i)$ . Setiap simpul menggambarkan pesawat. Dua buah simpul yang bertetangga menggambarkan bahwa kedua penerbangan dari pesawat tersebut berada pada waktu yang sama. Penjadwalan pesawat merupakan kasus penjadwalan berinterval sehingga masalah ini dapat diselesaikan secara optimal dengan waktu polinomial.



Gambar 4 Penjadwalan pesawat dengan graf interval

Graf interval dapat dituliskan sebagai berikut:

$$\{I_1, I_2, \dots, I_n\} \subset \mathcal{P}(R)$$

menyatakan set interval waktu. Terdapat suatu graf  $G = (V, E)$  dimana:

$$V = \{I_1, I_2, \dots, I_n\}$$

dan

$$\{I_\alpha, I_\beta\} \in E \iff I_\alpha \cap I_\beta \neq \emptyset.$$

### 2.2 Biprocessor task

Misalkan terdapat suatu set prosesor dan sekumpulan *job*. Setiap *job* harus dikerjakan oleh kedua prosesor secara simultan. Suatu prosesor tidak dapat bekerja dengan dua *job* dalam waktu yang sama.

Misalkan simpul menyatakan prosesor dan sisi menyatakan jika ada *job* yang harus dikerjakan pada prosesor  $i$  dan  $j$ . Tujuannya adalah agar setiap warna pada setiap simpul hanya muncul sekali

### 2.3 Frequency assignment

Misalkan terdapat beberapa stasiun radio dalam peta dengan koordinat  $(x, y)$ . Setiap stasiun memiliki frekuensi masing-masing. Namun karena adanya interferensi frekuensi dari stasiun lain maka stasiun dapat menerima gelombang frekuensi yang berbeda. Persoalan ini dapat digambarkan dalam bentuk graf.

## 3. ALGORITMA PEWARNAAN GRAF

Banyak algoritma yang dapat diterapkan dalam pewarnaan graf. Beberapa algoritma-algoritma yang dapat digunakan dalam pewarnaan graf:

### 3.1 Algoritma Greedy

Misalkan terdapat simpul  $v_1, \dots, v_n$ . Algoritma greedy bekerja dengan mewarnai simpul  $v_i$  dengan warna dalam urutan terkecil dan tidak sama dengan warna tetangga. Dan jika diperkenankan dapat menambah warna baru. Algoritma ini bekerja berdasarkan pemilihan urutan warna. Algoritma greedy dapat memberikan hasil yang baik tetapi juga bisa memberikan hasil yang sangat buruk (tidak optimal).

Misalkan dalam contoh graf bipartit yang dapat diwarnai dengan menggunakan dua warna saja, algoritma greedy dapat menggunakan  $n/2$  warna. Jadi dengan graf  $n=8$  pada graf bipartit dapat digunakan 4 buah warna yang berbeda.

### 3.2 Algoritma Brute-Force

Dengan algoritma brute-force, sejumlah  $k$  warna diperiksa terhadap  $n$  buah simpul apakah pewarnaan tersebut ilegal atau tidak.

### 3.3 Parallel and distributed

Graf dapat dimodelkan dengan pemecahan masalah simetri. Dalam graf simetri deterministic distributed algorithm tidak dapat menemukan warna untuk simpul. Oleh karena itu dibutuhkan semacam identifikasi dari tiap simpul misalnya dengan memberikan unique identifier  $\{1,2,3,\dots,n\}$ ,  $n$  adalah jumlah simpul. Tujuannya adalah untuk mengurangi sebanyak  $n$  buah warna menjadi  $\Delta + 1$ .

### 3.3 Welch-Powell

Cara kerja algoritma ini adalah:

1. Urutkan semua simpul berdasarkan derajatnya, dari derajat besar ke derajat kecil.
2. Ambil warna pertama (misalnya merah), warnai simpul pertama yang sudah kita urutkan berdasarkan derajatnya tadi. Kemudian warnai simpul berikutnya yang tidak berdampingan dengan simpul pertama tadi dengan warna yang masih sama (merah).
3. Kemudian lanjutkan dengan warna kedua, dan seterusnya, sampai semua simpul telah diberi warna.

### 3.3 Algoritma runut balik

Runut-Balik atau *backtracking* adalah algoritma yang berbasis pada DFS (Depth First Search) untuk mencari solusi secara lebih mangkus. Runut-balik merupakan perbaikan dari algoritma *brute-force*, secara sistematis mencari solusi persoalan di antara semua kemungkinan yang ada. Hanya pencarian yang mengarah ke solusi saja yang dikembangkan, sehingga waktu pencarian dapat dihemat. Runutbalik lebih alami dinyatakan dalam algoritma rekursif.

Untuk menyelesaikan persoalan pewarnaan graf, perlu membuat sebuah matriks ketetanggaan  $\text{GRAPH}[1..n][1..n]$  yang merupakan *matrix of boolean*.  $\text{GRAPH}(i,j)$  bernilai true jika simpul  $i$  dan  $j$  bertetangga dan bernilai false jika sebaliknya. Matrix of boolean lebih direkomendasikan karena algoritma lebih mementingkan apakah kedua simpul bertetangga atau tidak.

Jenis warna atau bilangan kromatik direpresentasikan dengan tipe bilangan  $1,2,\dots,m$  dan solusi akan direpresentasikan dengan  $n$ -tuple  $\{X(1),X(2),\dots,X(n)\}$  dimana  $X(i)$  bernilai bilangan warna dari simpul  $i$ .

## 4. MASALAH PEWARNAAN GRAF DALAM PENJADWALAN

Berikut adalah masalah-masalah konflik dalam pewarnaan graf pada sistem penjadwalan:

### 4.1 Multicoloring

Dalam *multicoloring* setiap simpul  $v$  memiliki *demand*  $x(v)$  sehingga harus sebanyak  $x(v)$  warna diberikan pada setiap simpul  $v$  dan setiap tetangga memiliki warna yang berbeda. *Multicoloring* dapat diterapkan dalam penjadwalan *job* yang membutuhkan waktu yang berbeda. Ada dua jenis *multicoloring* yaitu:

1. *Non-preemptive* yaitu set warna yang diberikan pada setiap simpul harus memiliki interval yang kontinu. Ini berarti setiap *job* tidak dapat diinterupsi karena waktu yang digunakan secara kontinu.
2. *Preemptive*, pada varian ini *job* dapat diinterupsi karena set warna tidak berdefat kontinu

### 4.2 Precoloring extension

Dalam penjadwalan, jadwal tidak dapat dikontrol secara penuh. Ada saat ketika terdapat suatu *job* yang sudah memiliki waktu kerja yang tertentu (pengambilan slot waktu telah diputuskan sebelumnya dan tidak dapat diubah atau disebut juga *precoloring*).

Misalkan dalam contoh kasus penjadwalan pesawat terbang. Setiap pesawat telah memiliki waktu terbangnya sesuai jadwal. Akan tetapi terdapat waktu *maintenance* untuk setiap pesawat sehingga dalam waktu tersebut pesawat tidak dapat terbang. Kasus ini dapat dimodelkan dengan menggunakan waktu penerbangan *dummy*. Waktu *maintenance* ke- $i$  adalah waktu untuk pesawat ke- $i$ . Masalah ini berupa *NP-complete* graf interval. Tetapi dapat diselesaikan secara polinomial jika setiap warna hanya digunakan dalam masa *precoloring* (setiap pesawat hanya memiliki satu waktu jadwal *maintenance*

### 4.3 List coloring

Dalam *list coloring* setiap simpul  $v$  memiliki sejumlah/set warna yang dapat diberikan. Dan tujuan dari *list coloring* adalah untuk mencari warna yang tepat dari himpunan warna yang tersedia. Model ini dapat diterapkan jika suatu *job* hanya dapat dikerjakan oleh suatu orang/suatu mesin tertentu saja tetapi terdapat beberapa opsi waktu sampai mesin tersebut tersedia.

#### **4.4 *Minimum sum coloring***

Tujuan lain dalam penjadwalan adalah meminimalkan waktu total yang dibutuhkan dalam pengerjaan *job* dalam jadwal. *Minimum sum coloring* mewarnai graf sehingga jumlah warna yang diberikan pada seluruh *job* bernilai minimum.

#### **IV. KESIMPULAN**

Masalah penjadwalan dapat dimodelkan dalam bentuk graf. Untuk mencari solusi konflik dalam penjadwalan dapat digunakan pewarnaan graf. Dalam graf simpul menggambarkan *job* dan sisi menggambarkan hubungan antar *job*. Simpul yang bertetangga menyatakan *job* yang dikerjakan pada waktu yang sama.

#### **REFERENSI**

- [1] Marx, Daniel, "Graph Coloring Problems and Their Applications in Scheduling", 2004.
- [2] Munir, Rinaldi, "Diktat Kuliah IF2153 Matematika Diskrit", 2006.
- [3] [http://en.wikipedia.org/wiki/Graph\\_coloring](http://en.wikipedia.org/wiki/Graph_coloring). Tanggal akses 19 Desember 2009 pukul 16.00 WIB.