

Penerapan Algoritma Backtracking pada Pewarnaan Graf

Deasy Ramadiyan Sari¹, Wulan Widyasari²,
Eunice Sherta Ria³

*Laboratorium Ilmu Rekayasa dan Komputasi
Departemen Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Bandung
(ITB)*

Kampus ITB Jl. Ganesha No.10 Bandung

ifl3008@students.if.itb.ac.id, ifl3055@students.if.itb.ac.id, ifl3057@students.if.itb.ac.id

Abstrak

Teori graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan sampai saat ini. 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*. Salah satu aplikasi yang berkaitan dengan graf adalah pewarnaan graf (*graph colouring*) yang terdiri dari pewarnaan simpul, sisi, dan wilayah. Dalam makalah ini akan dibahas pewarnaan simpul saja. Pewarnaan simpul adalah memberi warna pada simpul-simpul suatu graf sedemikian sehingga tidak ada dua simpul bertetangga mempunyai warna yang sama. Jumlah warna minimum yang dapat digunakan untuk mewarnai graf dinyatakan dengan bilangan kromatik. Pewarnaan graf dapat diselesaikan salah satunya dengan menggunakan algoritma runut-balik (*backtracking*). Algoritma runut-balik adalah algoritma yang berbasis pada *Depth First Search* untuk mencari solusi persoalan secara lebih mangkus.

Kata kunci : *graf, simpul, vertex, edge, graf colouring, algoritma runut-balik, bilangan kromatik.*

1. Pendahuluan

Teori graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan sampai saat ini. 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*. Salah satu aplikasi yang berkaitan dengan graf adalah pewarnaan graf (*graph colouring*).

Pengertian graf menurut Kamus Istilah Teknologi Informasi adalah himpunan simpul yang dihubungkan oleh sisi-sisi.

Sedangkan secara matematis sebuah graf G didefinisikan sebagai pasangan himpunan (V, E) , yang dalam hal ini :

V = himpunan tidak kosong dari simpul-simpul (vertex)
 $= \{v_1, v_2, \dots, v_n\}$

dan

E = himpunan sisi (edge) yang menghubungkan sepasang simpul $= \{e_1, e_2, \dots, e_n\}$

atau dapat ditulis singkat notasi $G = (V, E)$. [1]

2. Pewarnaan Graf

2.1 Pengertian Pewarnaan Graf

Pengertian pewarnaan graf menurut situs en.wikipedia.org/wiki/Graph_coloring adalah pemberian warna, yang biasanya direpresentasikan sebagai bilangan terurut mulai dari 1, pada objek tertentu pada graf. Objek tersebut dapat berupa simpul, sisi, wilayah ataupun kombinasi ketiganya.

2.2 Klasifikasi Pewarnaan Graf

Ada tiga macam pewarnaan graf, yaitu : [bk matdis]

1. Pewarnaan simpul (vertex colouring), merupakan pemberian warna atau label pada setiap simpul sehingga tidak ada 2 simpul bertetangga yang memiliki warna sama.[9]
2. Pewarnaan sisi (edge coloring), merupakan pemberian warna pada setiap sisi pada graf sehingga sisi-sisi yang berhubungan tidak memiliki warna yang sama[8].
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. [5]

3. Algoritma Runut-Balik

3.1 Pengertian 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. Runut-balik lebih alami dinyatakan dalam algoritma rekursif.[2]

3.2 Prinsip Pencarian Solusi dengan Metode Runut-Balik

Langkah-langkah pencarian solusi pada pohon ruang status yang dibangun secara dinamis :[2]

1. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti metode pencarian mendalam (DFS). Simpul-simpul yang sudah dilahirkan dinamakan **simpul hidup** (*live node*). Simpul hidup yang sedang diperluas dinamakan **simpul-E** (*Expand-node*). Simpul dinomori dari atas ke bawah sesuai dengan urutan kelahirannya.
2. Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi maka simpul-E tersebut “dibunuh” sehingga menjadi **simpul mati** (*dead node*). Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan **fungsi pembatas** (*bounding function*). Simpul yang sudah mati tidak akan pernah diperluas lagi.
3. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian diteruskan dengan membangun simpul anak yang lainnya. Bila tidak ada lagi simpul anak yang dapat dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut-balik ke simpul hidup terdekat (simpul orang tua). Selanjutnya simpul ini menjadi simpul-E yang baru. Lintasan baru dibangun kembali sampai lintasan tersebut membentuk solusi.
4. Pencarian dihentikan bila kita telah menemukan solusi atau tidak ada lagi simpul hidup untuk runut-balik.

4. Algoritma Backtracking untuk Pewarnaan Graf

Untuk menyelesaikan persoalan pewarnaan graf, kita perlu membuat sebuah matriks ketetanggaan $GRAPH[1..n][1..n]$ yang merupakan *matrix of boolean*. $GRAPH(i,j)$ 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]

```

procedure MCOLORING(k)
//k adalah indeks selanjutnya dari simpul //yang akan
diberi warna
global integer m,n,X(1..n)
boolean GRAPH[1..n][1..n]
integer k
loop //bangkitkan semua kemungkinan
//untuk X(k)
call NEXTVALUE(k) //set X(k) dengan
//warna yang
//memungkinkan
if (X(k)=0) then exit //warna sudah habis
endif
if (k=n)
then print(X)
else call MCOLORING(k+1)
endif
repeat
end MCOLORING

```

```

procedure NEXTVALUE(k)
global integer m,n,X(1..n)
boolean GRAPH[1..n][1..n]
integer j,k
loop
X(k)-(X(k)+1) mod (m+1) //warna
//berikutnya
If (X(k)=0) then
return endif //warna sudah
//habis
for (j=1 to n) do
if (GRAPH(k,j) and (X(k)=X(j)))
//jika (k,j) adalah sebuah sisi dengan
//simpul awal dan akhir berwarna sama
then exit
endif
if j=n+1 then
return endif
repeat
end NEXTVALUE

```

5. Analisis Algoritma Runut Balik untuk Pewarnaan Graf

Dengan menggunakan algoritma runut-balik yang rekursif, algoritma runut balik untuk pewarnaan graf menghasilkan pohon ruang status dengan derajat m dan tinggi $n+1$. Setiap simpul pada level i memiliki m anak yang ekuivalen dengan jumlah kemungkinan warna untuk $X(i)$, $1 \leq i \leq n$. Simpul pada level $n+1$ adalah daun.

Prosedur MCOLORING dimulai dengan menginisialisasi matriks ketetanggaan, mengisi larik

X dengan 0 dan memanggil MCOLORING(1). Prosedur NEXTVALUE menghasilkan kemungkinan warna untuk $X(k)$ setelah $X(1)$ melalui $X(k-1)$ didefinisikan. Kalang utama dari MCOLORING secara berulang mengambil elemen dari kemungkinan yang ada, mengisikannya ke $X(k)$ dan memanggil MCOLORING secara rekursif.[4]

Pada prosedur MCOLORING, didapat bahwa jumlah simpul internal dalam pohon ruang status adalah $\sum m^i$ dari $i = 0$ sampai n . Untuk setiap simpul internal, NEXVALUE menghabiskan $O(mn)$ waktu untuk menentukan anak-anak yang ekuivalen untuk kemungkinan warna. Jadi, total waktu untuk algoritma runut balik untuk pewarnaan graf adalah

$$n(m^{n+1} - 1) / (m-1) = O(nm^n). \quad (1)$$

6. Perbandingan Algoritma Backtrack dengan Algoritma Lainnya

Saat ini sudah ditemukan algoritma lainnya yang dapat digunakan untuk mewarnai graf. Salah satunya adalah algoritma Brelaz. Algoritma ini menggunakan prinsip heuristic dengan sedikit algoritma runut balik. Adapun prinsip heuristic yang digunakan dalam algoritma ini adalah dengan memilih simpul yang belum diberi warna dan memiliki sisa warna paling sedikit, kemudian subgraf yang belum memiliki warna dilepaskan dan warna dapat dipilih secara acak. Algoritma ini sangat efektif tapi performanya pada fase boundary sangat bervariasi bahkan untuk graf yang sama karena heuristic tidak selalu mencapai pilihan yang unik.

Walaupun begitu, kompleksitas waktu Algoritma Brelaz adalah $O(m \log n)$ dengan m adalah jumlah sisi dan n adalah jumlah simpul.[11]

Terlihat bahwa dengan algoritma Brelaz waktu yang dibutuhkan untuk menyelesaikan persoalan pewarnaan graf cukup singkat dibandingkan dengan algoritma runut-balik. Namun dengan menggunakan algoritma runut-balik dapat dicapai hasil pilihan yang unik, tidak seperti algoritma Brelaz.

7. Kesimpulan

Algoritma runut balik dapat diterapkan pada permasalahan pewarnaan graf. Algoritma ini dapat diimplementasikan untuk sembarang graf dengan n simpul dan m jumlah warna. Algoritma ini juga dapat digunakan untuk menguji apakah benar sebuah graf planar dapat diwarnai hanya dengan 4 warna.

Kompleksitas algoritma ini adalah $O(nm^n)$ yang sebenarnya termasuk tidak mangkus untuk jumlah n

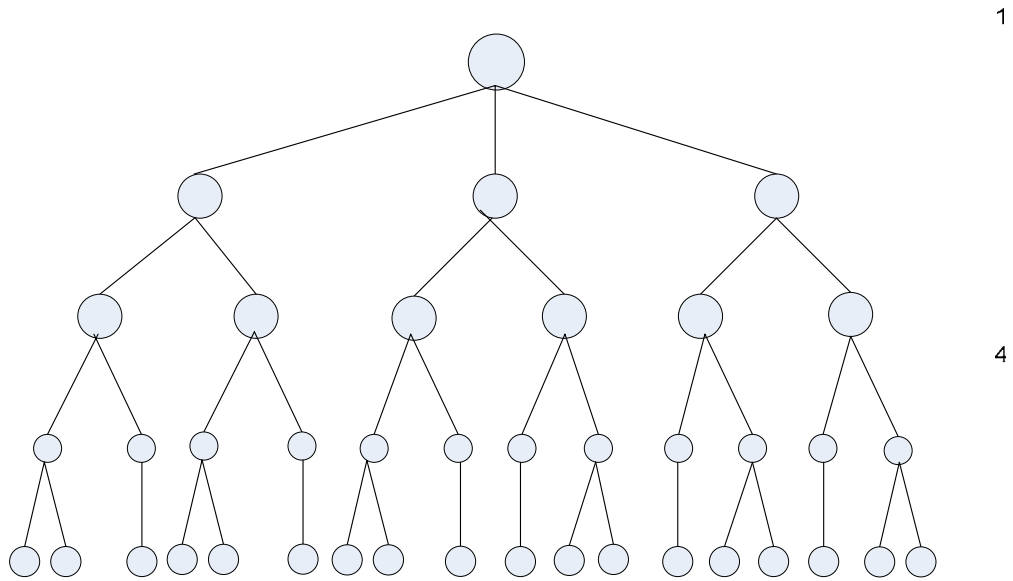
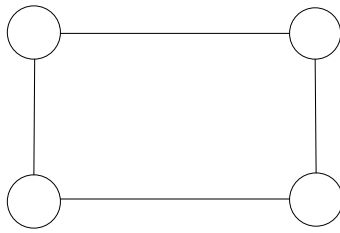
simpul yang besar. Tapi untuk n kecil, algoritma ini cukup efektif.

Kelebihan dari algoritma ini adalah algoritma ini selalu dapat mencapai hasil yang unik.

Daftar Pustaka

- [1] Munir, Rinaldi. 2004. Diktat Kuliah Matematik Diskrit. Departemen Teknik Informatika ITB
- [2] Munir, Rinaldi. 2005. Diktat Kuliah Strategi Algoritmik. Departemen Teknik Informatika ITB
- [3] Wahid, Fathul. Kamus Istilah Teknologi Informasi. Yogyakarta : Penerbit Andi
- [4] Horowitz, Ellis and Sartij Sahni. 1978. Fundamental of Computer Algorithmic. McGraww
- [5] Deo, Narsingh. Graph Theory with Applications to Engineering and Computer Science. Prentice Hall
- [6] Read, Ronald C. Graph Theory and Computing. Academic Press

- [7] <http://www.geom.uiuc.edu/~zaremba/graph1.html>. Diakses tanggal 15 Mei 2005 pukul 13.00 WIB
- [8] Edge Coloring. <http://mathworld.wolfram.com/EdgeColoring.html>. Diakses tanggal 19 Mei 2005 pukul 11.00 WIB
- [9] Vertex Coloring. <http://mathworld.wolfram.com/VertexColoring.html>. Diakses tanggal 19 Mei 2005 pukul 11.00 WIB
- [10] en.wikipedia.org/wiki/Graph_coloring . Diakses tanggal 16 Mei 2005 pukul 10.00 WIB
- [11] Brelaz Algorithm. <http://www.arl.wustl.edu/~jst/pubs/joa88.ps>. Diakses tanggal 17 Mei 2005 pukul 13.00 WIB



Pohon ruang status untuk graf dengan 4 simpul dan 3 kemungkinan warna

1

2

3

1

1

3

1

2

2

3

2

3

2

2

3

3

1₅

3

1

