Pemanfaatan Algoritma Sequential Search dalam Pewarnaan Graf untuk Alokasi Memori Komputer

Vivi Lieyanda - 13509073
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
vivi.lieyanda@students. itb.ac.id

***Abstract* —** *Graf dapat digunakan sebagai model dalam berbagai bidang, dimana dapat merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Banyak hal yang dapat diimplementasikan dengan teori graf, antara lain lintasan terpendek, persoalan pedagang keliling, persoalan tukang pos Cina, dan pewarnaan graf. Aplikasi graf yang akan dibahas adalah pewarnaan graf. Pewarnaan graf ini dapat diaplikasikan ke dalam berbagai bidang atau masalah. Salah satu contohnya adalah aplikasinya dalam alokasi memori komputer. Memori komputer dapat dihemat dengan teori perwarnaan graf. Penghematan tersebut dapat dilakukan dengan menggunakan sebuah algoritma yang disebut dengan Algoritma Sequential Color.*

***Kata Kunci* :** *graf, algoritma sequential color, alokasi memori*

I. PENDAHULUAN

Teori graf berasal dari bidang ilmu matematika. Meskipun demikian, aplikasi dari graf ini dapat dihubungkan dengan berbagai ilmu di bidang lainnya hingga dalam kehidupan sehari-hari. Sepertinya, setiap bidang ilmu dapat dikaitkan dengan teori graf ini. Mulai dari jejaring sosial, hingga alokasi memori komputer dapat diselesaikan dengan bantuan graf. Berbagai algoritma yang berkaitan dengan graf mulai dikembangkan. Salah satunya adalah Algoritma *Sequential Color.*

Pewarnaan graf merupakan salah satu kajian yang cukup menarik dan memiliki banyak aplikasi dalam kehidupan sehari-hari. Salah satu aplikasinya yang penting adalah dalam alokasi memori komputer.

Dalam mengeksekusi sebuah program, komputer menyimpan instruksi-instruksi dalam bahasa mesin. Di dalam instruksi-instruksi yang disimpan, tersimpan pula variabel-variabel yang digunakan dalam setiap instruksi. Variabel yang tersimpan tersebut tidaklah digunakan semuanya secara sekaligus. Bahkan hanya ada sebuah variabel yang hanya dipanggil sekali. Hal ini menyebabkan akan ada variabel yang tidak digunakan dalam beberapa instruksi. Penyimpanan variabel ini dapat memakan banyak memori. Dalam kajian matematika diskrit, teori graf memberikan solusi untuk permasalahan ini, yaitu dengan proses pewarnaan graf. Dengan algoritma *Sequential Color*, memori-memori tersebut dapat dikurangi, sehingga akan menghemat alokasi memori dalam komputer.

II. TEORI

2.1. Teori Graf

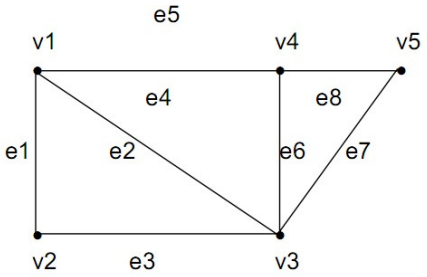
Secara informal, suatu graf adalah himpunan benda-benda yang disebut simpul (*vertex* atau *node*) yang terhubung oleh sisi (*edge*) atau busur (*arc*). Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan simpul) yang dihubungkan oleh garis-garis (melambangkan sisi) atau garis berpanah (melambangkan busur). Suatu sisi dapat menghubungkan suatu simpul dengan simpul yang sama. Sisi yang demikian dinamakan gelang (*loop*). Secara matematis, graf G didefinisikan sebagai pasangan himpunan (V, E), yang dalam hal ini, V adalah himpunan tidak kosong dari simpul-simpul (vertices atau node). Dan E adalah himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul.

$V = \{ v_1, v_2, v_3, ..., v_n\}$
$E = \{ e_1, e_2, e_3, ..., e_n\}$.

Graf G dapat ditulis singkat dengan notasi G = (V, E). Dari definisi di atas, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, namun harus mempunyai minimal satu buah simpul.

Simpul pada graf dapat dinomori dengan huruf ataupun dengan bilangan, ataupun keduanya. Sedangkan dalam penamaan sisi, biasanya dilakukan dengan lambang e_1, e_2, e_3,.... e menyatakan dua buah simpul yang dihubungkan oleh sisi e tersebut. Jika e adalah sisi yang menghubungkan simpul v_i dan simpul v_j, maka e dapat ditulis dengan $e = (v_i, v_j)$.

Graf pada gambar 1 di bawah diperlihatkan himpunan simpul V dan himpunan sisi E.
$V = \{ v_1, v_2, v_3. v_4. v_5\}$
$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$
Sisi $e_1 = (v_1, v_2)$, $e_2 = (v_1, v_3)$, $e_3 = (v_2,v_3)$, $e_4 = (v_1,v_3)$, dan seterusnya.



Gambar 1 Contoh Graf

2.2. Beberapa Terminologi Dasar Graf

Dalam pewarnaan graf, terdapat beberapa terminologi dasar yang perlu dipahami.

2.2.1. Bertetangga (*Adjacent*)

Dua buah simpul pada graf tak berarah G dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain, v_i bertetangga dengan v_j jika (v_i, v_j) adalah sebuah sisi pada graf G. Dari gambar 1, dapat dilihat bahwa simpul v_3 dan v_5 bertetangga karena dihubungkan oleh sebuah sisi, yaitu sisi e_7. Tetapi, simpul v_2 dan v_4 tidak bertetangga, karena tidak ada sebuah sisi yang menghubungkan langsung kedua simpul tersebut.

2.2.2. Bersisian (*Incident*)

Untuk sembarang sisi e (v_i, v_j), sisi e dikatakan bersisian dengan simpul v_i dan v_j. Pada gambar 1, dapat dilihat bahwa sisi e_6 bersisian dengan simpul v_3 dan v_4. Tetapi e_7 tidak bersisian dengan simpul v_2, karena e_7 menghubungkan simpul v_3 dan v_5.

2.2.3. Lintasan (*Path*)

Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang terbentuk $v_0,e_1,v_1,e_2,...,v_{n-1},e_n,v_n$ sedemikian sehingga $e_1 = (v_0,v_1)$, $e_2 = (v_1,v_2)$,..., $e_n = (v_{n-1},v_n)$ adalah sisi-sisi pada graf G.

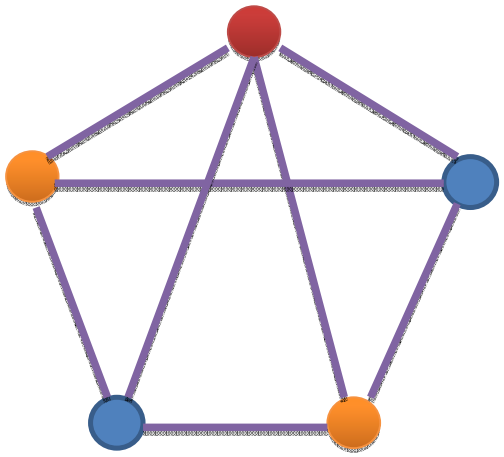
2.2.4. Terhubung (*Connected*)

Graf tak berarah G disebut graf terhubung jika untuk setiap pasang simpul v_i dan v_j di dalam himpunan V terdapat lintasan dari v_i ke v_j (yang berarti juga harus ada lintasan dari v_j ke v_i). Jika tidak, maka G disebut graf tak-terhubung (*disconnected graph*).

2.3. Pewarnaan Graf (*Graph Coloring*)

Pewarnaan graf adalah memberikan warna pada titik-titik pada batas tertentu. Ada tiga macam pewarnaan pada graf, yaitu pewarnaan simpul (*vertex coloring*), pewarnaan sisi (*edge coloring*), dan pewarnaan daerah (*area coloring*). Pewarnaan simpul adalah memberi warna-warna pada simpul suatu graf sedemikian sehingga tidak ada dua simpul bertetangga yang mempunyai warna yang sama. Pewarnaan sisi adalah memberikan warna berbeda pada sisi yang bertetangga sehingga tidak ada dua sisi yang bertetangga yang mempunyai wana yang sama. Sedangkan pewarnaan daerah adalah memberikan warna pada bidang sehingga tidak ada bidang yang bertetangga yang memiliki warna yang sama.

Pada proses Alokasi memori ini, akan digunakan pewarnaan simpul. Graf G dikatakan berwarna n apabila terdapat sebuah pewarnaan graf G dengan menggunakan n warna. Dalam pewarnaan simpul, kita tidak hanya sekedar mewarnai simpul-simpul dengan berbagai warna yang berbeda dari simpul tetangganya, namun kita menginginkan jumlah warna yang digunakan adalah sesedikit mungkin. Jumlah warna minimum yang dapat digunakan untuk mewarnai sebuah graf disebut dengan bilangan kromatik. Sebuah simpul dapat diwarnai dengan sembarang warna asalkan warna yang diberikan untuk simpul tetangganya berbeda dengan warnanya. Pewarnaan graf sehingga diperoleh jumlah warna minimum dapat dilakukan dengan menggunakan Algoritma *Sequential Color.*



Gambar 2 Tiga Buah Warna Cukup untuk Mewarnai Graf Ini

2.4. Algoritma *Sequential Color* (Algoritma Pewarnaan Berurutan)

Algoritma Pewarnaan Berurutan adalah algoritma yang digunakan untuk mewarnai sebuah graf dengan k warna, dimana k adalah bilangan integer positif. Metode yang digunakan adalah pewarnaan graf secara langsung dengan warna sesedikit mungkin.

Adapun *pseudo code* yang digunakan untuk pewarnaan simpul untuk memperoleh warna minimum adalah sebagai berikut.

```
{ input : simpul (v) : v1, v2, v3, ..., vn}
{proses : pewarnaan graf}
{output : warna-warna tiap simpul yang telah diminimisasi}
```

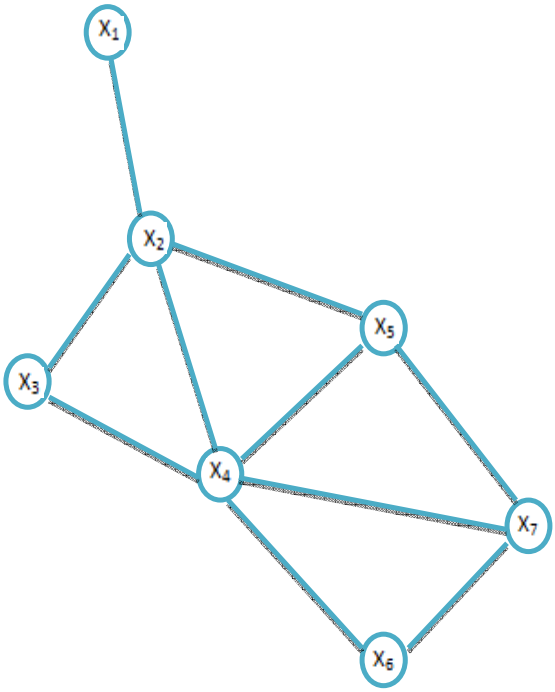
```
KAMUS DATA
        V, E, Li, Ci, Lj : integer
```

ALGORITMA

1. *Start*
2. Masukkan simpul(v) dan sisi (e)
3. Proses Li = 1,...,v
4. Proses i = 1 to v
5. Proses $C_i = X_i$ pada L_i
6. Proses for j = 1 to v
7. Pilih if ((X_i, X_j) ϵ E(G))
8. Proses $L_j = L_j - C_i$
9. Keluaran $X_i..X_v = C_i$
10. Keluaran n = C_v
11. *Stop*

Gambar 3 Pseudo Code Dalam Pewarnaan Simpul

Langkah-langkah pada algoritma sequential color di atas dapat digunakan untuk berbagai macam hal. Salah satu contohnya adalah untuk mewarnai simpul-simpul pada graf berikut ini.



Gambar 4 Salah Satu Contoh Graf

Kita akan mewarnai graf pada gambar 4 dengan menggunakan algoritma *sequential color*. Hal pertama yang harus kita lakukan adalah dengan mendaftarkan simpul-simpul dan sisi-sisi yang ada.

G = (E,V)

V = {X_1, X_2, X_3, X_4, X_5, X_6, X_7}

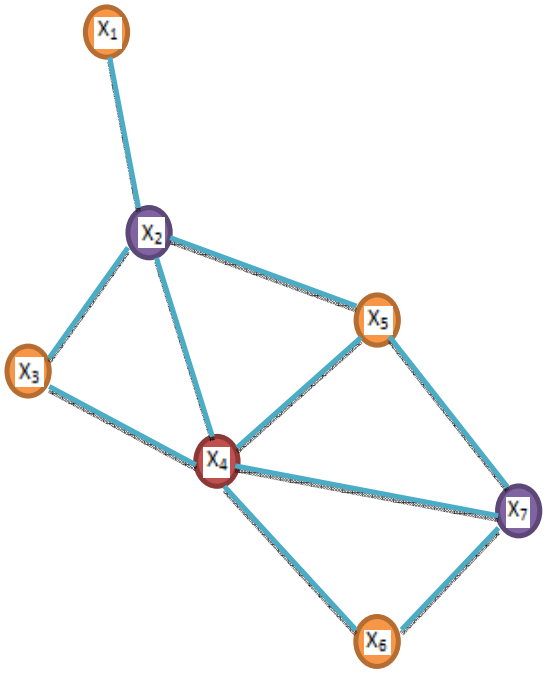
E = { (X_1,X_2), (X_2,X_3), (X_2,X_4), (X_2,X_5), (X_4,X_5), (X_4,X_6), (X_4,X_7), (X_5,X_7), (X_6,X_7) }

Ci = 1,2,3,... (warna-warna yang akan digunakan)

Tabel 1 Langkah-Langkah Pewarnaan Graf

Langkah	i	Li	Ci	j	Lj
3	1	<1>			
3	2	<1,2>			
3	3	<1,2,3>			
3	4	<1,2,3,4>			
3	5	<1,2,3,4,5>			
3	6	<1,2,3,4,5,6>			
3	7	<1,2,3,4,5,6,7>			
4,5	1		1		
6,7,8	1			2	
4,5	2		2		
6,7,8	2			3	<1,3>
				4	<1,3,4>
				5	<1,3,4, 5>
4,5	3		1		
6,7,8	3			4	<2,3,4>
4,5	4		3		
6,7,8	4			5	<1,2,4, 5>
				6	<1,2,4, 5,6>
				7	<1,2,4, 5,6,7>
4,5	5		1		
6,7,8	5			7	<2,3,4, 5,6,7>
4,5	6		1		
6,7,8	6			7	<2,3,4, 5,6,7>
4,5	7		2		

Dari tabel di atas, diperoleh simpul X_1 diwarnai oleh warna 1, simpul X_2 diwarnai oleh warna 2, simpul X_3 diwarnai oleh warna 1, simpul X_4 diwarnai oleh warna 3, simpul X_5 diwarnai oleh warna 1, simpul X_6 diwarnai oleh warna 1, dan simpul X_7 diwarnai oleh warna 2. Jadi, graf pada gambar 3 memiliki bilangan kromatik 3, karena hanya membutuhkan tiga buah warna dalam prosses pewarnaan graf. Misalkan warna 1 adalah warna oren, warna 2 adalah warna ungu, dan warna 3 adalah warna merah. Adapun graf yang telah diwarnai adalah sebagai berikut.



Gambar 5 Graf dari Gambar 4 yang Telah Diberi Warna

2.5. Alokasi Memori

Alokasi adalah sebuah proses menempatkan sejumlah variabel ke dalam sejumlah memori pada CPU. Alokasi dapat terjadi pada sebuah blok dasar, keseluruhan fungsi atau prosedur, atau pada pemanggilan sebuah fungsi. Sebuah kompiler adalah sebuah program komputer yang dapat menerjemahkan satu bahasa komputer ke bahasa lainnya.

Banyak programmer yang menggunakan banyak variabel dengan seenaknya. Bagaimanapun juga, kompiler harus memutuskan bagaimana agar semua variabel-variabel yang digunakan hanya akan memakan memori yang kecil. Tidak semua variabel digunakan dalam waktu yang bersamaan. Dua buah variabel yang berbeda dapat disimpan dalam memori yang sama apabila digunakan dalam waktu yang bebeda. Akan tetapi, dua buah variabel yang digunakan dalam waktu yang bersamaan tidak dapat ditempatkan pada memory yang sama tanpa mengubah nilainya. Variabel yang tidak dapat ditenpatkan di beberapa memori harus disimpan di dalam RAM dan dimuat pada setiap proses baca dan tulis. Mengakses RAM membutuhkan waktu yang lebih lama dibandingkan dengan mengakses memori langsung. Dan itu tentu akan mempengaruhi waktu yang digunakan dalam menjalankan sebuah program. Jadi, sangat dibutuhkan untuk menyimpan variabel ke dalam memori agar tidak disimpan di dalam RAM.

Alokasi memori dapat dimodelkan dengan proses pewarnaan graf. Variabel-variabel berbeda yang digunakan dilambangkan dengan simpul-simpul yang berbeda pada graf. Sisi akan menghubungkan simpul-simpul yang digunakan pada waktu yang bersamaan. Apabila graf dapat diwarnai dengan jumlah minimum k warna, maka variabel-variabel yang digunakan juga dapat disimpan dengan k memori. Ini tentu dapat mengurangi alokasi memori yang digunakan. Dua buah variabel dikatakan saling berinteferensi apabila kedua variabel tersebut digunakan dalam waktu yang bersamaan saat program akan dijalankan. Sebaliknya, kedua variabel dikatakan tidak saling berinteferensi.

III. PENERAPAN ALGORITMA *SEQUENTIAL COLOR*

Dalam pengalokasian memori komputer, dapat digunakan algoritma *sequential color* yang mengaplikasikan pewarnaan graf. Berikut adalah contoh kasus dimana algoritma ini sangat membantu dalam efesiensi penggunaan memori.

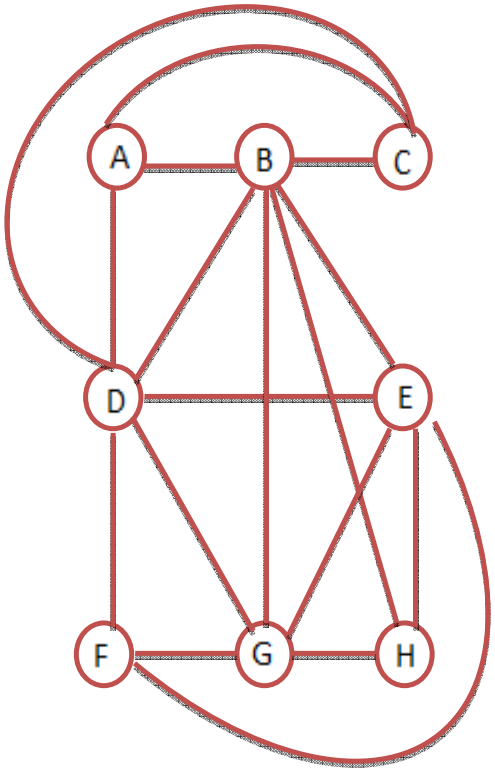
Sebuah algoritma mempunyai langkah-langkah sebagai berikut.

Langkah1 : *Input* A, B, C
Langkah 2: D = A*B
Langkah 3: E = D*C
Langkah 4: *Input* F
Langkah 5: G = D*F
Langkah 6: C := 5
Langkah 7: H= G*C
Langkah 8: *print* E, G, H

Dari langkah-langkah di atas, dapat kita tentukan variabel-variabel yang berinteferensi dan yang tidak berinteferensi. Dua buah variabel dikatakan saling berinteferensi apabila kedua variabel tersebut digunakan dalam waktu yang bersamaan saat program akan dijalankan. F dan C dikatakan tidak saling berintefernsi karena F dan C tidak digunakan dalam waktu yang bersamaan. C digunakan pada langkah 1,2,3,6, dan 7. Sedangkan F hanya digunakan pada langkah 4 dan 5. A dan B dikatakan saling berintefernsi, karena A dan B muncul bersama-sama pada langkah 2.

Dari hubungan interferensi antara simpul-simpul yang muncul bersamaan, maka dapat dibuat graf interfernsi. Simpul-simpul pada graf ini melambangkan variabel-variabel yang digunakan dalam algoritma, sedangkan sisi-sisi menghubungkan simpul-simpul yang berintefernsi.

Dari graf 6 tersebut kita dapat melakukan pewarnaan graf yang akan melambangkan jumlah memori yang akan digunakan dalam alokasi. Untuk mempermudah dalam proses pewarnaan graf, maka A akan diganti dengan 1, B diganti dengan 2, dan seterusnya. Adapun langkah-langkah pewarnaan graf dengan algoritma sequential color dapat dilihat pada tabel 2.



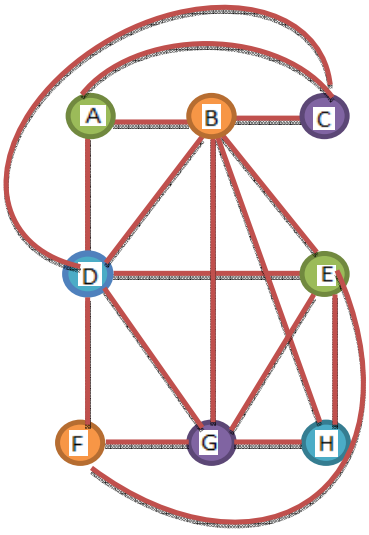
Gambar 6 Graf dari Contoh Kasus Algoritma

Tabel 2 Langkah-Langkah Pewarnaan Graf Berdasarkan Graf pada Gambar 6

Langkah	i	Li	Ci	j	Lj
	1	<1>			
	2	<1,2>			
	3	<1,2,3>			
	4	<1,2,3,4>			
	5	<1,2,3,4,5>			
	6	<1,2,3,4,5,6>			
	7	<1,2,3,4,5,6,7>			
	8	<1,2,3,4,5,6,7,8>			
	1		1		
	1			2	<2>
				3	<2,3>
				4	<2,3,4>
	2		2		
	2			3	<1,3>
				4	<1,3,4>
				5	<1,3,4,5>
				7	<1,3,4,5,6,7>
				8	<1,3,4,5,6,7,8>
	3		3		
	3			4	<1,2,4>
	4		4		
	4			5	<1,2,3,5>
				6	<1,2,3,5,6>
				7	<1,2,3,5,6,7>
	5		1		
	5			6	<2,3,4,5,6>
				7	<2,3,4,5,6,7>
				8	<2,3,4,5,6,7,8>
	6		2		
	6			7	<1,3,4,5,6,7>
	7		3		
	7			8	<1,2,4,5,6,7,8>
	8		4		

Dari Tabel 2 di atas, diperoleh simpul A diwarnai oleh warna 1, simpul B diwarnai oleh warna 2, simpul C diwarnai oleh warna 3, simpul D diwarnai oleh warna 4, simpul E diwarnai oleh warna 1, simpul F diwarnai oleh warna 2, simpul G diwarnai oleh warna 3, dan simpul H diwarnai oleh warna 4. Warna yang dibutuhkan adalah warna 1, 2, 3, dan 4 atau dapat dikatakan bilang kromatiknya sama dengan 4. Sehingga dapat disimpulkan, dari 8 buah variabel yang terdapat dalam algoritma tersebut, kita hanya membutuhkan empat memori untuk menyimpannya.

Graf pada gambar 7 berikut adalah hasil pewarnaan, dimana dimisalkan warna 1 adalah warna hijau, warna 2 adalah warna oren, warna 3 adalah warna ungu, dan warna 4 adalah warna biru.



Gambar 7 Graf Hasil Pewarnaan Berdasarkan Algoritma Sequential Color

IV. KESIMPULAN

Pewarnaan graf (coloring graf) dapat diimplementasikan dalam alokasi memori komputer. Untuk menentukan warna minimum yang diperlukan dalam pewarnaan sebuah graf, dapat digunakan algoritma *sequential color*. Jumlah warna minimum ini menunjukkan jumlah memori yang dibutuhkan. Sebelum ditentukan warna minimum yang diperlukan, harus dibentuk terlebih dahulu graf dari kasus yang ada. Graf ini memiliki simpul yang menunjukkan variabel-variabel yang dibutuhkan dalam algoritma, kemudian sisi yang ada menunjukkan hubungan interferensi dari simpul-simpul. Dua buah simpul dikatakan saling berinteferensi apabila simpul-simpul tersebut digunakan pada saat yang bersamaan dalam suatu instruksi algoritma. Dengan adanya algoritma *sequential color*, pewarnaan graf menjadi lebih mudah dan membantu dalam pengalokasian memori komputer.

V. DAFTAR REFERENSI

[1] M. Rinaldi, "Diktat Kuliah IF 2153 Matematika Diskrit", Program Studi Teknik Informatika, 2006, Bandung, Indonesia..

[2] http://en.wikipedia.org/wiki/Graph_coloring. Tanggal akses : 9 Desember 2010. Pukul 14.35.

[3] http://www.macalester.edu/~hutchinson/book/alberts7.pdf. Tanggal akses : 9 Desember 2010. Pukul 15.30.

[4] http://id.wikipedia.org/wiki/Teori_graf. Tanggal akses : 9 Desember 2010. Pukul 15.45.

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, 17 Desember 2010

Vivi Lieyanda
13509073