

Perbandingan Algoritma Pewarnaan *LDO*, *SDO*, dan *IDO* pada Graf Sederhana

Khairani Permata Sari^{#1}, Armiami^{*2}, Mirna^{*3},

[#]Student of Mathematic Departement State University of Padang

^{*}Lecture of Mathematic Departement State University of Padang

Jl. Prof. Dr. Hamka Air Tawar Padang, 25131, Telp. (0751) 444648, Indonesia

¹khairani_permatasari@yahoo.com

Abstract – Vertex colouring is coloring graph vertex such that two neighbor vertexes have different colors. The main purpose of colouring is to get minimum number of colors which is called chromatic number. In the application of graph coloring, which is the minimum number of colors means minimizing the amount of resources such as space and time in solving a problem. To perform the necessary vertex a coloring tool is an algorithm that will govern how the coloring process on a graph. The sequence of the color on a vertex can affect many colors needed to color all vertexes of the graph. There are three algorithms that can be used to perform the coloring vertex by vertex for the election to be colored, the algorithm *LDO*, *SDO*, and *IDO*. After testing in a few simple graphs, it is concluded that none amongst algorithm *LDO*, *SDO*, and *IDO* which always produces the minimum number of colors.

Keywords – Vertex colouring, Chromatic number, *LDO* algorithm, *SDO* algorithm, *IDO* algorithm

Abstrak – Pewarnaan titik adalah mewarnai titik pada suatu graf sedemikian sehingga dua titik yang bertetangga memiliki warna yang berbeda. Tujuan utama pewarnaan titik adalah mendapatkan banyaknya warna minimum yang biasa disebut bilangan kromatik. Dalam penerapan pewarnaan graf, jumlah warna yang minimum berarti meminimumkan jumlah sumber daya seperti ruang dan waktu dalam menyelesaikan suatu permasalahan. Untuk melakukan pewarnaan titik diperlukan suatu alat bantu yaitu sebuah algoritma yang akan mengatur bagaimana proses pewarnaan pada suatu graf. Urutan dalam pemberian warna pada titik dapat mempengaruhi banyak warna yang diperlukan untuk mewarnai semua titik graf. Ada 3 algoritma yang dapat digunakan untuk melakukan pewarnaan titik berdasarkan cara pemilihan titik yang akan diwarnai, yaitu algoritma *LDO*, *SDO*, dan *IDO*. Setelah dilakukan pengujian pada beberapa graf sederhana, diperoleh kesimpulan bahwa tidak satupun diantara algoritma *LDO*, *SDO*, dan *IDO* yang selalu menghasilkan jumlah warna yang minimum.

Kata kunci – Pewarnaan titik pada graf, Bilangan kromatik, Algoritma *LDO*, Algoritma *SDO*, Algoritma *IDO*

PENDAHULUAN

Pewarnaan graf (*graph colouring*) merupakan salah satu topik dalam teori graf yang telah banyak diaplikasikan dalam berbagai permasalahan, diantaranya; pewarnaan peta, penyusunan jadwal, pemasangan frekuensi radio, dan penyimpanan senyawa kimia berbahaya. Terdapat tiga macam pewarnaan graf yaitu pewarnaan titik (*vertex colouring*), pewarnaan sisi (*edge colouring*), dan pewarnaan wilayah (*face colouring*). Topik yang dibahas pada tugas akhir ini adalah pewarnaan titik.

Pewarnaan titik adalah memberi warna pada titik-titik graf sedemikian sehingga dua titik bertetangga mempunyai warna yang berbeda. Satu cara untuk menjamin dua titik yang bertetangga tidak memiliki warna yang sama adalah menggunakan warna yang berbeda untuk setiap titik. Namun dalam permasalahan pewarnaan graf tidak hanya sekedar mewarnai titik-titik

dengan warna berbeda dari titik tetangganya, tetapi juga diinginkan jumlah warna yang digunakan sesedikit mungkin, karena dalam penerapan pewarnaan graf jumlah warna yang minimum berarti meminimumkan jumlah sumber daya seperti ruang dan waktu dalam menyelesaikan suatu permasalahan [6].

Jumlah warna minimum yang dapat digunakan untuk mewarnai graf dinyatakan dengan bilangan kromatik, yang disimbolkan dengan $\chi(G)$ untuk setiap graf G . Untuk graf dengan sifat tertentu (graf khusus), bisa dengan mudah disebutkan bilangan kromatiknya dan mewarnai titik-titiknya, misalnya graf lengkap dan graf lingkaran. Tetapi umumnya graf yang dihadapi adalah graf yang tidak memiliki keteraturan seperti itu. Karena itu dibutuhkan algoritma yang dapat membantu melakukan pewarnaan graf. Algoritma adalah urutan logis langkah-langkah penyelesaian masalah yang disusun

secara sistematis. Pada pewarnaan graf, algoritma mengatur bagaimana proses mewarnai suatu graf [6].

Banyak algoritma yang telah dikembangkan untuk membantu menyelesaikan pewarnaan graf. Dalam kajian ini akan diberikan tiga algoritma yang dapat digunakan untuk menyelesaikan masalah pewarnaan titik berdasarkan urutan titik yang akan diwarnai yaitu; algoritma *Largest Degree Ordering (LDO)*, *Saturated Degree Ordering (SDO)*, dan *Incident Degree Ordering (IDO)*.

Algoritma *LDO*, *SDO*, dan *IDO* adalah suatu teknik untuk memilih titik awal dan selanjutnya yang akan diwarnai. Algoritma *LDO* merupakan algoritma yang bekerja berdasarkan pada jumlah derajat dari setiap titik. Titik yang memiliki derajat yang lebih besar diwarnai terlebih dahulu. Jumlah derajat titik x ditulis $d(x)$, menyatakan banyaknya sisi yang terkait pada titik x [3]. Algoritma Welch-Powell adalah salah satu algoritma pewarnaan graf yang bekerja berdasarkan teknik *LDO* [2].

Algoritma *SDO* adalah algoritma yang berprinsipkan pada jumlah warna yang berlainan yang ada pada tetangga-tetangga dari sebuah titik. Titik yang memiliki tetangga dengan lebih banyak warna, akan diwarnai terlebih dahulu. Derajat saturasi titik x ditulis $deg_s(x)$, menyatakan banyaknya warna berbeda dari tetangga-tetangga titik x [4]. Algoritma *IDO* berprinsipkan pada jumlah tetangga yang telah diwarnai dari suatu titik. Titik yang lebih banyak bertetangga dengan titik yang telah diwarnai akan diwarnai terlebih dahulu [1]. Derajat *incident* titik x ditulis $deg_i(x)$, menyatakan banyaknya tetangga yang telah diwarnai dari titik x [5].

Menurut Budayasa [3] urutan dalam pemberian warna pada titik dapat mempengaruhi banyak warna yang diperlukan untuk mewarnai semua titik pada graf. Untuk itu penelitian ini dilakukan untuk melihat perbandingan algoritma *LDO*, *IDO*, dan *SDO* berdasarkan jumlah macam warna yang dihasilkan, sehingga dapat diketahui algoritma yang dapat menghasilkan jumlah warna yang minimum. Untuk itu algoritma-algoritma tersebut diujicobakan pada beberapa graf sederhana dengan kepadatan (*density*) yang berbeda-beda. Kepadatan (*density*) adalah jumlah sisi yang ada dibandingkan jumlah sisi yang dapat dibuat dari titik-titik suatu graf [7].

METODE

Penelitian ini adalah penelitian dasar (teoritis). Metode yang digunakan dalam penelitian ini adalah studi kepustakaan dengan menganalisis teori-teori yang relevan dengan permasalahan yang dibahas.

Langkah-langkah dalam penelitian ini adalah sebagai berikut:

1. Menelaah teori-teori mengenai graf, pewarnaan titik pada graf, dan algoritma *LDO*, *IDO*, dan *SDO*.
2. Menerapkan algoritma *LDO*, *IDO*, dan *SDO* pada beberapa graf sederhana.
3. Membandingkan hasil pewarnaan dari algoritma *LDO*, *IDO*, dan *SDO* berdasarkan jumlah warna yang dihasilkan.
4. Memberikan kesimpulan akhir dari hasil penelitian.

HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas tentang algoritma *LDO*, *SDO*, dan *IDO* serta pengujian algoritma tersebut pada beberapa graf sederhana berorde lima sampai dengan dua puluh dengan kepadatan (*density*) yang berbeda-beda untuk melihat jumlah macam warna yang dihasilkan.

A. Langkah Algoritma *LDO*, *SDO*, dan *IDO*

1. Langkah Algoritma *LDO*

Algoritma *LDO* merupakan algoritma yang prinsipnya berdasarkan pada jumlah derajat dari setiap titik. Pada algoritma ini titik yang mempunyai derajat paling tinggi akan diwarnai terlebih dahulu. Jumlah derajat titik x ditulis $d(x)$, menyatakan banyaknya sisi yang terkait pada titik x . Berdasarkan definisi algoritma *LDO*, langkah-langkah algoritma *LDO* dapat dirumuskan sebagai berikut:

- a. Urutkan titik-titik pada graf dalam derajat menurun.
- b. Warnai titik berderajat tertinggi dengan warna 1. Gunakan warna 1 ini untuk mewarnai semua titik yang mungkin sesuai dengan urutan derajatnya sehingga tidak ada dua titik bertetangga yang mempunyai warna yang sama. Jika semua titik sudah diberi warna, langkah selesai. Jika tidak ambil warna 2.
- c. Warnai titik berderajat tinggi selanjutnya dengan warna 2. Gunakan warna 2 ini untuk mewarnai titik lain yang mungkin dan belum diberi warna sesuai dengan urutan derajatnya dengan syarat dua titik bertetangga tidak boleh berwarna sama. Jika semua titik sudah diberi warna, langkah selesai. Jika belum, ambil warna 3. Demikian seterusnya sampai semua titik diberi warna.

2. Langkah Algoritma *SDO*

Algoritma *SDO* merupakan algoritma yang prinsip kerjanya berdasarkan pada derajat saturasi dari setiap titik. Derajat saturasi adalah jumlah warna berbeda pada tetangga dari suatu titik. Pada algoritma ini titik yang mempunyai derajat saturasi paling besar akan diwarnai terlebih dahulu. Derajat saturasi titik x ditulis $deg_s(x)$, menyatakan banyaknya warna berbeda dari tetangga-tetangga titik x . Berdasarkan definisi algoritma *SDO*, langkah-langkah algoritma *SDO* dapat dirumuskan sebagai berikut:

- a. Pilih sembarang titik pada graf, warnai dengan warna 1.
- b. Perbaharui himpunan titik yang belum diwarnai.
- c. Hitung derajat saturasi semua titik yang belum diwarnai.
- d. Pilih titik dengan derajat saturasi terbesar, periksa apakah titik tersebut bertetangga dengan titik yang telah diwarnai sebelumnya.
 - 1) Jika bertetangga, ambil warna 2.

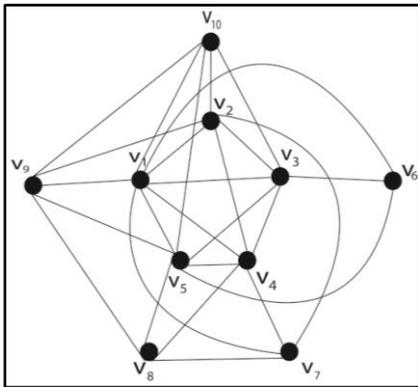
- 2) Jika tidak, ambil warna1, warnai titik yang telah dipilih dengan warna1.
 - e. Ulangi langkah b sampai semua titik diwarnai.
3. Langkah Algoritma *IDO*

Algoritma *IDO* merupakan algoritma yang prinsip kerjanya berdasarkan pada derajat *incident* dari setiap titik. Derajat *incident* adalah jumlah tetangga yang telah diwarnai dari suatu titik. Pada algoritma ini titik yang mempunyai derajat *incident* paling besar akan diwarnai terlebih dahulu. Derajat *incident* titik x ditulis $deg_i(x)$, menyatakan banyaknya tetangga yang telah diwarnai dari titik x . Berdasarkan definisi algoritma *IDO*, langkah-langkah algoritma *IDO* dapat dirumuskan sebagai berikut:

- a. Pilih sembarang titik pada graf, warnai dengan warna 1.
- b. Perbaharui himpunan titik yang belum diwarnai.
- c. Hitung derajat *incident* semua titik yang belum diwarnai.
- d. Pilih titik dengan derajat *incident* terbesar, periksa apakah titik tersebut bertetangga dengan titik yang telah diwarnai sebelumnya.
 - 1) Jika bertetangga, ambil warna2.
 - 2) Jika tidak, ambil warna1, warnai titik yang telah dipilih dengan warna1.
- e. Ulangi langkah b sampai semua titik diwarnai.

B. Contoh Penerapan Algoritma *LDO*, *SDO*, dan *IDO*

Untuk memperjelas langkah algoritma *LDO*, *SDO*, dan *IDO* diberikan sebuah contoh graf seperti pada gambar 1 untuk menerapkan algoritma-algoritma tersebut.



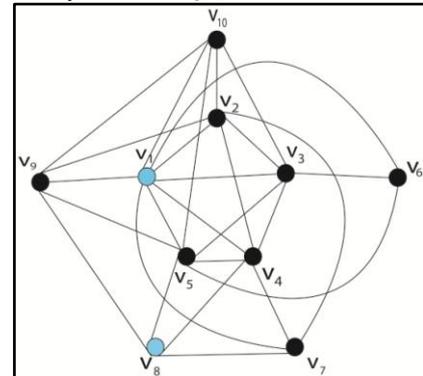
Gambar 1. Contoh graf

1. Contoh Penerapan Algoritma *LDO*

Pewarnaan graf pada gambar 1 menggunakan algoritma *LDO* adalah sebagai berikut:

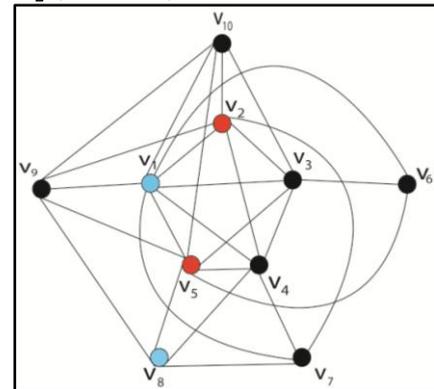
- a. Urutkan titik-titik graf dengan urutan derajat menurun.
 - $d(v_1) = 8, d(v_5) = 7,$
 - $d(v_2) = d(v_3) = d(v_4) = 6,$
 - $d(v_9) = d(v_{10}) = 5, d(v_7) = d(v_8) = 4,$
 - $d(v_6) = 3.$

- b. Pilih titik dengan jumlah derajat terbesar, yaitu v_1 , beri warna 1 (biru). Warnai juga titik yang tidak bertetangga dengan v_3 dengan warna 1 yaitu titik v_8 .(Gambar 2)



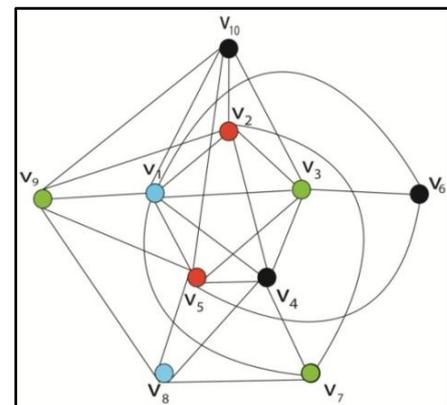
Gambar 2. Langkah b algoritma *LDO*

- c. Pilih titik yang mempunyai derajat terbesar selanjutnya yaitu v_5 dan beri warna 2 (merah), karena titik v_5 bertetangga dengan titik v_1 . Warnai juga titik yang tidak bertetangga dengan v_5 dengan warna 2 yaitu titik v_2 .(Gambar 3)



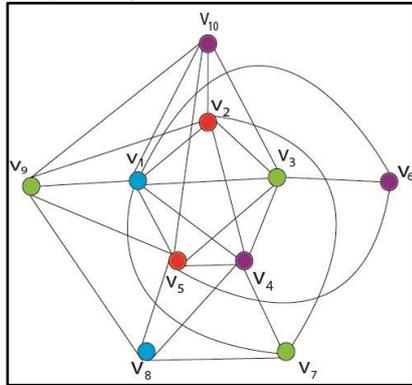
Gambar 3. Langkah c algoritma *LDO*

- d. Pilih titik berderajat terbesar selanjutnya yaitu v_3 , beri warna 3 (hijau) karena v_3 bertetangga dengan titik v_1 yang berwarna biru dan v_5 yang berwarna merah. Warnai juga titik-titik yang tidak bertetangga dengan v_5 dengan warna 3, yaitu v_9 dan v_7 .(Gambar 4)



Gambar 4. Langkah d algoritma *LDO*

- e. Pilih titik berderajat terbesar selanjutnya yaitu v_4 , beri warna 4 (ungu) karena v_4 bertetangga dengan titik v_1 yang berwarna biru, v_5 yang berwarna merah, dan v_3 yang berwarna hijau. Warnai juga titik-titik yang tidak bertetangga dengan v_4 dengan warna 4, yaitu v_{10} dan v_6 . (Gambar 5)



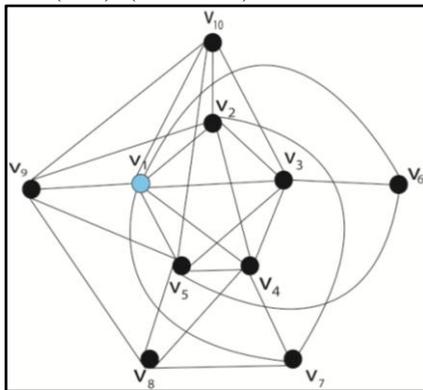
Gambar 5. Langkah e algoritma LDO

Semua titik telah diwarnai, maka proses pewarnaan selesai. Pewarnaan graf berorde 10 pada gambar 1 menghasilkan 4 warna berbeda.

2. Contoh Penerapan algoritma SDO

Berikut diberikan contoh pewarnaan graf menggunakan algoritma SDO. Misalkan terdapat graf seperti pada gambar 1. Proses pewarnaan graf tersebut menggunakan algoritma SDO adalah sebagai berikut:

- a. Pilih titik sembarang, misalkan v_1 dan beri warna 1 (biru). (Gambar 6)

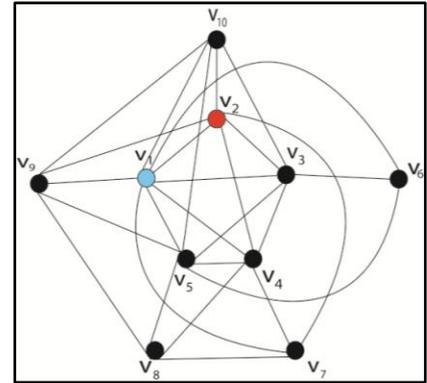


Gambar 6. Langkah a algoritma SDO

- b. Hitung derajat saturasi titik-titik yang belum diwarnai:

$$\begin{aligned} deg_s(v_2) &= deg_s(v_3) = deg_s(v_4) = \\ deg_s(v_5) &= deg_s(v_6) = deg_s(v_7) = \\ deg_s(v_9) &= deg_s(v_{10}) = 1, \quad deg_s(v_8) = 0. \end{aligned}$$

- c. Pilih titik dengan derajat saturasi terbesar. Oleh karena terdapat 8 titik yang mempunyai derajat saturasi terbesar, pilih titik dengan indeks terkecil yaitu v_2 dan beri warna 2 (merah). (Gambar 7)

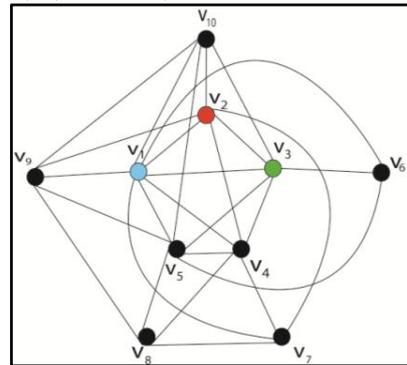


Gambar 7. Langkah c algoritma SDO

- d. Hitung derajat saturasi titik-titik yang belum diwarnai:

$$\begin{aligned} deg_s(v_3) &= deg_s(v_4) = deg_s(v_7) = \\ deg_s(v_9) &= deg_s(v_{10}) = 2, \\ deg_s(v_5) &= deg_s(v_6) = 1, \quad deg_s(v_8) = 0. \end{aligned}$$

- e. Pilih titik dengan derajat saturasi terbesar. Oleh karena terdapat 5 titik yang mempunyai derajat saturasi terbesar, pilih titik dengan indeks terkecil yaitu v_3 dan beri warna 3 (hijau). (Gambar 8)

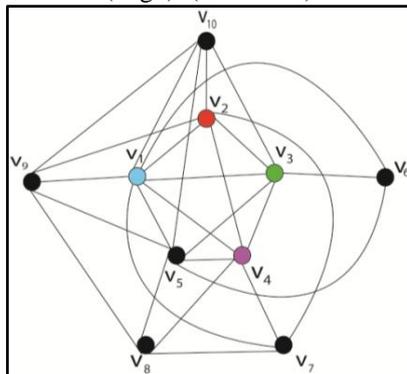


Gambar 8. Langkah e algoritma SDO

- f. Hitung derajat saturasi titik-titik yang belum diwarnai:

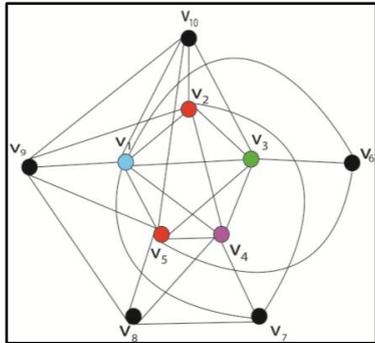
$$\begin{aligned} deg_s(v_4) &= deg_s(v_{10}) = 3, \\ deg_s(v_5) &= deg_s(v_6) = deg_s(v_7) = \\ deg_s(v_9) &= 2, \quad deg_s(v_8) = 0. \end{aligned}$$

- g. Pilih titik dengan derajat saturasi terbesar. Oleh karena terdapat 2 titik yang mempunyai derajat saturasi terbesar maka pilih v_4 dan beri warna 4 (ungu). (Gambar 9)



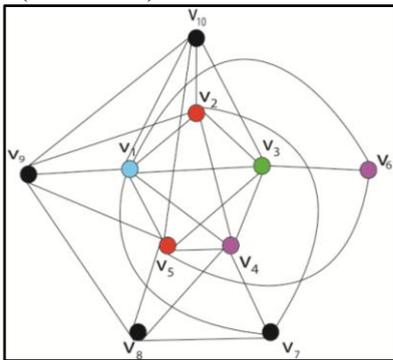
Gambar 9. Langkah g algoritma SDO

- h. Hitung derajat saturasi titik-titik yang belum diwarnai:
 $deg_s(v_5) = deg_s(v_7) = deg_s(v_{10}) = 3$,
 $deg_s(v_6) = deg_s(v_9) = 2, deg_s(v_8) = 1$.
- i. Pilih titik dengan derajat saturasi terbesar. Oleh karena terdapat 3 titik yang mempunyai derajat saturasi terbesar maka pilih v_5 dan beri warna 2 (merah). (Gambar 10)



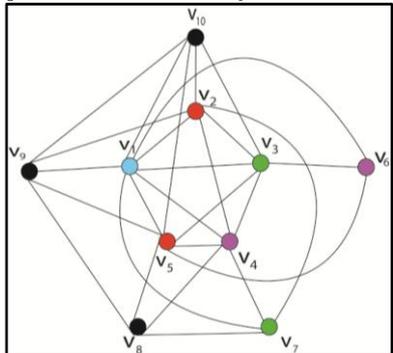
Gambar 10. Langkah i algoritma SDO

- j. Hitung derajat saturasi titik-titik yang belum diwarnai:
 $deg_s(v_6) = deg_s(v_7) = deg_s(v_{10}) = 3$,
 $deg_s(v_8) = deg_s(v_9) = 2$.
- k. Pilih titik dengan derajat saturasi terbesar. Oleh karena terdapat 4 titik yang mempunyai derajat saturasi terbesar maka pilih v_6 dan beri warna 4 (ungu). (Gambar 11)



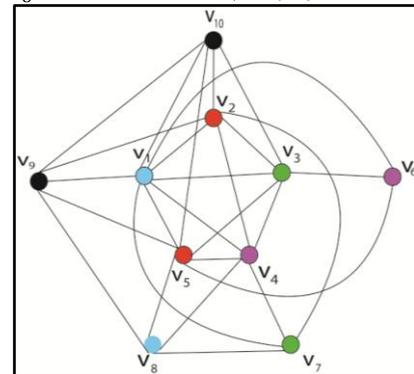
Gambar 11. Langkah k algoritma SDO

- l. Hitung derajat saturasi titik-titik yang belum diwarnai:
 $deg_s(v_7) = deg_s(v_{10}) = 3$,
 $deg_s(v_8) = deg_s(v_9) = 2$.
- m. Pilih v_6 dan beri warna 3 (hijau). (Gambar 12)



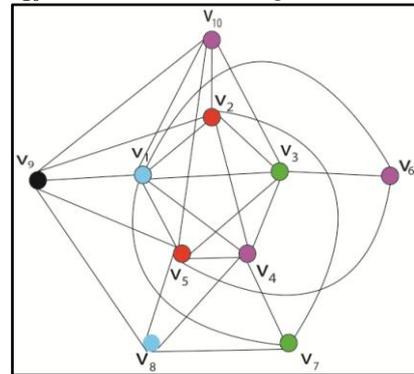
Gambar 12. Langkah m algoritma SDO

- n. Hitung derajat saturasi titik-titik yang belum diwarnai:
 $deg_s(v_8) = deg_s(v_{10}) = 3, deg_s(v_9) = 2$.
- o. Pilih v_8 dan beri warna 1 (biru). (Gambar 13)



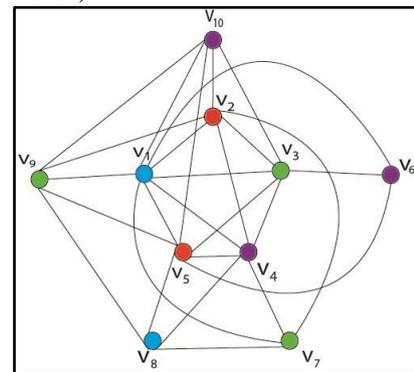
Gambar 13. Langkah o algoritma SDO

- p. Hitung derajat saturasi titik-titik yang belum diwarnai:
 $deg_s(v_{10}) = 3, deg_s(v_9) = 2$.
- q. Pilih v_{10} dan beri warna 4 (ungu). (Gambar 14)



Gambar 14. Langkah q algoritma SDO

- r. Hitung derajat saturasi titik yang belum diwarnai.
 $deg_s(v_9) = 3$.
- s. Pilih titik v_9 dan warnai dengan warna 3 (hijau). (Gambar 15)



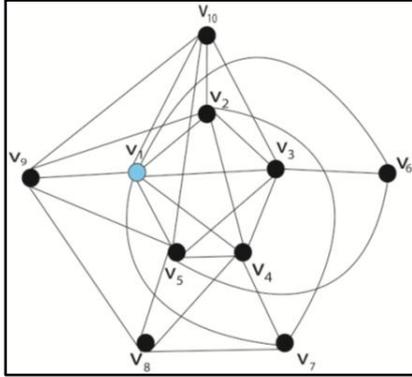
Gambar 15. Langkah s algoritma SDO

Semua titik telah diwarnai, maka proses pewarnaan selesai. Pewarnaan graf pada gambar 1 menggunakan algoritma SDO menghasilkan 4 warna berbeda.

3. Contoh Penerapan Algoritma *IDO*

Berikut diberikan contoh pewarnaan graf menggunakan algoritma *IDO*. Misalkan terdapat graf seperti pada gambar 1. Proses pewarnaan graf tersebut menggunakan algoritma *IDO* adalah sebagai berikut:

- a. Pilih titik sembarang, misalkan v_1 dan beri warna 1 (biru). (Gambar 16)

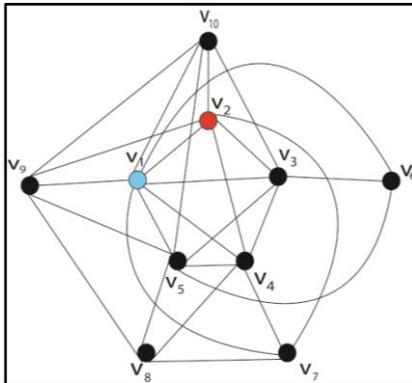


Gambar 16. Langkah a algoritma *IDO*

- b. Hitung derajat *incident* titik-titik yang belum diwarnai:

$$\begin{aligned} deg_i(v_2) &= deg_i(v_3) = deg_i(v_4) = \\ deg_i(v_5) &= deg_i(v_6) = deg_i(v_7) = \\ deg_i(v_9) &= deg_i(v_{10}) = 1, \quad deg_i(v_8) = 0. \end{aligned}$$

- c. Pilih titik dengan derajat *incident* terbesar. Oleh karena terdapat 8 titik yang mempunyai derajat *incident* terbesar, pilih titik dengan indeks terkecil yaitu v_2 dan beri warna 2 (merah). (Gambar 17)

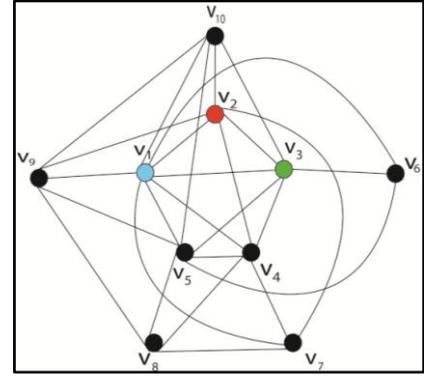


Gambar 17. Langkah c algoritma *IDO*

- d. Hitung derajat *incident* titik-titik yang belum diwarnai:

$$\begin{aligned} deg_i(v_3) &= deg_i(v_4) = deg_i(v_7) = \\ deg_i(v_9) &= deg_i(v_{10}) = 2, \\ deg_i(v_5) &= deg_i(v_6) = 1, \quad deg_i(v_8) = 0. \end{aligned}$$

- e. Pilih titik dengan derajat *incident* terbesar. Oleh karena terdapat 5 titik yang mempunyai derajat *incident* terbesar, pilih titik dengan indeks terkecil yaitu v_3 dan beri warna 3 (hijau). (Gambar 18)

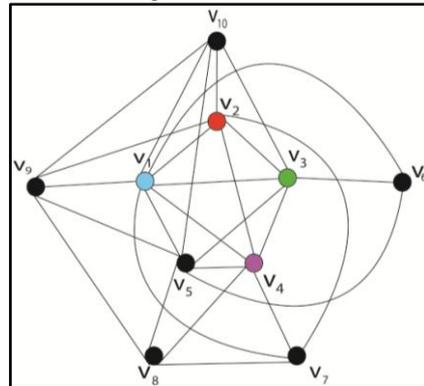


Gambar 18. Langkah e algoritma *IDO*

- f. Hitung derajat *incident* titik-titik yang belum diwarnai:

$$\begin{aligned} deg_i(v_4) &= deg_i(v_{10}) = 3, \\ deg_i(v_5) &= deg_i(v_6) = deg_i(v_7) = \\ deg_i(v_9) &= 2, \quad deg_i(v_8) = 0. \end{aligned}$$

- g. Pilih titik dengan derajat *incident* terbesar. Oleh karena terdapat 2 titik yang mempunyai derajat *incident* terbesar maka pilih v_4 dan beri warna 4 (ungu). (Gambar 19)

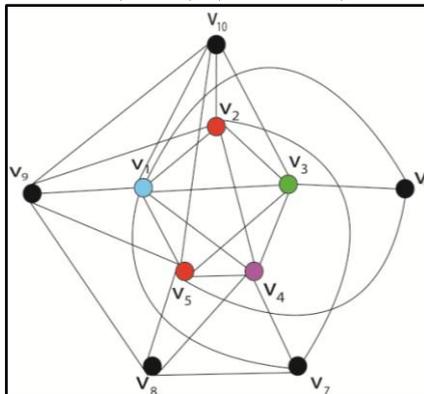


Gambar 19. Langkah g algoritma *IDO*

- h. Hitung derajat *incident* titik-titik yang belum diwarnai:

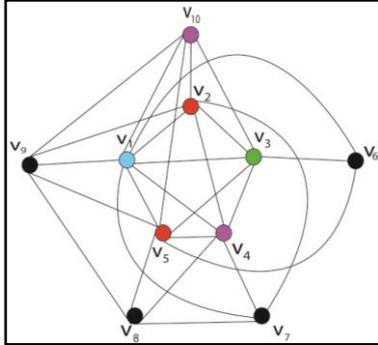
$$\begin{aligned} deg_i(v_5) &= deg_i(v_7) = deg_i(v_{10}) = 3, \\ deg_i(v_6) &= deg_i(v_9) = 2, \quad deg_i(v_8) = 1. \end{aligned}$$

- i. Pilih titik dengan derajat *incident* terbesar. Oleh karena terdapat 3 titik yang mempunyai derajat *incident* terbesar maka pilih v_5 dan beri warna 2 (merah). (Gambar 20)



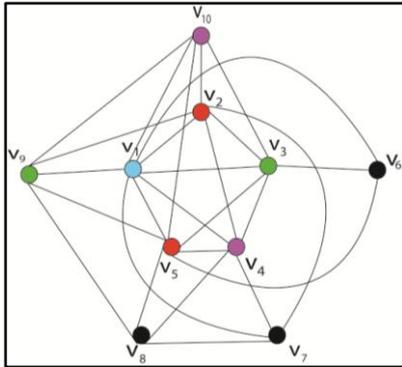
Gambar 20. Langkah i algoritma *IDO*

- j. Hitung derajat *incident* titik-titik yang belum diwarnai:
 $deg_i(v_{10}) = 4, \quad deg_i(v_6) = deg_i(v_7) = 2,$
 $deg_i(v_9) = 3, \quad deg_i(v_8) = 2.$
- k. Pilih titik dengan derajat *incident* terbesar. Pilih v_{10} dan beri warna 4 (ungu)



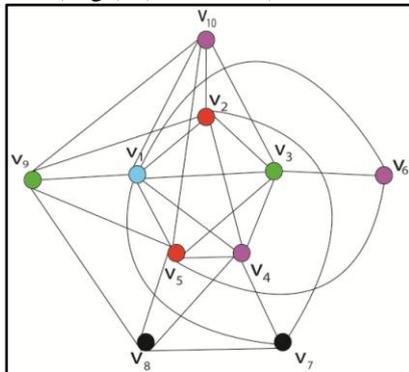
Gambar 21. Langkah k algoritma *IDO*

- l. Hitung derajat *incident* titik-titik yang belum diwarnai:
 $deg_i(v_9) = 4, \quad deg_i(v_6) = deg_i(v_7) = 3,$
 $deg_i(v_8) = 2.$
- m. Pilih titik dengan derajat *incident* terbesar, Pilih v_9 dan beri warna 3 (hijau).(Gambar 21)



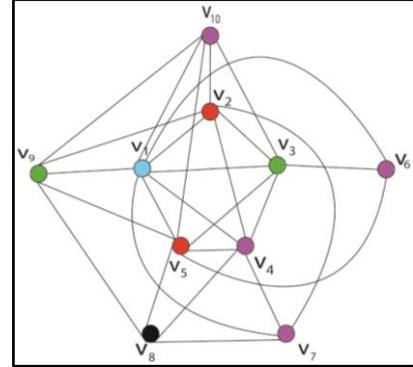
Gambar 22. Langkah m algoritma *IDO*

- n. Hitung derajat *incident* titik-titik yang belum diwarnai:
 $deg_i(v_6) = deg_i(v_7) = deg_i(v_8) = 3.$
- o. Pilih titik dengan derajat *incident* terbesar, oleh karena terdapat dua titik mempunyai derajat *incident* terbesar, maka pilih titik dengan indeks terkecil yaitu v_6 dan beri warna 4 (ungu). (Gambar 23)



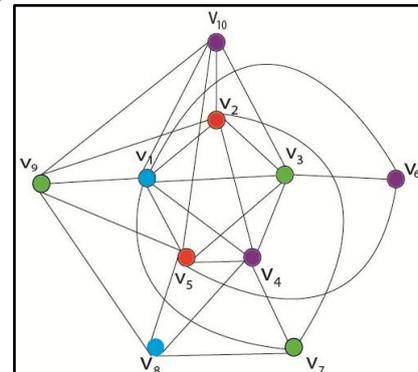
Gambar 23. Langkah o algoritma *IDO*

- p. Hitung derajat *incident* titik-titik yang belum diwarnai:
 $deg_i(v_7) = deg_i(v_8) = 3.$
- q. Pilih titik dengan derajat *incident* terbesar yaitu v_7 dan beri warna 3 (hijau). (Gambar 24)



Gambar 24. Langkah q algoritma *IDO*

- r. Hitung derajat *incident* titik yang belum diwarnai:
 $deg_i(v_8) = 4.$
- s. Pilih titik terakhir yang belum diwarnai yaitu v_8 dan beri warna 1 (biru), karena v_8 bertetangga dengan titik yang berwarna merah, hijau, ungu dan tidak bertetangga dengan titik v_1 yang berwarna biru. (Gambar 25)



Gambar 25. Langkah s algoritma *IDO*

Semua titik telah diwarnai, maka proses pewarnaan selesai. Pewarnaan graf pada gambar 1 menggunakan algoritma *IDO* menghasilkan 4 macam warna berbeda.

Dari contoh yang diberikan, dapat dilihat perbedaan langkah-langkah algoritma *LDO*, *SDO*, dan *IDO*. Algoritma *LDO* memulai pewarnaan pada titik yang mempunyai jumlah derajat terbanyak yang berpotensi bertetangga dengan titik-titik yang mempunyai banyak macam warna. Selain itu, algoritma *LDO* pada langkah awal menelusuri titik-titik yang tidak bertetangga untuk diberikan warna yang sama, setelah tidak ada lagi titik yang tidak bertetangga yang dapat diberikan warna 1 maka pewarnaan dilanjutkan pada titik yang mempunyai

derajat terbesar selanjutnya. Sedangkan algoritma *SDO* dan *IDO* sama-sama memulai pewarnaan dari sembarang titik, dan titik yang dipilih selanjutnya adalah titik yang bertetangga dengan titik yang telah diwarnai untuk diberikan warna yang berbeda. Namun pemilihan titik yang akan diwarnai selanjutnya pada algoritma *SDO* lebih rumit daripada algoritma *IDO* karena menghitung aneka warna berbeda yang ada pada tetangga perlu ketelitian yang tinggi, sedangkan pada algoritma *IDO* pemilihan titik yang akan diwarnai tidak perlu memperhatikan warna tetangga-tetangganya, hanya perlu menghitung tetangga-tetangga yang sudah diwarnai saja.

C. Hasil Pewarnaan Algoritma *LDO*, *SDO*, dan *IDO*

Setelah diterapkan pada beberapa graf sederhana berorde n , dimana $n \leq 20$, algoritma *LDO*, *SDO*, dan *IDO* tidak selalu menghasilkan jumlah warna yang sama pada setiap graf, dan diantara algoritma *LDO*, *SDO*, dan *IDO* tidak ada satupun algoritma yang konsisten dalam menghasilkan jumlah warna minimum. Hasil pewarnaan menggunakan algoritma *LDO*, *IDO*, dan *SDO* disajikan dalam tabel berikut:

TABEL I.

HASIL PEWARNAAN GRAF P_6 , C_6 , C_9 , T_9 , DAN ICOSAHEDRON DENGAN MENGGUNAKAN ALGORITMA *LDO*, *SDO* DAN *IDO*

No.	Graf	Bilangan Kromatik	Jumlah Warna yang Dihasilkan		
			<i>LDO</i>	<i>SDO</i>	<i>IDO</i>
1	P_6	2	2	2	2
2	C_6	2	2	2	2
3	C_9	3	3	3	3
4	T_9	2	2	2	2
5	Icosahedron	4	4	4	5

TABEL II.

PERBANDINGAN JUMLAH WARNA YANG DIHASILKAN ALGORITMA *LDO*, *IDO*, DAN *SDO* PADA GRAF SEDERHANA

Jumlah Titik Graf Sederhana	Kepadatan (<i>Density</i>)	Jumlah Warna yang Dihasilkan		
		<i>LDO</i>	<i>SDO</i>	<i>IDO</i>
5	20%	2	2	2
5	40%	2	2	2
5	60%	3	3	3
5	80%	3	3	3
10	20%	2	2	2
10	40%	3	3	3
10	60%	4	4	4
10	80%	6	6	6
15	20%	3	3	3
15	40%	4	4	4
15	60%	5	5	5
15	80%	7	7	8
16	20%	3	3	3

16	40%	5	4	4
16	60%	6	6	6
16	80%	8	8	8
17	20%	3	3	3
17	40%	5	4	4
17	60%	7	7	7
17	80%	8	9	9
18	20%	3	3	3
18	40%	5	4	4
18	60%	6	7	7
18	80%	9	9	9
19	20%	3	4	4
19	40%	6	5	6
19	60%	8	7	7
19	80%	9	10	10
20	20%	4	4	4
20	40%	6	6	7
20	60%	8	8	8
20	80%	9	9	10

Pewarnaan titik menggunakan algoritma *LDO*, *SDO*, dan *IDO* pada beberapa graf sembarang berorde n tidak selalu menghasilkan jumlah warna yang minimum. Dari pengujian algoritma *LDO*, *SDO* dan *IDO* pada 5 graf khusus dan 32 graf sembarang, 14 pengujian menghasilkan jumlah warna yang berbeda-beda. Pada 5 pengujian algoritma *LDO* menghasilkan jumlah warna yg lebih sedikit daripada algoritma *SDO* dan *IDO*, pada 1 pengujian algoritma *SDO* menghasilkan jumlah warna yg lebih sedikit dari algoritma *LDO* dan *IDO*, pada 4 pengujian algoritma *LDO* dan *SDO* menghasilkan jumlah warna yg lebih sedikit dari algoritma *IDO*, dan pada 4 pengujian algoritma *SDO* dan *IDO* menghasilkan jumlah warna yg lebih sedikit dari algoritma *LDO*.

SIMPULAN

Berdasarkan hasil pengujian algoritma *LDO*, *SDO*, dan *IDO* pada beberapa graf sederhana berorde n , dimana $n \leq 20$, dapat disimpulkan bahwa tidak satupun diantara algoritma *LDO*, *SDO*, dan *IDO* yang bisa ditetapkan sebagai algoritma terbaik dalam menghasilkan jumlah warna minimum. Namun berdasarkan langkah dari masing-masing algoritma, langkah algoritma *LDO* lebih sederhana dibandingkan dengan algoritma *SDO* dan *IDO*. Algoritma *SDO* dan *IDO* lebih rumit karena derajat saturasi dan derajat *incident* harus dihitung pada setiap langkah algoritma sampai semua titik selesai diwarnai, sehingga dalam menjalankan algoritma *SDO* dan *IDO* lebih memerlukan banyak waktu dibandingkan algoritma *LDO*.

REFERENSI

- [1] Al-Omari, Husein & Khair Eddin Sabri . 2004. *New Graph Coloring Algorithms*. University of Jordan, Amman, Jordan.
- [2] Astuti, Setia. 2011. *Penyusunan Jadwal Ujian Mata Kuliah Dengan Algoritma Pewarnaan Graf Welch Powell*. Jurnal, Vol 11, No.1, 68-74. Universitas Dian Nuswantoro, Semarang, Indonesia.
- [3] Budayasa, Ketut. 2007. *Teori graph dan aplikasinya*. Surabaya: Universitas Negeri Surabaya.
- [4] Harary, Frank. 1969. *Graph Theory* (Copyright). Philipina : Addison-Wesley Publishing Company.
- [5] Mansuri, Akhlak. 2010. *Coloring Programs in Graph Theory*. Int. Journal of Math. Analysis, Vol. 4, no.50, 2473 – 2479.
- [6] Munir, Rinaldi. 2005. *Buku Teks Ilmu Komputer Matematika Diskrit*. Informatika. Bandung.
- [7] webwhompers.com/graph-theory.html diakses pada tanggal 28 Agustus 2013