

## IMPLEMENTASI ALGORITMA GREEDY UNTUK MELAKUKAN GRAPH COLORING: STUDI KASUS PETA PROPINSI JAWA TIMUR

Ardiansyah<sup>1)</sup>, Fery Sofian Efendi<sup>2)</sup>, Syaifullah<sup>2)</sup>, Mateus Pinto<sup>2)</sup>, Pujiyanto<sup>2)</sup>, Hendro Steven Tempake<sup>2)</sup>

<sup>1)</sup> Program Studi Teknik Informatika, Fakultas Teknologi Industri  
Universitas Ahmad Dahlan,  
Kampus III UAD Jl. Prof. Dr. Soepomo, Janturan, Umbulharjo, Yogyakarta  
Telp 0274-379418,381523, Fax 0274-381523  
E-mail : ardi@uad.ac.id

<sup>2)</sup> Program Studi Magister Ilmu Komputer, Universitas Gadjah Mada  
Gedung SIC, Lantai III, F-MIPA UGM, Sekip Utara, Bulaksumur, Yogyakarta

### *Abstract*

*This paper will describe us how to coloring a graph by using greedy algorithm with the case study province of Jawa Timur. From this research we will know that for graph coloring at Jawa Timur Province only use four difference colors.*

**Keywords:** *edge, graph coloring, vertex.*

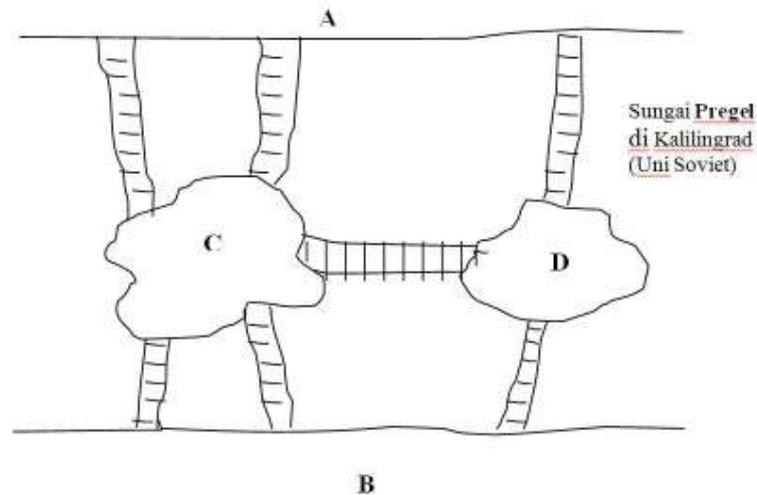
### *Abstrak*

*Paper ini akan memperlihatkan sebuah teknik penggunaan algoritma Greedy untuk melakukan pewarnaan graf (graph coloring) pada peta Propinsi Jawa Timur. Dari penelitian ini diperoleh bahwa untuk melakukan pewarnaan graph di Propinsi Jawa Timur dibutuhkan sebanyak empat buah warna yang berbeda.*

**Kata kunci:** *edge, graph coloring, vertex.*

## 1. PENDAHULUAN

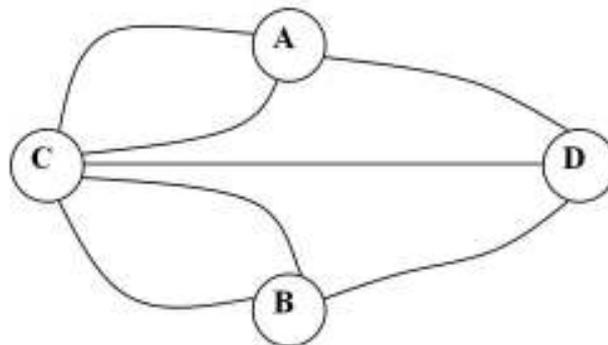
Teori Graf mulai dikenal pada saat seorang matematikawan berkebangsaan Swiss, bernama Leonhard Euler, berhasil mengungkapkan *Misteri Jembatan Konigsberg* pada tahun 1736. Di kota Konigsberg (sekarang bernama Kalilingrad, di Uni Soviet) mengalir sebuah sungai bernama sungai Pregel. Di tengah sungai tersebut terdapat dua buah pulau. Dari kedua pulau tersebut terdapat jembatan yang menghubungkan ke tepian sungai dan diantara kedua pulau. Jumlah jembatan tersebut adalah 7 buah seperti ditunjukkan pada gambar 1:



Gambar 1. Jembatan di sungai Pregel di Kalilingrad

Konon kabarnya, penduduk kota Königsberg sering berjalan-jalan ke tempat tersebut pada hari-hari libur. Kemudian muncul suatu keinginan untuk dapat menikmati daerah tersebut dengan melalui ketujuh jembatan tepat satu kali, yakni bermula dari satu tempat (A, B, C atau D) dan kembali ke tempat semula. Mereka berusaha untuk memperoleh rute yang sesuai dengan keinginan tersebut, dengan selalu mencoba menjalaninya. Setelah mencoba berkali-kali dan karena sudah cukup lama tidak diperoleh rutennya, akhirnya penduduk tersebut mengirim surat kepada Euler. Euler dapat memecahkan masalah tersebut, yakni bahwa perjalanan / rute yang diinginkan (yakni berawal dari suatu tempat, melalui ketujuh jembatan tepat satu kali, dan kembali ke tempat semula) tidak mungkin dicapai.

Secara singkat, dalam tulisannya, Euler menyajikan keadaan jembatan Königsberg tersebut seperti gambar 2:



Gambar 2. Representasi Jembatan dalam Bentuk Graph

Dalam masalah di atas, daratan (tepi A dan B, serta pulau C dan D) disajikan sebagai titik dan jembatan disajikan sebagai ruas *edge*. Euler mengemukakan teoremanya yang mengatakan bahwa perjalanan yang diinginkan di atas (yang kemudian dikenal sebagai perjalanan Euler) akan ada apabila *graf* terhubung dan banyaknya *edge* yang datang pada setiap titik (*derajat vertex*) adalah genap.

### Definisi Graph

Definisi sebuah graf itu sendiri adalah sebagai pasangan himpunan  $(V, E)$  yang dalam hal ini:

$V$  = himpunan tidak kosong dari *vertex-vertex* (*vertices* atau *node*) dan

$E$  = himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang *vertex*.

Dalam notasi matematika, graf dapat ditulis dengan:

$$G = (V, E)$$

### Pewarnaan Graph

Salah satu topik yang menarik pada graf adalah masalah pewarnaan graf (*graph colouring problem*). Bidang ini memiliki sejarah yang sangat menarik dan teori-teorinya telah menimbulkan banyak perdebatan pada kalangan matematikawan.

Pewarnaan graf adalah pemberian warna terhadap *vertex-vertex* graf di mana 2 buah vertex yang berdampingan tidak boleh mempunyai warna yang sama.

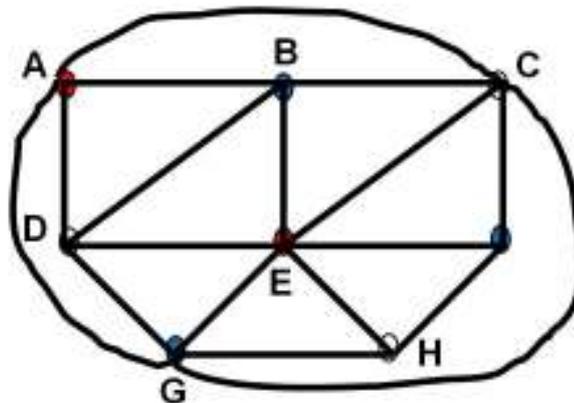
$G$  berwarna  $n$  artinya graf tersebut menggunakan  $n$  warna.

Bilangan kromatis dari  $G = K(G)$  adalah jumlah minimum warna yang dibutuhkan. Algoritma yang dapat digunakan untuk mendapatkan bilangan kromatis dari sebuah graf adalah Algoritma Welch-Powell.

Adapun langkah-langkahnya adalah :

1. Urutkan *vertex-vertex* berdasarkan derajatnya.  
Dari besar ke kecil.
2. Warnai.

Contoh :



Gambar 3. Contoh pewarnaan graph

#### Langkah 1 :

Urutan vertexnya dari besar ke kecil adalah : E, C, G, A, B, D, F, H

#### Langkah 2 :

*mewarnai :*

warna Merah : E, A

warna Putih : C, D, H

warna Biru : G, B, F

Sehingga bilangan kromatis graf di atas adalah 3.

Masalah pewarnaan graf diyakini pertama kali muncul sebagai masalah pewarnaan peta, dimana warna setiap daerah pada peta yang berbatasan dibuat berlainan sehingga mudah untuk dibedakan. Hal ini kemudian mengembangkan teorema-teorema menarik dan berujung pada teorema 4 warna, yang menyatakan : “*bilangan kromatik graf planar tidak lebih dari 4.*” Teorema ini pertama kali muncul sebagai suatu perkiraan oleh Francis Guthrie, seorang mantan murid dari Augustus De Morgan, pada tahun 1852 dan akhirnya dibuktikan oleh Kenneth Appel dan Wolfgang Haken. Pembuktian teorema ini menggunakan komputer dengan waktu yang melebihi 1000 jam.

Masalah pewarnaan graf memiliki banyak aplikasi di dalam bidang lain, misalnya membuat jadwal, pemetaan, penentuan frekuensi untuk radio, pencocokan pola, dan lain-lain. Masalah ini bahkan telah berkembang luas menjadi suatu permainan yang sangat terkenal di kalangan masyarakat luas, yaitu *sudoku*.

### 1.1. Algoritma Greedy

Algoritma *Greedy* merupakan algoritma yang menghasilkan solusi optimum melalui penyelesaian langkah per langkah (*step by step*) dengan menerapkan 2 hal berikut pada tiap langkahnya:

- a. Pilihan yang diambil merupakan pilihan terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensinya ke depan nanti, hal ini bersesuaian dengan prinsip Algoritma *Greedy* yaitu “*take what you can get now*”.
- b. Berharap dengan memilih pilihan terbaik saat itu (optimum lokal/*local optimum*) dapat mencapai solusi terbaik dari permasalahan yang dihadapi (**optimum global/global optimum**). Dalam algoritma *Greedy* diasumsikan bahwa optimum lokal merupakan bagian dari optimum global. Sedangkan untuk aplikasinya algoritma *Greedy* digunakan untuk pemecahan yang memerlukan solusi.

### Komponen Algoritma Greedy

Komponen algoritma Greedy terdiri dari :

#### a. Himpunan Kandidat C

Merupakan himpunan yang berisi elemen pembentuk Solusi.

#### b. Himpunan Solusi S

Himpunan yang berisi elemen solusi pemecahan masalah.

#### c. Fungsi Seleksi

Fungsi yang memilih kandidat yang paling memungkinkan dari Himpunan Kandidat untuk dimasukkan ke dalam himpunan solusi agar solusi optimal terbentuk. Kandidat yang sudah terpilih pada suatu langkah tidak akan dipertimbangkan lagi pada langkah selanjutnya.

#### d. Fungsi Kelayakan

Fungsi yang memeriksa apakah suatu kandidat yang terpilih akan menimbulkan solusi yang layak, yaitu kandidat tersebut, bersama dengan himpunan solusi yang terpilih tidak akan melanggar kendala yang berlaku pada masalah.

#### e. Fungsi Obyektif

Fungsi yang memaksimalkan atau meminimalkan nilai solusi.

**Algoritma Greedy : Coloring Graph**

- i. Inisialisai himpunan solusi dengan kosong
- ii. Urutkan vertex berdasarkan jumlah edge terbanyak (pengurutan dari besar ke kecil)
- iii. Melakukan pemilihan vertex yang akan diisi warnannya dengan fungsi seleksi vertex.
- iv. Memilih kandidat warna dengan menggunakan himpunan kandidat kurangi warna anggota himpunan kandidat dengan warna yang diambil.
- v. Periksa kelayakan warna yang dipilih menggunakan langkah 3. Jika layak dimasukkan ke himpunan solusi.
- vi. Periksa apakah solusi sudah meliputi perwarnaan seluruh vertex. Jika sudah maka berhenti, jika belum maka akan kembali ke langkah 3.

```

Given G=(V,E) :
Compute Degree(v) for all v in V.
Set uncolored = V sorted in decreasing
order of Degree(v).
set currentColor = 0.
while there are uncolored nodes:
  set A=first element of uncolored
  remove A from uncolored
  set Color(A) = currentColor
  set coloredWithCurrent = {A}
  for each v in uncolored:
    if v is not adjacent to anything in
    coloredWithCurrent:
      set Color(v)=currentColor.
      add v to currentColor.
      remove v from uncolored.
    end if
  end for
  currentColor = currentColor + 1.
end while
    
```

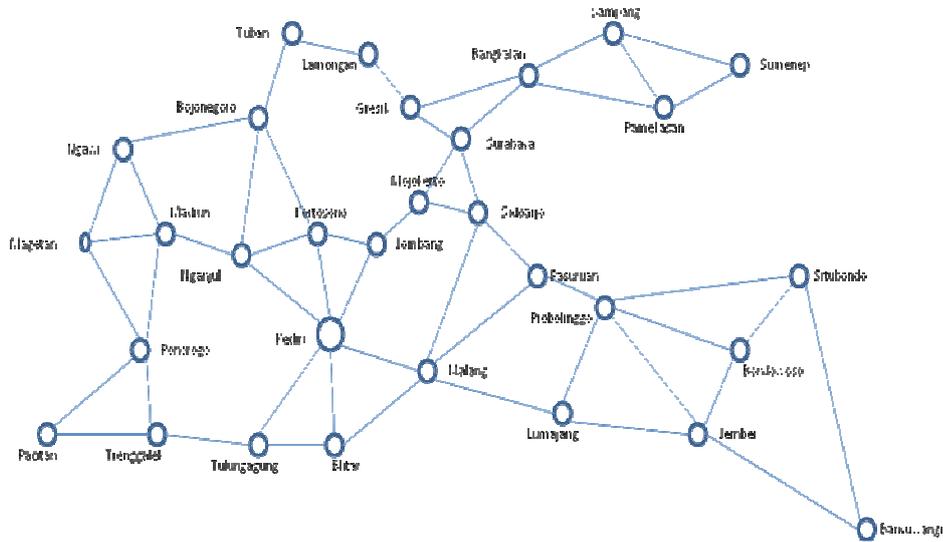
**2. MODEL, ANALISA, dan IMPLEMENTASI**

Pada permasalahan ini studi kasus yang diangkat adalah peta Jawa Timur yaitu bagaimana membentuk suatu *graf coloring* dari sebuah peta, dengan kota sebagai vertexnya dan jalan protokol sebagai *edge*-nya. Adapun dalam implementasi ke bentuk program perangkat lunak dibuat menggunakan bahasa Java.



Gambar 4. Peta propinsi Jawa Timur

1. Bagaimana memberikan warna pada kota-kota (*vertex*) di peta Jawa timur?
2. Berapa warna minimal yang dibutuhkan untuk mewarnai kota-kota (*vertex*) di Jawa timur?

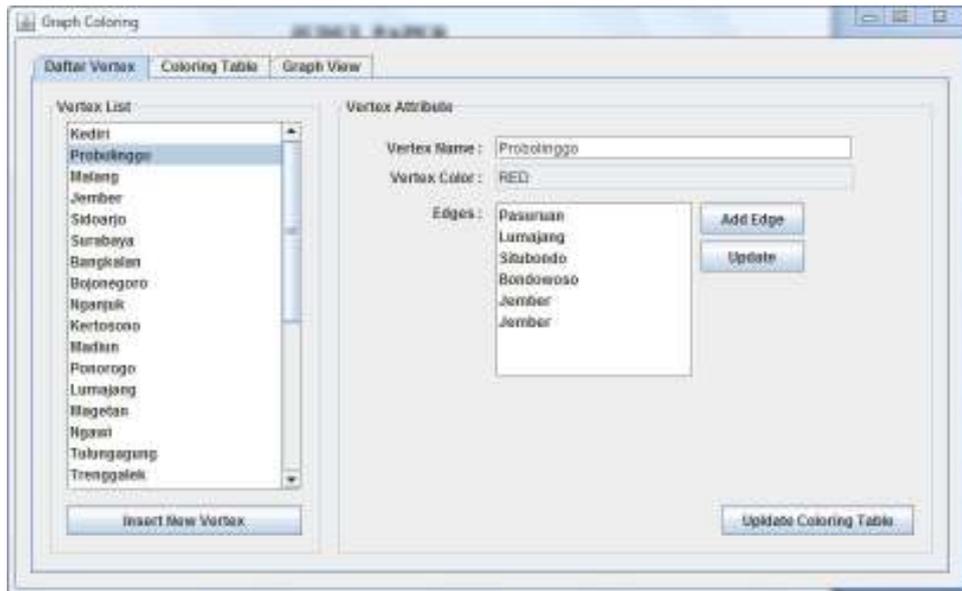


Gambar 5. Graph Map Propinsi Jawa Timur

Tabel 1. Pengurutan Vertex Berdasarkan Jumlah Edge Terbanyak (Large Degree Ordering)

No	Nama Kota	Jumlah Edge
1	Kediri	6
2	Probolinggo	5
3	Malang	5
4	Lumajang	3
5	Jember	4
6	Sidoarjo	4
7	Surabaya	4
8	Bangkalan	4
9	Bojonegoro	4
10	Nganjuk	4
11	Kertosono	4
12	Madiun	4
13	Ponorogo	4
14	Magetan	3
15	Ngawi	3
16	Tulungagung	3
17	Trenggalek	3
18	Blitar	3
19	Jombang	3
20	Mojokerto	3
21	Gresik	3
22	Pasuruan	3
23	Sampang	3
24	Pamekasan	3

25	Situbondo	3
26	Bondowoso	3
27	Banyuwangi	2
28	Sumenep	2
29	Lamongan	2
30	Tuban	2
31	Pacitan	2



Gambar 6. Penentuan kota (*vertex*) dan *edge* dalam aplikasi Java

**Himpunan Kandidat (C)**

$C = \{\text{Merah, Biru, Hijau, Ungu, Orange, Hitam, } \dots, N\}$



**Himpunan Solusi (S)**

$S = \{\text{Merah, Biru, Hijau, Ungu}\}$

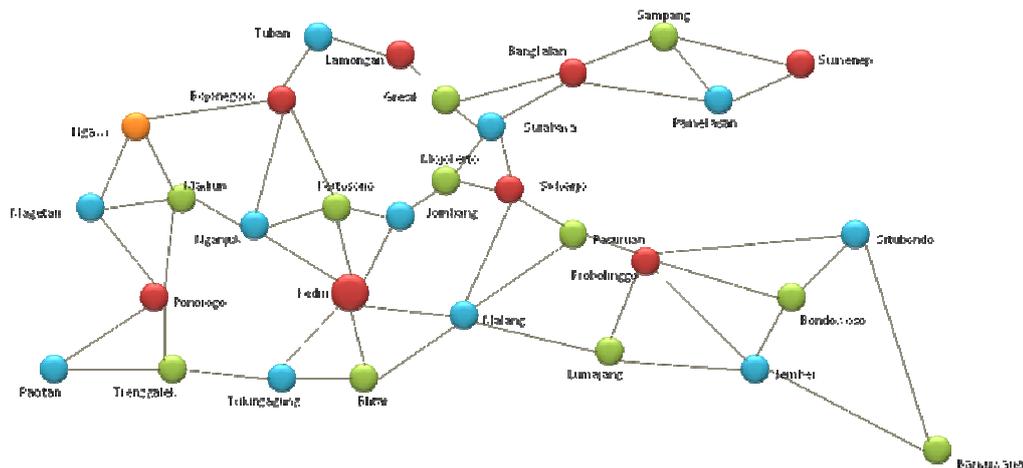


Vertex Name	Connecting To	Color
Kediri	Kertosono   Jombang   Malang   Blitar   Ng	RED
Probolinggo	Pasuruan   Lumajang   Situbondo   Bondo	RED
Malang	Kediri   Sidoarjo   Pasuruan   Lumajang   Ba	BLUE
Jember	Probolinggo   Probolinggo   Lumajang   Ba	BLUE
Sidoarjo	Malang   Surabaya   Mojokerto   Pasuruan	RED
Surabaya	Sidoarjo   Bangkalan   Mojokerto   Gresik	BLUE
Bangkalan	Surabaya   Gresik   Sampang   Pamekasa	RED
Bojonegoro	Tuban   Kertosono   Nganjuk   Ngawi	RED
Nganjuk	Kediri   Bojonegoro   Kertosono   Madiun	BLUE
Kertosono	Kediri   Bojonegoro   Nganjuk   Jombang	YELLOW
Madiun	Nganjuk   Ngawi   Magetan   Ponorego	RED
Ponorego	Madiun   Magetan   Pacitan   Trenggalek	BLUE
Lumajang	Probolinggo   Malang   Jember	YELLOW
Magetan	Madiun   Ponorego   Ngawi	YELLOW
Ngawi	Bojonegoro   Madiun   Magetan	BLUE
Tulungagung	Kediri   Blitar   Trenggalek	BLUE
Trenggalek	Ponorego   Tulungagung   Pacitan	RED
Blitar	Kediri   Malang   Tulungagung	YELLOW
Jombang	Kediri   Kertosono   Mojokerto	BLUE
Mojokerto	Sidoarjo   Surabaya   Jombang	YELLOW
Gresik	Surabaya   Bangkalan   Lamongan	YELLOW
Pasuruan	Probolinggo   Malang   Sidoarjo	YELLOW
Sampang	Bangkalan   Pamekasan   Sumenep	BLUE
Pamekasan	Bangkalan   Sampang   Sumenep	YELLOW
Situbondo	Probolinggo   Bondowoso   Banyuwangi	BLUE
Bondowoso	Probolinggo   Jember   Situbondo	YELLOW
Banyuwangi	Jember   Situbondo	RED
Sumenep	Sampang   Pamekasan	RED
Lamongan	Gresik   Tuban	RED
Tuban	Bojonegoro   Lamongan	BLUE
Pacitan	Ponorego   Trenggalek	YELLOW

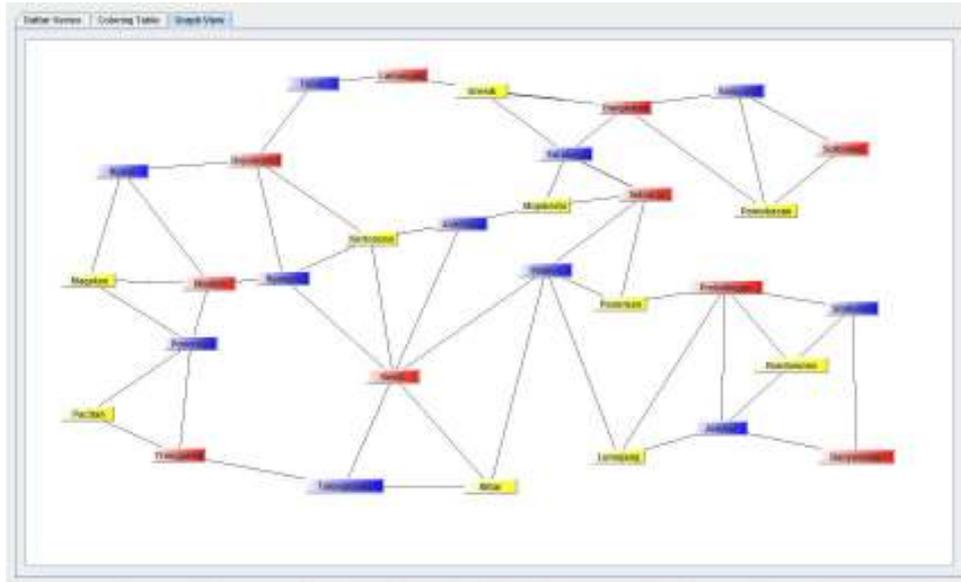
Gambar 7. Hasil skema *graph coloring*

### Hasil Solusi Pewarnaan Vertex

Gambar 8 memperlihatkan hasil solusi dari pewarnaan Vertex. Didapatkan bahwa antara vertex (kota) yang berdekatan/bersebelahan tidak memiliki warna yang sama.



Gambar 8. Hasil Pewarnaan Vertex pada Graph Map Propinsi Jawa Timur



Gambar 9. Hasil *graph coloring* setelah dilakukan *view graph*

### Fungsi Obyektif

Optimalitas warna yang berhasil digunakan dalam pewarnaan peta di atas adalah 4 warna yaitu Merah, Biru, Hijau, dan Ungu. Jumlah warna optimum inilah yang disebut sebagai bilangan Cromatic.

### 3. KESIMPULAN DAN SARAN

Dari hasil penelitian berupa pewarnaan grap (*graph coloring*) pada peta Propinsi Jawa Timur dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Jumlah *vertex* (kota) yang terdapat di peta Jawa Timur berjumlah 31.
2. Kediri adalah kota yang memiliki jumlah *edge* paling banyak yaitu 6 (enam).
3. Kota-kota yang ada di peta Jawa Timur dapat diwarnai hanya menggunakan empat warna, dengan warna antar kota yang terhubung dengan satu *edge* memiliki warna yang berbeda.

Penelitian lanjutan yang disarankan berdasarkan dari penelitian ini adalah bagaimana menemukan model *graph coloring* untuk *social networking* (jejaring sosial) atau dalam pembahasan lain disebut sebagai *social graph*.

### DAFTAR PUSTAKA

- Dharwadker, A., "The Vertex Coloring Algorithm", [http://www.dharwadker.org/vertex\\_coloring/](http://www.dharwadker.org/vertex_coloring/), diakses 12 Januari 2010.
- O'Connor, J.J and Robertson, E. F., "A History of Topology", [http://www-history.mcs.st-and.ac.uk/HistTopics/Topology\\_in\\_mathematics.html](http://www-history.mcs.st-and.ac.uk/HistTopics/Topology_in_mathematics.html), diakses 12 Januari 2010.
- Mulia, F, "Perencanaan Jadwal dengan Graph Coloring", Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- Skiena, S., "Vertex Coloring", <http://www.cs.sunysb.edu/~algorithm/files/vertex-coloring.shtml>, diakses 12 Januari 2010.
- Weisstein, Eric W. "Vertex Coloring" From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/VertexColoring.html>, diakses 12 Januari 2010.