

## Penerapan Teknik Pewarnaan *Graph* Pada Penjadwalan Ujian Dengan Algoritma *Welch-Powell*

Supiyandi<sup>1</sup>, Muhammad Eka<sup>2</sup>

<sup>1</sup>Fakultas Sains dan Teknologi, Program Studi Sistem Komputer, Universitas Pembangunan Panca Budi, Medan, Indonesia

<sup>2</sup>Universitas Nahdlatul Ulama Sumatera Utara, Medan, Indonesia

Email: <sup>1</sup>supiyandi@dosen.pancabudi.ac.id, <sup>2</sup>belimbing04@gmail.com

### Abstrak

Banyak hal dalam dunia ini yang merupakan implementasi dari *graph theory*, karena model-modelnya sangat bermanfaat untuk aplikasi yang luas, seperti: penjadwalan, optimisasi, ilmu komputer, jaringan komunikasi, analisis algoritma dan *graph coloring*. *Graph coloring* dan penyamarataannya menggunakan *tools* dalam membuat model yang beraneka ragam untuk menyelesaikan masalah penjadwalan dan masalah pemberian tugas. Salah satu aplikasi dalam *graph theory* adalah memberikan warna pada sebuah simpul, baik warna minimum maupun warna maksimum. Proses pewarnaan dilakukan dengan menghindari warna yang sama pada *vertex* yang *adjacency*, sehingga dapat diperoleh warna minimum. Dengan demikian pengguna dapat lebih mudah dalam pembuatan jadwal.

**Kata Kunci :** *Graph, jadwal, pewarnaan, algoritma Welch-powell*

### Abstract

Many things in this world are the implementation of *graph theory*, because the models are very useful for broad applications, such as scheduling, optimization, computer science, communication networks, algorithm analysis and *graph coloring*. *Graph coloring* and generalization using tools to create a variety of models to solve scheduling problems and assignment problems. One application in *graph theory* is to give color to a node, both minimum color and maximum color. The coloring process is done by avoiding the same color on the adjacency vertex, so that the minimum color can be obtained. Thus the user can more easily make the schedule.

**Keywords:** *Graph, schedule, coloring, Welch-Powell algorithm*

## 1. PENDAHULUAN

Teori *graph* merupakan topik yang banyak mendapat perhatian saat ini, karena model-model yang ada pada teori *graph* berguna untuk aplikasi yang luas. Walaupun teori *graph* berasal dari bidang ilmu Matematika, namun pada penerapannya, teori *graph* dapat dihubungkan dengan berbagai bidang ilmu dan juga kehidupan sehari-hari. Sedemikian banyaknya pengaplikasian *graph* dalam dunia ini, bila perlu dikatakan tidak ada habis-habisnya jika dibahas setiap aplikasi *graph*, karena setiap bidang ilmu dapat dikaitkan dengan *graph* seperti masalah dalam jaringan komunikasi, transportasi, ilmu komputer, riset operasi, ilmu kimia, Sosiologi, Kartografi dan lain sebagainya. Teori-teori mengenai *graph* ini telah banyak dikembangkan dengan berbagai algoritma yang memiliki kelebihan dan kelemahan masing-masing dalam menyelesaikannya. *Graph* adalah himpunan pasangan tak berurut antara *vertex* (titik atau *node*) dan *edge* (garis atau *arcs*).

Algoritma yang digunakan dalam menentukan warna pada penjadwalan ujian, yaitu algoritma *Welch-Powell*. Keuntungan dari algoritma *Welch-Powell* adalah efisien. Algoritma *Welch-Powell* adalah merupakan salah satu algoritma pewarnaan graf yang melakukan pewarnaan berdasarkan derajat tertinggi dari simpul-simpulnya atau disebut *Largest Degree Ordering* (LDO). Metode yang digunakan algoritma ini adalah dengan pewarnaan langsung pada sebuah *graph* dengan warna yang sesedikit mungkin. Namun Algoritma *Welch-Powell* ini tidak selalu memberikan jumlah warna minimum yang diperlukan untuk mewarnai.

## 2. TEORITIS

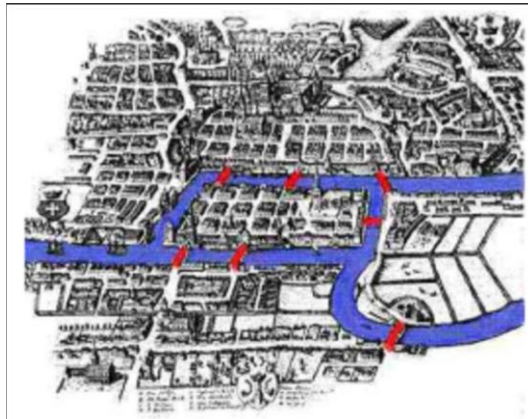
### 2.1 Teori Dasar Graph

Teori *Graph* merupakan cabang ilmu matematika diskrit yang banyak penerapannya dalam berbagai bidang ilmu seperti *engineering*, fisika, biologi, kimia, arsitektur, transportasi, teknologi komputer, ekonomi, sosial dan bidang lainnya. Teori *Graph* juga dapat diaplikasikan untuk menyelesaikan persoalan-persoalan, seperti *Travelling Salesperson Problem*, *Chinese Postman Problem*, *Shorest Path*, *Electrical Network Problems*, *Seating Problem* serta *Graph Coloring*.

Teori dasar *graph* merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan dalam kehidupan sehari-hari sampai saat ini. *Graph* adalah suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat. Dalam kehidupan sehari-hari, *graph* digunakan untuk menggambarkan berbagai macam struktur yang ada. Tujuannya adalah sebagai visualisasi objek-objek agar lebih mudah dimengerti.

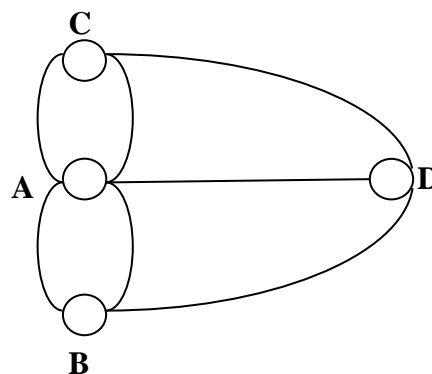
## 2.2. Latar Belakang Graph

Masalah jembatan Königsberg (Königsberg Bridge Problem) bisa menjadi contoh yang paling dekat dalam teori *graph*, dahulu merupakan masalah yang cukup rumit hingga pada akhirnya dipecahkan oleh Leonhard Euler, Matematikawan dari Swiss (1707-1783) tahun 1736. [Narsingh Deo, 1980] Königsberg adalah sebuah kota di sebelah timur Prussia (Jerman sekarang) dimana terdapat Sungai Pregel (Pregolya sekarang) dan merupakan tempat tinggal *duke of Prussia* pada abad ke-16 (tahun 1736). Kota tersebut saat ini bernama Kaliningrad dan merupakan pusat ekonomi dan industri utama di Rusia Barat. Sungai Pregel membagi kota menjadi empat daratan dengan mengalir mengitari pulau Kneiphof lalu bercabang menjadi dua buah anak sungai, seperti tampak pada gambar 1 berikut ini :



Gambar 1. Königsberg Bridge

Pada abad kedelapan belas, dibangunlah tujuh jembatan yang menghubungkan keempat daratan tersebut. Pada hari Minggu, masyarakat Königsberg biasanya berjalan-jalan dari daratan ke suatu daratan lainnya melalui jembatan tersebut. Mereka berfikir apakah mungkin untuk berjalan menyeberangi ketujuh jembatan tanpa melalui jembatan yang sama dari suatu daratan dan kembali ke tempat semula. Masalah ini pertama kali dipecahkan oleh Leonard Euler, ahli Matematika dari Swiss yang menemukan salah satu cabang dari Matematika yang saat ini dikenal sebagai “Teori *Graph*”. Solusi Euler merepresentasikan masalah ini ke dalam sebuah *graph* dengan ke empat daratan sebagai empat *vertex (node)* dan ke tujuh jembatan sebagai empat sisi (*edge*). *Graph* yang dibuat Euler diperlihatkan pada gambar 2 di berikut ini :

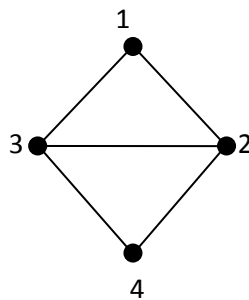


Gambar 2. *Graph* yang merepresentasikan jembatan Königsberg

## 2.3. Jenis-jenis Graph

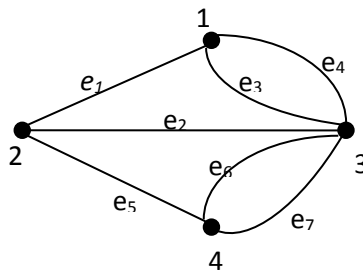
Jenis-jenis *graph* dapat diklasifikasikan berdasarkan beberapa faktor-faktor berikut :

- a. Berdasarkan ada tidaknya gelang atau *edge* ganda pada suatu *graph*, maka *graph* digolongkan menjadi dua jenis, yaitu :
  1. *Graph* sederhana (*Simple graph*)  
*Graph* sederhana yaitu *graph* yang tidak mengandung *edge* maupun *edge* ganda. Gambar 3 di bawah ini adalah contoh *graph* sederhana.



Gambar 3. Graph Sederhana

2. *Graph* tak-sederhana (*Unsimple-graph*)  
*Graph* tak-sederhana yaitu *graph* yang mengandung *edge* ganda atau *edge*. Gambar 2.10 di bawah ini adalah contoh *graph* tidak sederhana.



Gambar 4. Grap tak-sederhana

### 2.3. Algoritma Welch-Powell

Algoritma Welch-Powell dapat digunakan untuk mewarnai sebuah *graph*  $G$  secara efisien. Algoritma ini tidak selalu memberikan jumlah warna minimum yang diperlukan untuk mewarnai  $G$ , namun algoritma ini cukup praktis untuk digunakan dalam pewarnaan simpul sebuah *graph*. Algoritma Welch-Powell hanya cocok digunakan untuk *graph* dengan orde yang kecil. Oleh karena itu algoritma Welch-Powell hanya dapat menentukan batas atas warna. Langkah-langkahnya adalah sebagai berikut :

- a. *Langkah 1* (melabel titik dengan derajatnya). Label titik  $V_1, V_2, \dots, V_n$  sedemikian hingga derajat  $(V_1) >$  derajat  $(V_2) > \dots >$  derajat  $(V_n)$ .
- b. *Langkah 2* (warnai titik belum berwarna pertama dari titik-titik belum berwarna yang berdekatan dengan titik itu). Berikan warna yang belum digunakan pada titik belum berwarna yang pertama pada daftar titik itu. Lakukan hal itu pada semua titik dalam daftar secara terurut, berikan warna baru ini pada setiap titik yang tidak berdekatan dengan setiap titik lain yang telah diwarnai ini.
- c. *Langkah 3* (*graph*nya telah diwarnai?). Jika beberapa titiknya belum berwarna, maka kembalilah ke langkah 2.
- d. *Langkah 4* (selesai). Pewarnaan *graph* telah dilakukan.

## 3. ANALISA DAN PEMBAHASAN

Analisis berguna untuk mengetahui kebutuhan perangkat lunak dan bagaimana aplikasi yang akan dibangun. Dalam penelitian diperlukan pengumpulan-pengumpulan sumber pengetahuan dan data yang dapat menganalisis hasil perangkat yang baik. Salah satu aplikasi penerapan pewarnaan *graph* dalam kehidupan sehari-hari adalah dalam penyusunan sebuah jadwal. Sebuah jadwal yang ada mula-mula dipetakan menjadi bentuk *graph* terlebih dahulu. Proses pewarnaan *graph* ini nantinya akan dilakukan pada *graph* yang terbentuk. Pemetaan dilakukan dengan mengasumsikan bahwa setiap jadwal adalah sebuah *vertices* (simpul) dan urutan jadwal atau dua jadwal yang tidak bisa diadakan bersamaan dipetakan dengan membuat *edge* (sisi) antara dua titik tersebut.

Terdapat delapan orang mahasiswa yang akan mengambil jadwal ujian,  $M = \{ \text{Andi, Okta, Indah, Doni, Dani, Wina, Novya, Hadi} \}$ ,  $M =$  Mahasiswa dan 5 mata kuliah yang dapat dipilih oleh kedelapan mahasiswa tersebut,  $MK = \{ \text{Pemrograman Visual, Jaringan Komputer, Matematika Diskrit, Kalkulus, Bahasa Inggris} \}$ ,  $MK =$  Mata Kuliah. Tabel 1 akan memperlihatkan matriks lima mata kuliah dan delapan orang mahasiswa.

**Tabel 1.** Tabel Mata kuliah yang akan diambil 8 Mahasiswa

Mata Kuliah Mahasiswa	Pemrograman Visual	Jaringan Komputer	Matematika Diskrit	Kalkulus	Bahasa Inggris
Andi	0	1	0	0	1
Okta	0	1	0	1	0
Indah	0	0	1	1	0
Doni	1	1	0	0	0
Dani	0	1	0	1	0
Wina	0	0	1	1	0
Novya	1	0	1	0	0
Hadi	0	0	1	1	0

Keterangan: Mata kuliah yang akan diambil oleh delapan Mahasiswa

0 = Mahasiswa tidak mengontrak mata kuliah

1 = Mahasiswa mengontrak mata kuliah

Ani = { Jaringan Komputer, Bahasa Inggris }

Budi = { Jaringan Komputer, Kalkulus }

Jesika = { Matematika Diskrit, Kalkulus }

Steven = { Pemrograman Visual, Jaringan Komputer }

Indra = { Jaringan Komputer, Kalkulus }

Maria = { Matematika Diskrit, Kalkulus }

Demet = { Pemrograman Visual, Matematika Diskrit }

Poltak = { Matematika Diskrit, Kalkulus }

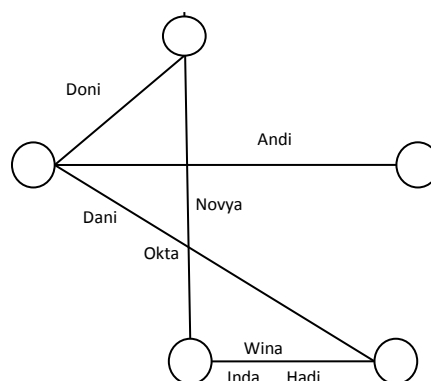
Pada tabel 1 terlihat matriks lima mata kuliah dan delapan orang mahasiswa. Angka 1 pada elemen (i,j) menandakan bahwa mahasiswa I memilih mata kuliah j, sedangkan angka 0 menyatakan bahwa mahasiswa tersebut tidak memilih mata kuliah j. Berdasarkan tabel tersebut, akan ditentukan sebuah jadwal ujian sedemikian sehingga semua mahasiswa dapat mengikuti semua ujian mata kuliah tersebut. Oleh karena itu tidak boleh terdapat jadwal ujian mata kuliah yang bertabrakan dengan jadwal ujian mata kuliah lainnya yang juga diambil oleh mahasiswa tersebut. Ujian dua buah mata kuliah dapat dijadwalkan pada waktu yang sama jika tidak ada mahasiswa yang sama yang mengikuti ujian dua mata kuliah tersebut. Penyelesaian untuk masalah ini sama dengan persoalan menentukan bilangan kromatik untuk sebuah graph. Pertama-tama, persoalan tersebut dipetakan ke dalam sebuah graph, dimana setiap simpul dalam graph tersebut menyatakan mata kuliah. Dan sisi yang menghubungkan dua simpul menyatakan ada mahasiswa yang memilih kedua mata kuliah tersebut.

### 3.1 Representasi Pedjwalan ke dalam suatu Graph

*Representasi Graph* dari perancangan program teknik pewarnaan graph pada tabel 1. adalah sebagai berikut :

1. Program akan dirancang dengan terlebih dahulu membentuk titik-titik *verteks* yang berjumlah 5 *verteks* sebagai penempatan jadwal kuliah.
2. Setiap *verteks* akan diberi label dengan nama-nama jadwal kuliah yang akan diambil mahasiswa,  $V(G) = \{PV, JK, MD, KAL, BI\}$
3. *Verteks-verteks* yang telah dibentuk akan dihubungkan dengan sisi (*edge*),
4. Setiap *edge* yang menghubungkan antar *verteks* akan diberi label nama mahasiswa untuk membentuk graph menjadi sempurna.  $E(G) = \{Andi, Okta, Indah, Doni, Dani, Wina, Novya, Hadi\}$ .

Gambar hasil rancangan awal program adalah sebagai berikut :



**Gambar 5.** Graph Mata Kuliah Delapan Orang Mahasiswa

Keterangan : Graph Mata Kuliah Delapan Orang Mahasiswa

PV = Pembrograman Visual

JK = Jaringan Komputer

MD = Matematika Diskrit

KAL = Kalkulus

BI = Bahasa Inggris

Berdasarkan gambar di atas :

$G=(V,E)$

$V= \{ PV, JK, MD, KAL, BI \}$

$E= \{(PV,JK), ( JK,KAL), (PV,MD), ( MD,KAL), (JK,BI)\}$

Dapat dilihat pada graph Gambar 3.1 bahwa apabila terdapat dua buah simpul yang dihubungkan oleh kedua sisi, maka ujian kedua mata kuliah tersebut tidak dapat diadakan secara bersamaan. Simpul (mata kuliah) tidak boleh mendapat alokasi waktu (warna simpul) yang sama. Warna-warna yang berbeda dapat diberikan kepada simpul-simpul graph tersebut. Jadwal yang efisien adalah jadwal yang memungkinkan waktu sedikit mungkin untuk melaksanakan semua kegiatan tersebut. Oleh karena itu, disini yang akan dicari adalah bilangan kromatik graph tersebut,  $\chi(G)$ .

## 4. IMPLEMENTASI

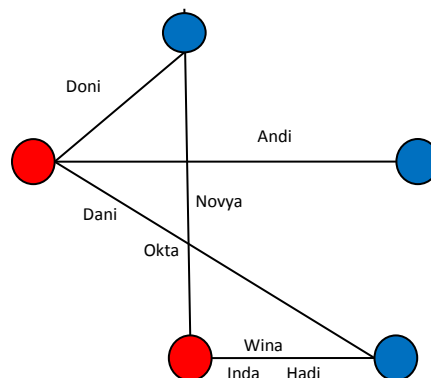
### 4.1 Penerapan

Cara yang digunakan dalam pewarnaan *vertex* dengan menggunakan algoritma pewarnaan *vertex*, adalah :

1. Nyatakan mata kuliah sebagai *vertex*, dan mahasiswa yang mengontrak mata kuliah sebagai *edge*.
2. Setiap *vertex* bertetangga harus mempunyai warna berbeda (warna setiap *vertex* harus berbeda).

Algoritma Welch-Powell merupakan salah satu algoritma pewarnaan graph yang melakukan pewarnaan berdasarkan derajat tertinggi dari simpul-simpulnya. Dari graph gambar 5. Maka teknik pewarnaannya dengan algoritma WELCH-POWELL

- a. Urutkan simpul berdasarkan derajatnya dari besar ke kecil : Simpul berderajat terbesar adalah JK, yaitu 3 (mempunyai 3 sisi) kemudian simpul MD,PV,KAL berderajat 2 dan BI berderajat 1. Jadi Urutannya adalah : JK,MD,PV,KAL,BI
- b. Ambil warna pertama, misalnya Merah. Beri warna Merah simpul JK (karena JK adalah simpul urutan pertama).
- c. Kemudian cari simpul yang tidak berdampingan dengan simpul JK, beri warna yang sama (merah). Simpul MD adalah simpul yang tidak berdampingan dengan JK sehingga diperoleh urutan simpul yang belum diberi warna adalah PV,KAL,BI
- d. Ambil warna kedua, misalnya Biru, warnai simpul PV( karena simpul PV sekarang ada di urutan pertama).
- e. Kemudian cari simpul yang tidak berdampingan dengan simpul PV beri warna yang sama (Biru).
- f. Diberikan warna yang sama pada simpul KAL dan BI dengan warna simpul PV yaitu biru karena Simpul KAL dan BI tidak berdampingan dengan simpul MD.
- g. Dan pada gambar 3.3 merupakan hasil pewarnaan graph tersebut adalah:



**Gambar 6.** Graph yang Telah Diberi Warna Tiap Simpulnya

**Tabel 2.**

Simpul	PV	JK	MD	KAL	BI
Derajat	2	3	2	2	1
Warna	Biru	Merah	Merah	Biru	Biru

Hasil

Pewarnaan dengan menggunakan Algoritma Welch Powell

Pada tabel 2 terlihat bahwa ujian untuk mata kuliah PV, KAL, dan BI dapat dilaksanakan pada waktu yang bersamaan, begitu pula dengan mata kuliah JK dan MD. Warna yang paling sedikit digunakan adalah 2, jadi bilangan kromatis nya  $\chi(G) = 3$ . Perbedaan warna simpul menunjukkan bahwa ujian mata kuliah tersebut dilaksanakan pada waktu yang berbeda.

## 5. KESIMPULAN

Setelah dilakukan pembahasan pada bab-bab sebelumnya, maka dapat disimpulkan sebagai berikut :

1. *Coloring graph* dapat diimplementasikan untuk penjadwalan ujian.
2. Langkah awal penyelesaian adalah dengan memetakan suatu jadwal ke dalam graf lalu menentukan bilangan kromatik graf tersebut.
3. Dengan penerapan *coloring graph* dapat membantu dalam penyusunan jadwal ujian sehingga tidak terjadi bentrokan jadwal yang dikontrak mahasiswa
4. Algoritma ini cukup praktis untuk digunakan dalam pewarnaan simpul sebuah graf

## REFERENCES

1. Munir, Rinaldi. 2005. Buku Teks Ilmu Komputer “Matematika Diskrit”, Informatika, Bandung, Edisi Ketiga
2. Lipschutz, Seymour & Lipson, L .L. 2002. Seri Penyelesaian Soal Schaum: Matematika Diskrit 2 (jilid 2). Jakarta: Salemba Teknika
3. Johnsonbaugh, Richard. 2002. Matematika Diskrit: Edisi Bahasa Indonesia. Jakarta : PT Prenhallindo
4. Jek, Jong. 2002. Matematika Diskrit dan Aplikasinya Pada Ilmu Komputer. Yogyakarta: Andi
5. Rinaldi, M. 2006. *Diktat Kuliah IF 2153 Matematika Diskrit*”, Program Studi Teknik Informatika, Bandung, Indonesia.