PENGEMBANGAN METODE *GRAPH COLORING* UNTUK *UNIVERSITY COURSE TIMETABLING PROBLEM* PADA FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS TARUMANAGARA

Lely Hiryanto dan Jacklin Sinthia Thio

Laboratorium Penelitian *Distributed System,* Fakultas Teknologi Informasi, Universitas Tarumanagara, Jl. Letjen.S.Parman No. 1 Blok R Lantai XI Grogol, 11440, Jakarta Barat, Indonesia

E-mail: lely@fti.untar.ac.id

Abstrak

University Course Timetabling Problem merupakan proses penjadwalan mata kuliah di sebuah universitas yang hasilnya diusahakan seoptimal mungkin untuk tidak saling berbenturan dengan batasan-batasan dan syarat-syarat (*constraints*) tertentu. Dalam menentukan penjadwalan berbasis perhitungan, salah satu metode yang dapat digunakan adalah *Graph Coloring*. *Graph Coloring* merupakan merupakan metode yang paling sederhana dan dapat digunakan untuk menentukan penjadwalan yang memiliki berbagai macam *constraints*. Pada penelitian ini, peneliti mengusulkan pengembangan dari metode *Graph Coloring* yang ada untuk membuat penjadwalan mata kuliah yang optimal dengan memertimbangkan berbagai macam *constraints*. Pengembangan ini diujicobakan ke penjadwalan mata kuliah di Fakultas Teknologi Informasi Universitas Tarumanagara (FTI Untar). Hasil percobaan menunjukkan bahwa pengembangan metode *Graph Coloring* memberikan hasil penjadwalan yang memenuhi rata-rata 93% seluruh *constraints* yang ditentukan. Rata-rata 7% pelanggaran *constraints* dikarenakan keterbatasan jumlah ruang dan total slot waktu kuliah, serta permintaan jadwal tertentu oleh dosen.

Kata Kunci: *university course timetabling problem, vertex-based graph coloring, welsh-powell algorithm*

Abstract

University Course timetabling problem is the process of scheduling courses at a university whose results are optimally arranged to not collide with the limits and conditions (constraints) specified. In determining the scheduling komputatif, one method that can be used is the Graph Coloring. Graph Coloring is the simplest method and can be used to determine which have a variety of scheduling constraints. In the present study, the researcher proposes the development of the existing methods of Graph Coloring to make optimal scheduling of courses taking into account various constraints. This development was tested to the scheduling of courses in the Faculty of Information Technology University Tarumanagara (FTI Untar). The experimental results show that the development of methods of Graph Coloring deliver results that meet the scheduling of an average 93% of all the specified constraints. Average of 7% violation constraints due to limitations of space and the total number of time slots in college, and request a specific schedule by the lecturer.

Keywords: *university course timetabling problem, vertex-based graph coloring, welsh-powell algorithm*

1. Pendahuluan

Untuk menjalankan aktivitas sehari-hari, umumnya kita membutuhkan penjadwalan kegiatan agar setiap kegiatan dapat dijalankan pada waktunya dan tidak bertabrakan satu sama lain. Namun adakalanya, penyusunan jadwal untuk kegiatan-kegiatan tersebut mengalami kendala karena jumlah kegiatan yang sangat banyak, sehingga seringkali tidak ditemukan penjadwalan yang tepat untuk setiap kegiatan agar dapat terlaksana dengan baik.

Hal yang serupa juga seringkali terjadi di dalam melakukan penjadwalan untuk mata kuliah di universitas. Di setiap universitas perlu dilakukan suatu penjadwalan mata kuliah yang dilakukan pada saat setiap semester hampir berakhir. Permasalahan penjadwalan ini pada umumnya disebut sebagai *University Course Timetabling Problem*.

Penjadwalan mata kuliah pada suatu universitas dengan universitas lain berbeda satu sama lain karena masing-masing universitas memiliki *constraints* yang berbeda yang harus dipenuhi oleh masing-masing universitas. Begitu

pula, dalam satu universitas, setiap fakultas bahkan program studi dapat memiliki aturan penjadwalan yang berbeda. Jumlah mata kuliah yang diselenggarakan oleh universitas tidaklah sedikit dan tentunya akan memakan waktu yang cukup lama jika penjadwalan mata kuliah tersebut dilakukan secara manual.

Dalam menentukan penjadwalan secara komputatif, sejumlah peneliti berfokus memberikan solusi untuk *University Course Timetabling Problem* dengan menerapkan *Genetic Algorithm* (GA) melalui sejumlah pengulangan untuk menghasilkan populasi yang baru (pemilihan kromoson terbaik) menuju solusi berbagai macam penjadwalan yang cukup optimal [1-5]. Permasalahan utama dari GA adalah waktu pemrosesan yang cukup lama untuk optimalisasi penjadwalan. Hasil penjadwalan didasarkan pada pemilihan nilai *fitness* terbaik yang merupakan formulasi dari setiap pelanggaran *constraints*.

Neural Network (NN) adalah metode lain yang banyak digunakan untuk proses penjadwalan. Untuk *University Course Timetabling Problem*, penerapan NN terbukti dapat memberikan hasil penjadwalan yang cukup optimal [6].

Metode lain yang lebih sederhana, tetapi dapat memberikan hasil yang optimal adalah metode *Graph Coloring*. Metode ini merupakan metode yang paling sederhana dan dapat digunakan untuk menentukan penjadwalan mata kuliah yang memiliki berbagai macam *constraints.*

Metode *Graph Coloring* merupakan metode yang digunakan untuk menentukan jumlah warna minimum yang dapat diberikan pada *nodes* atau *edges* di dalam sebuah *graph* sehingga *node*satau *edges* yang saling berhubungan tidak memiliki warna yang sama. Berdasarkan cara kerjanya tersebut, metode ini dapat dipastikan menghasilkan penjadwalan kegiatan dapat meminimalkan atau sampai menghilangkan pelanggaran terhadap *constraints*.

Malkawi dkk [7] menerapkan metode ini dan berdasarkan hasil percobaan mereka, penggunaan metode *Graph Coloring* ini mampu membantu di dalam menentukan jadwal ujian yang optimal dan akurat dengan *constraints* yang sedemikian rupa. Terkait dengan penjadwalan mata kuliah, Koyuncu dan Seçir [8] menerapkan metode *Graph Coloring* dan mendapatkan suatu penjadwalan mata kuliah yang tidak bertabrakan satu sama lain, dengan mengasumsikan *vertex* sebagai mata kuliah dan jika antar dua *vertex* tidak boleh bertabrakan, maka dibuatkan *edge*. *vertex* yang langsung terhubung akan diwarnai dengan warna yang berbeda. Dandashi dan Mouhamed [9] melakukan modifikasi terhadap *vertex*-based *Graph Coloring* untuk mendapatkan distribusi mata kuliah yang merata untuk setiap warna dan menyeimbangkan jumlah mata kuliah pada setiap slot waktu yang tersebar di ruang-ruang kelas yang ditentukan. Hasil percobaan mereka menunjukkan bahwa tetap terdapat tabrakan dalam penjadwalan, tetapi terjadi peningkatan dalam tingkat pemanfaatan sumber daya yaitu ruang kuliah.

Penelitian ini berfokus untuk mengembangkan metode *Graph Coloring* untuk *University Course Timetabling Problem*. Contoh kasus spesifik digunakan permasalahan penjadwalan mata kuliah di FTI Untar. Tujuan dari pengembangan metode tersebut adalah untuk menentukan algoritma baru yang didasarkan pada metode *Graph Coloring* yang dapat menghasilkan penjadwalan mata kuliah yang optimal agar jadwal kelas mata kuliah yang tidak boleh bertabrakan dapat diminimalkan atau sampai tidak lagi bertabrakan.

Paper ini disusun dalam 4 bagian. Bagian 2 membahas tentang metode *Graph Coloring*. beserta pengembangan metode *Graph Coloring*, berbentuk model solusi penjadwalan mata kuliah. Bagian 3 melaporkan hasil percobaan menggunakan kasus penjadwalan mata kuliah di FTI Untar, yang kemudian hasilnya disimpulkan pada bagian 4. Pada bagian 4 ini pun diusulkan rencana penelitian selanjutnya terkait dengan topik penelitian ini.

2. Metodologi

Graph didefinisikan sebagai kumpulan dari titik-titik simpul yang disebut *vertex* atau *node* serta kumpulan dari garis-garis yang disebut *edge* [10]. *vertex*s yang berada di dalam *graph* ini akan dihubungkan satu sama lain oleh *edge*s yang berada di dalam *graph* tersebut.

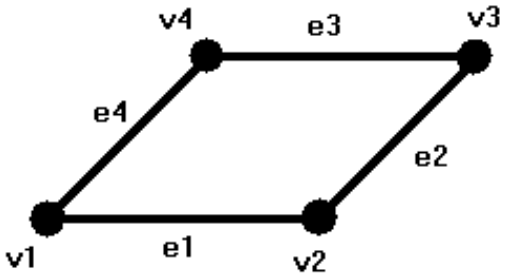
Kegunaan *graph* sangat banyak. Umumnya *graph* digunakan untuk memodelkan suatu masalah sehingga menjadi lebih mudah untuk diselesaikan, yaitu dengan cara merepresentasikan objek-objek tersebut. Contoh pemodelan suatu masalah dengan menggunakan *graph* dapat dilihat pada penggambaran rangkaian listrik, jaringan *network* komputer, peta, dan lain-lain.

Dalam matematika diskrit, *graph* dikenal sebagai pasangan himpunan (V,E), di mana:

V : himpunan dari titik-titik simpul dan bukan merupakan himpunan yang kosong = {v1, v2, v3, …, vn)
E : himpunan sisi dari *graph* tersebut yang menghubungkan sepasang simpul pada *graph* = {e1, e2, …, en}.

Jika sebuah *edge* e1 menghubungkan sepasang *vertex* vi dan vj, maka secara sederhana hubungan tersebut digambarkan pada gambar 1.

$$e1 = (vi,vj) \quad (1)$$



Gambar 1. Bentuk sebuah *graph* sederhana [10].

Pada gambar di atas terdapat sebuah *graph* sederhana G yang berisikan sekumpulan *vertex* dan *edge*. Secara matematis, kondisi tersebut dapat dituliskan sebagai berikut [10]:

$$G = (V, E) \quad (2)$$

Keterangan :
V = (v1, v2, v3, v4)
E = (e1, e2, e3,e4)

Pada *graph* di atas, v1 dan v4 disebut sebagai *vertex-vertex* yang saling bertetangga. Dua buah *vertex* dikatakan bertetangga jika kedua *vertex* dihubungkan langsung oleh sebuah *edge*. Karenanya selain v1 dan v4, v1 dan v2 serta v3 dan v4, juga disebut sebagai *vertex* yang saling bertetangga.

Graph Coloring merupakan salah satu metode yang paling sering dibahas di dalam pengaplikasian teori *Graph* ke dalam berbagai permasalahan. Pewarnaan pada *graph* dapat diartikan sebagai pemberian tanda atau label pada tiap-tiap *vertex*, *edge*, atau wilayah sehingga setiap *vertex*, *edge*, atau wilayah yang saling berhubungan satu sama lain tidak memiliki label yang sama. Label yang dimaksudkan di sini adalah warna yang diberikan pada masing-masing *edge*, *vertex*, atau wilayah.

Pewarnaan pada *graph* dapat dibedakan menjadi tiga yaitu pewarnaan *vertex*, pewarnaan *edge*, dan pewarnaan wilayah. Pewarnaan *edge* merupakan pemberian warna pada tiap-tiap *edge* di dalam *graph* sehingga setiap *edge* yang berhubungan dengan *vertex* yang sama tidak memiliki warna yang sama [11]. Pewarnaan wilayah merupakan pemberian wilayah pada tiap-tiap bagian di dalam *graph* sehingga tidak ada wilayah yang bersebelahan yang memiliki warna yang sama. Pewarnaan wilayah banyak diterapkan di dalam masalah pewarnaan peta.

Pewarnaan *vertex* merupakan pemberian warna pada tiap-tiap *vertex* di dalam *graph* sehingga setiap *vertex* yang bertetangga satu sama lain tidak memiliki warna yang sama. Pewarnaan *vertex* ini banyak digunakan di dalam menyelesaikan permasalahan penjadwalan. Pada subbab berikutnya akan dijelaskan mengenai pewarnaan *vertex* lebih lanjut.

Pewarnaan *vertex* merupakan pemberian warna pada tiap-tiap *vertex* di dalam *graph* sehingga setiap *vertex* yang saling bertetangga tidak memiliki warna yang sama. Di dalam pewarnaan *vertex*, jumlah warna yang boleh dipergunakan haruslah seminimal mungkin. Jumlah warna paling minimum yang dapat diterapkan pada *Graph* ini sering disebut dengan angka kromatik ($\chi(G)$).

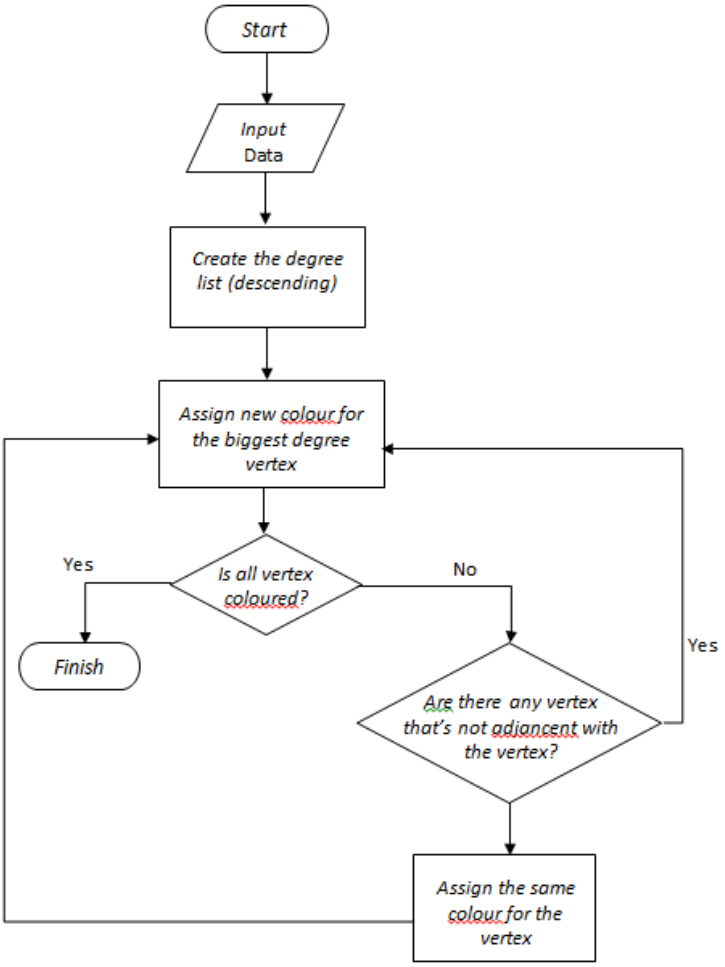
Pada pewarnaan *vertex* banyak digunakan teorema-teorema yang pada dasarnya bertujuan untuk menentukan batas jumlah warna maksimum yang dapat digunakan oleh sebuah *graph*. Meskipun yang ingin dicari pada pewarnaan *vertex* adalah jumlah warna paling minimum yang harus dipakai agar lebih efisien, namun jumlah warna minimum tersebut belum dapat ditentukan jika *graph* belum diwarnai. Karenanya, sebelum mewarnai *vertex* pada *graph*, perlu ditentukan batas jumlah warna yang diperbolehkan pada suatu *graph*. Salah satu *teorema* yang paling dikenal adalah *teorema Brook* [12]. *Teorema Brook* menyatakan bahwa jika suatu *Graph* merupakan simple *graph*, yaitu sebuah *Graph* yang tidak memiliki lebih dari satu *edge* yang menghubungkan dua buah *vertex*, maka batasan jumlah maksimum warna yang dapat digunakan pada *graph* tersebut dapat dirumuskan sebagai berikut [12]:

$$\chi(G) \leq d + 1 \quad (3)$$

dan jika *graph* yang ada bukan merupakan *simple graph*, maka batasan jumlah maksimum warna yang dapat digunakan pada *graph* ini dapat dirumuskan sebagai berikut [12]:

$$\chi(G) \leq d \quad (4)$$

keterangan :
$\chi(G)$: angka kromatik, jumlah minimum warna yang dapat dipakai pada *graph*
d : derajat terbesar dari *vertex-vertex* pada *Graph*

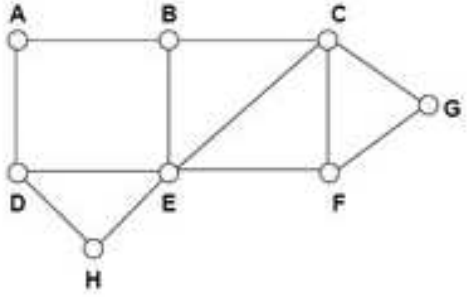


Gambar 2. *Flow chart* Algoritma Welsh-Powell.

Pada pewarnaan *vertex*, terdapat banyak algoritma yang dapat digunakan. Namun, pada umumnya algoritma yang paling sering digunakan untuk pewarnaan *vertex* ini adalah algoritma Welsh–Powell [13]. Langkah-langkah penyelesaian pewarnaan *vertex* menggunakan algoritma Welsh-Powell memiliki beberapa 3 tahapan [13]. Pertama, urutkan semua simpul berdasarkan derajatnya, dari derajat besar ke derajat kecil. Langkah kedua, ambil warna pertama (misalnya merah), warnai simpul pertama yang sudah terurut berdasarkan derajatnya tadi. Kemudian warnai simpul berikutnya yang tidak berdampingan dengan simpul pertama tadi dengan warna yang masih sama (merah). Kemudian lanjutkan pewarnaan dengan warna kedua, dan seterusnya, sampai semua simpul telah diberi warna

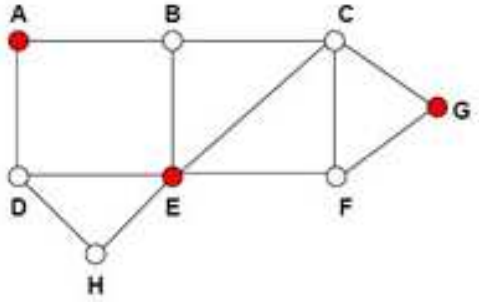
Langkah-langkah di atas dapat dibuat menjadi sebuah bagan alur yang dapat dilihat pada gambar 2. Sebagai contoh, dimisalkan terdapat sebuah *graph* yang memiliki delapan *vertex* seperti pada gambar 3.

Agar dapat mewarnai seluruh *vertex* tanpa adanya *vertex* bersebelahan yang memiliki warna yang sama, maka dipergunakan algoritma Welch – Powell. Adapun langkah-langkah penyelesaian *Graph Coloring* pada gambar 3 dengan menggunakan algoritma Welch - Powell terdiri dari beberapa langkah.



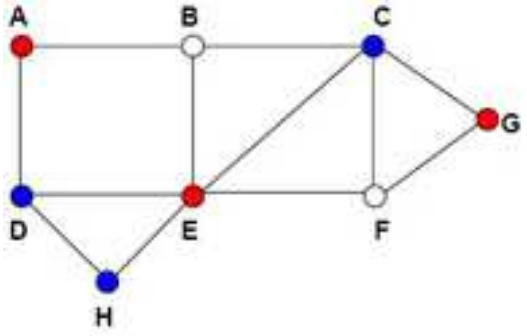
Gambar 3. Sebuah *Graph* dengan delapan *vertex.*

Pertama, urutkan *vertexs* di atas dari *vertex* yang memiliki derajat terbesar (banyaknya *edge* yang terhubung pada *vertex*) hingga yang terkecil. Dari gambar di atas diketahui bahwa urutan *vertexs* yang sudah disusun adalah : E, C, B, D, F, A, H, G. Maka pewarnaan dimulai dari *vertex* dengan derajat terbesar terlebih dahulu yaitu E. Tentukan sebuah warna yang akan diberikan pada *vertex* E, misalnya warna merah. Kemudian, *vertex* E diwarnai dengan warna merah. Selanjutnya, ditentukan *vertex* yang tidak bertetangga dengan *vertex* E dan diberi warna merah juga. Dalam *graph* ini hanya *vertex* G dan *vertex* A yang tidak bertetangga dengan *vertex* E. Gambaran dari langkah kedua tersebut dapat dilihat pada gambar 4.



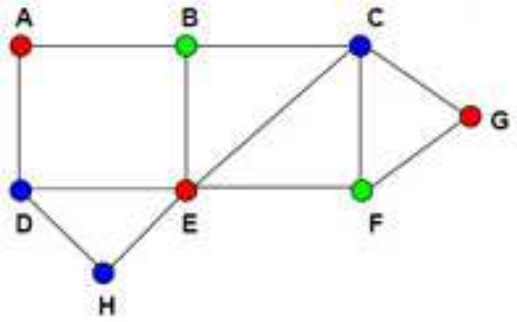
Gambar 4. *vertex* A, E, dan G selesai diwarnai.

Selanjutnya, abaikan *vertex* yang sudah diberi warna sehingga urutan *vertex* yang tersisa adalah : C, B, D, F, H. Karena *vertex* C kini merupakan *vertex* dengan derajat terbesar, maka *vertex* C diwarnai selanjutnya. Warna keduapun ditentukan dan berbeda dengan warna pertama, dalam hal ini warna biru dipilih sebagai warna kedua. Kemudian, tentukan *vertex-vertex* lain yang tidak bertetangga dengan *vertex* C. Pada *Graph* ini, *vertex* yang tidak bertetangga dengan *vertex* C hanya *vertex* D dan *vertex* H, seperti yang terlihat pada gambar 5.



Gambar 5. *vertex* C, D, H selesai diwarnai.

Pada langkah keempat, *vertex* yang tersisa hanya *vertex* B dan F. *vertex* B dan F memiliki derajat yang sama, sehingga bebas ditentukan *vertex* mana yang akan diwarnai terlebih dahulu. Warna ketiga ditentukan dan harus berbeda dari warna pertama dan warna kedua, dalam hal ini warna hijau dipilih sebagai warna ketiga. *vertex* B diberi warna hijau. Dan karena *vertex* F tidak bertetangga dengan *vertex* B, maka *vertex* F diberikan warna yang sama dengan *vertex* B seperti yang terlihat pada gambar 6.



Gambar 6. *vertex* B dan F selesai diwarnai.

Setelah seluruh *vertex* selesai diwarnai, dapat dilihat bahwa jumlah warna minimum yang diperlukan oleh *graph* pada contoh di atas adalah tiga warna yang berbeda. Sehingga angka kromatik yang dimiliki oleh *graph* di atas adalah:

$$\chi(G) = 3 \qquad (5)$$

Setiap penjadwalan, terutama untuk penjadwalan mata kuliah di sebuah universitas, memiliki batasan-batasan dan syarat-syarat tertentu, yang selanjutnya peneliti sebut sebagai *constraints*. Berdasarkan pengelompokkan harus dan tidak harusnya *constraints* tersebut dipenuhi, terdapat dua jenis, yaitu *hard constraints* dan *soft constraints*. *Hard constraints* adalah batasan-batasan dan syarat-syarat yang harus dipenuhi dalam penjadwalan mata kuliah tertentu. *Soft constraints* sebaliknya adalah batasan-batasan dan syarat-syarat yang tidak harus dipenuhi dalam penjadwalan mata kuliah tertentu.

Pembuatan model solusi dilakukan untuk memastikan bahwa *hard constraints* terpenuhi semua, dan *soft constraints* jika dimungkinkan pun dapat diikutsertakan seluruhnya. Model solusi yang dibuat didasarkan pada teknik *Graph Coloring* dengan algoritma Welsh-Powell [13].

Untuk model solusi yang diusulkan, terdapat dua poin yang ditambahkan pada algoritma Welsh–Powell [13] agar dapat memenuhi semua *hard constraints*, dan sebagian besar *soft*

constraints. Poin pertama adalah pembuatan matriks bobot hubungan antar dua simpul (untuk kasus penjadwalan mata kuliah, *node* tersebut adalah sebuah kelas mata kuliah). Poin kedua adalah optimalisasi penjadwalan, yaitu langkah yang memanfaatkan matriks bobot hubungan tersebut jika penggunaan warna melebihi total slot waktu yang dialokasikan dan jumlah simpul di satu atau beberapa warna melebihi jumlah ruang yang ditentukan.

Gambar 7 adalah model solusi berdasarkan modifikasi algoritma Welsh-Powell [13] yang diusulkan:

Diasumsikan terdapat:
m kelas mata kuliah M={M_1,M_2, …, M_m},
r ruang kuliah R={R_1,R_2, …, R_r}
h hari kuliah H={H_1,H_2, …, H_h}
s slot waktu kuliah S={S_1,S_2, …, S_s}

$matBobotRel_{i,j}$ adalah matriks bobot hubungan antar dua mata kuliah yang diinisialisasi dengan 0.

Terdapat 6 langkah penjadwalan. Pertama, tentukan himpunan *hard constraints*, yaitu *HARD*, dan *soft constraints*, *SOFT*. Kedua, *soft constraints*untuk pasangan (M_i, M_j) diurutkan berdasarkan M_i dan M_j yang dialikasikan ke dosen tidak tetap, untuk i=1,2, …, m; j=1,2, …, m, dan $i \neq j$:

```
if ((Mi, Mj) not statisfied HARD) then
matBobotReli,j = 3
else if ((Mi, Mj) not statisfied SOFT) then
matBobotReli,j = 1
```

Gambar 7. Kode program modifikasi algoritma Welsh-Powell.

Langkah ketiga, lakukan pewarnaan *graph* berdasarkan nilai-nilai $matBobotRel_{i,j}$ dengan cara urutkan semua *node* (kelas mata kuliah) berdasarkan derajatnya (Σ $matBobotRel_{k,l}$ untuk setiap nilai k di 1,2, …, m, sejumlah l di 1,2, …, m, dan $k \neq l$ dengan $matBobotRel_{k,l}$ = 3 and $matBobotRel_{k,l}$ = 1), dari derajat besar ke derajat kecil. Kemudian, ambil warna pertama (misalnya merah), warnai simpul pertama (M_i) yang sudah terurut berdasarkan derajatnya tadi. Kemudian warnai simpul berikutnya (M_j) yang tidak berdampingan ($matBobotRel_{i,j}$ = 0; $i \neq j$) dengan simpul pertama tadi dengan warna yang masih sama (merah). Lanjutkan pewarnaan dengan warna kedua, dan seterusnya, sampai semua simpul telah diberi warna. Catatan asumsikan diperoleh n kelompok kelas mata kuliah berwarna sama, yaitu W = { W_1, W_2, …, W_n }.

Langkah keempat, jika ($n > s*h$), lakukan optimalisasi sebagai berikut. Dapatkan t kelompok kelas mata kuliah berwarna sama dengan jumlah kelas mata kuliah terkecil, yaitu E = { E_1, E_2, …, E_t }, untuk $t = n - s*h$ dan $E \subset W$. Setelah itu, for each $e \in E_i$ do, dapatkan posisi e terbaik, yaitu posisi e di W_j dengan dengan jumlah derajat (Σ $matBobotRel_{e,l}$) terkecil, untuk setiap $l \in W_j$ dan $W_j \not\subset E$, lalu pindahkan e dari E_i ke W_j

Langkah kelima, jika terdapat u kelompok kelas mata kuliah berwarna sama dengan jumlah kelas mata kuliah lebih dari r ruang kuliah, F = { F_1, F_2, …, F_u }, for each $f \in F_i$ do, dapatkan posisi f terbaik, yaitu posisi f di W_j dengan dengan jumlah derajat (Σ $matBobotRel_{f,l}$) terkecil, untuk setiap $l \in W_j$ dan $W_j \not\subset F$, kemudian pindahkan f dari F_i ke W_j. Langkah keenam, alokasikan hari, waktu, dan slot waktu untuk setiap M_i. Cetak jadwal mata kuliah.

Berdasarkan hasil *survey* dari pihak Fakultas yang mengatur bagian penjadwalan mata kuliah, yaitu Jurusan Informatika, dan sejumlah mahasiswa FTI Untar dari berbagai angkatan, diperoleh sejumlah *constraints* yang dapat dikelompokkan sebagai *hard constraints* dan *soft constraints*.

Constraint pertama terkait dengan dosen, yaitu dosen yang sama dapat mengajar sejumlah mata kuliah baik di program studi Teknik Informatika dan Sistem Informasi. Oleh karena itu, jadwal kelas mata kuliah untuk dosen yang sama sudah tentu harus berbeda.

Constraint kedua terkait dengan jumlah ruang yang dapat digunakan untuk proses perkuliahan. Awalnya FTI Untar memiliki 18 ruangan yang dapat dipakai, mulai dari lantai 7 sampai dengan lantai 8, sehingga saat ini ada 12 ruang kuliah.

Jam kuliah yang diberlakukan oleh FTI Untar saat ini menggunakan sistem slot, yaitu Slot I pukul 08.00 – 11.00, Slot II pukul 11.00 – 14.00, dan Slot III pukul 14.00 – 17.00. Ketiga slot ini diterapkan untuk lima hari kuliah dari Senin sampai dengan Jumat. Slot dan hari kuliah tersebut menjadi *constraint* ketiga yaitu total slot waktu kuliah sejumlah 15 slot.

Constraint keempat, untuk mata kuliah dalam satu semester yang sama, untuk kelas yang berbeda dapat dibentrokan, tetapi jika sama, maka tidak boleh bentrokan. Jadi, diasumsikan mahasiswa yang mengambil kelas A untuk satu mata kuliah, selanjutnya harus mengambil mata kuliah lain di kelas A pula.

Selain itu, antara semester 1 dan 3 tidak boleh bertabrakan, semester 3 dan 5 tidak boleh bertabrakan, dan semester 5 dan 7 tidak boleh bertabrakan. Syarat-syarat ini merupakan *constraint* kelima.

Untuk mata kuliah pilihan, sebaiknya tidak ditabrakkan jadwalnya dengan mata kuliah di semester 5, 6, 7, dan 8, karena pada semester-semester tersebut, mahasiswa dapat mengambil mata kuliah tersebut. Syarat ini termasuk dalam *constraint* keenam.

Selanjutnya *constraint* ketujuh, khusus untuk program studi Teknik Informatika, saat ini setiap mata kuliah pilihan sudah dikelompokkan ke bidang kajian ilmunya masing-masing, sedangkan untuk program studi Sistem Informasi sampai saat ini belum ada pengelompokkan. Untuk mata kuliah pilihan yang termasuk dalam satu bidang kajian yang sama, tidak boleh ditabrakkan penjadwalannya.

Permintaan dosen akan hari dan slot waktu mengajar yang diinginkan menjadi *constraint* terakhir. *Constraint* ini dikhususkan untuk dosen-dosen tidak tetap (termasuk dosen lintas program studi).

Delapan *constraints* tersebut dikelompokkan menjadi dua: *hard constraints* (*constraint* pertama, kedua, ketiga, dan keempat) dan *soft constraints* (*constraint* kelima, keenam, ketujuh, dan kedelapan).

3. Hasil dan Pembahasan

Spesifikasi Program Aplikasi Penjadwalan. Untuk menerapkan usulan model solusi pada kasus FTI Untar, peneliti membuat sebuah program aplikasi penjadwalan mata kuliah di FTI Untar, dengan spesifikasi rancangan Visual C# .Net dan Microsoft SQL Server 2008 serta dua modul utama. Visual C# .Net sebagai bahasa pemrograman dan Microsoft SQL Server 2008 sebagai basis data. Dua modul utama, yaitu Modul Data dan Modul *Generate Schedule*. Pada Modul Data, *user* dapat mengatur berbagai macam *input* data yang perlu dimasukkan agar proses penjadwalan mata kuliah dapat dilakukan. Pengaturan *input* di dalam modul ini dibedakan menjadi beberapa sub-modul yaitu Sub-modul *input* data mata kuliah, Sub-modul *input* data dosen, Sub-modul *input* data Ruang, Sub-modul *input data* Jadwal Dosen Tidak Tetap, dan Sub-modul *input* data Kelas yang Diajar Dosen. Gambar 7 memperlihatkan tampilan layar untuk modul ini. Sedangkan, dalam modul *Generate Schedule*, model solusi pada Model Solusi Penjadwalan direalisasikan untuk *constraints* pada *Constraint* pada Penjadwalan Mata kuliah di FTI Untar. Gambar 8 memperlihatkan tampilan layar untuk modul ini. Pengujian *Black Box Testing* telah diterapkan untuk mencari jika ada kesalahan-kesalahan pada fungsi program, *interface,* maupun *output* program.

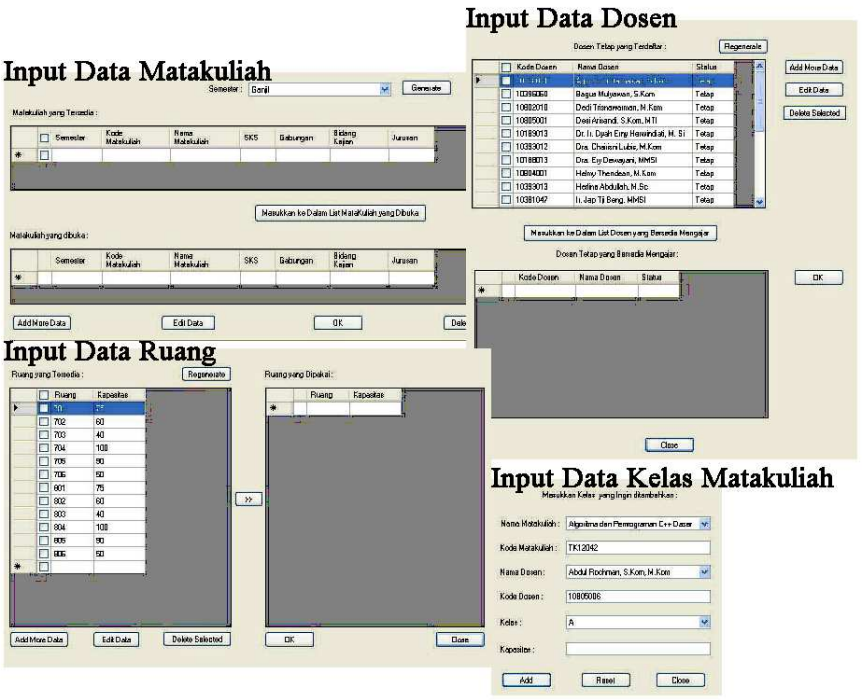
Hasil Percobaan untuk Data Mata Kuliah FTI Untar. Peneliti menggunakan data mata kuliah yang ditawarkan oleh program studi Teknik Informatika dan Sistem Informasi FTI Untar pada semester Ganjil dan Genap tahun akademik 2010/2011. Untuk semester Ganjil, terdapat 87 kelas mata kuliah dan 97 kelas mata kuliah untuk semester Genap.

Dari hasil uji coba yang diperoleh dapat dilihat bahwa penempatan kelas cukup merata dan setiap mata kuliah telah berhasil dijadwalkan serta tidak terdapat mata kuliah yang diajar oleh dosen yang sama diselenggarakan pada waktu bersamaan dan kelompok-kelompok kelas yang berwarna berbeda memiliki slotnya masing-masing. Tetapi, untuk pemanfaatan ruangan masih kurang optimal.

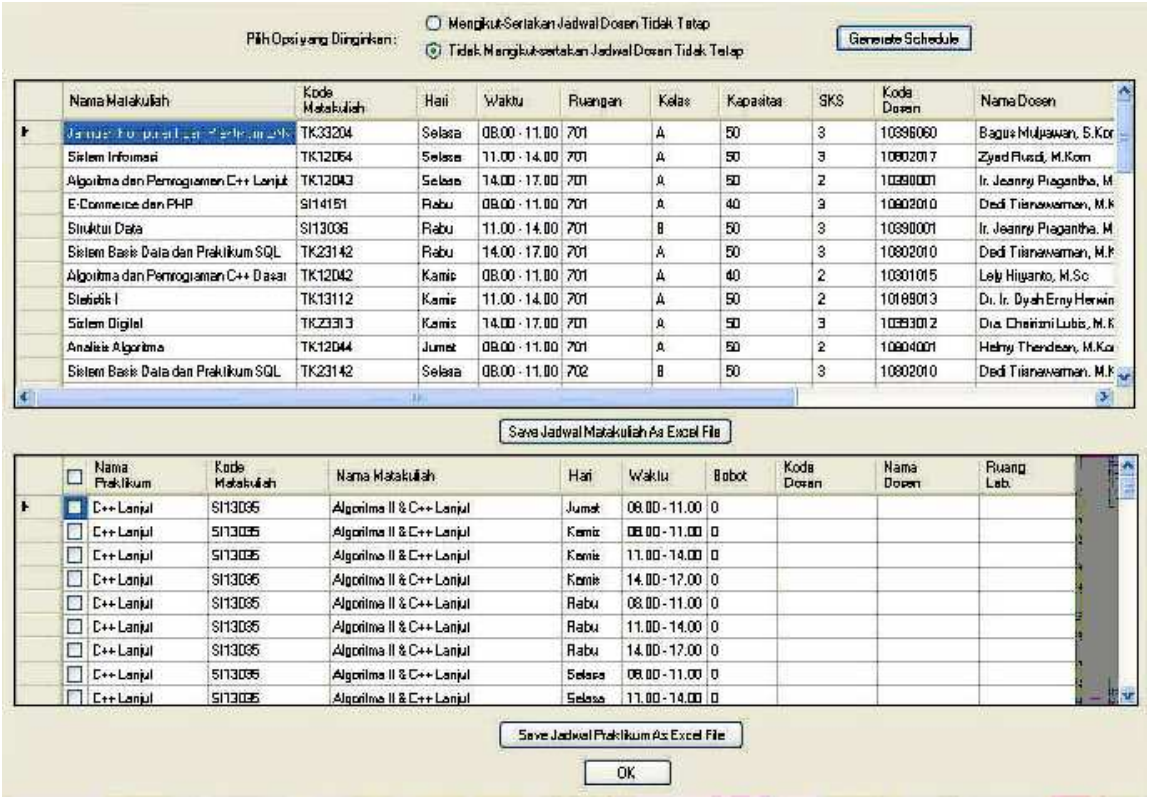
Dari hasil percobaan, dapat dilihat bahwa penjadwalan yang telah dihasilkan sudah memenuhi *contraints* yang ada. *Hard constraints* sudah tidak dilanggar pada program aplikasi ini, meskipun masih ada beberapa *soft constraints* yang terlanggar karena langkah optimalisasi yang dilakukan untuk menjadwalkan mata kuliah. Untuk semester Ganjil, terdapat 5 kelas mata kuliah dan 8 kelas mata kuliah untuk semester Genap yang melanggar *soft constraints*.

Optimalisasi tersebut dilakukan dikarenakan terdapat jumlah kelompok kelas berwarna sama melebihi total slot waktu yang ditetapkan oleh FTI Untar. Kelas-kelas mata kuliah untuk sejumlah kelompok kelas akan dijadwalkan ulang (*rescheduling*) dengan melanggar *soft constraints* yang ada, namun *hardconstraints* tidak akan dilanggar. Untuk kondisi satu atau lebih kelompok kelas mata kuliah berwarna sama memiliki kelas mata kuliah sebagai anggotanya melebihi jumlah ruang yang disediakan, tidak ditemukan dalam kasus penjadwalan mata kuliah FTI Untar.

Selain itu, ditemukan dari jadwal yang dihasilkan, terdapat sejumlah dosen yang harus mengajar lebih satu kali dalam satu hari. Peneliti menemukan bahwa mendahulukan penjadwalan untuk dosen tidak tetap membantu mengurangi permasalahan tersebut.



Gambar 8. Tampilan modul *Input.*



Gambar 9. Tampilan layar modul *Generate Schedule.*

Pengaturan tersebut juga dapat meminimalkan jumlah pelanggaran *soft constraints* dikarenakan adanya pemenuhan constraint kedelapan yaitu permintaan jadwal tertentu oleh dosen tidak tetap. Untuk kasus FTI Untar, pihak Jurusan tidak memiliki syarat berapa minimal dalam satu hari dosen yang sama harus mengajar.

Begitu pula keharusan dalam satu hari tidak boleh ada lebih dari satu kelas mata kuliah dalam semester yang sama yang diselenggarakan. Untuk sejumlah universitas, dua *constraints* tersebut dipertimbangkan dalam penyusunan jadwal.

Berdasarkan hasil survei kembali ke pihak Jurusan yang mengatur bagian penjadwalan mata kuliah, ditemukan bahwa adanya kebutuhan penggunaan slot waktu yang tidak dinamis, disesuaikan dengan jumlah SKS mata kuliah. Tapi, untuk sementara, pihak Jurusan dapat menggunakan solusi yang diusulkan ini dengan mengalokasikan kelas mata kuliah ber-SKS 2 di slot waktu yang ada yang memiliki rentang waktu 3 jam.

Pada penelitian ini, tiga syarat tambahan yang diuraikan di atas belum dipertimbangkan. Syarat tersebut dipertimbangkan untuk dicakup pada penelitian selanjutnya.

4. Kesimpulan

Berdasarkan hasil percobaan menggunakan data penawaran mata kuliah pada tahun akademik 2010/2011, dapat disimpulkan beberapa hal. Pertama, usulan model solusi penjadwalan mata kuliah di sebuah universitas melalui pengembangan metode *Graph Coloring* dapat digunakan untuk melakukan suatu penjadwalan mata kuliah yang memiliki berbagai macam *constraints* serta ketentuan yang harus dipenuhi dengan keterbatasan ruang dan waktu kuliah. Kedua, dengan model solusi tersebut, adanya kelas mata kuliah yang saling bertabrakan dapat dihindari karena metode ini memeriksa setiap kemungkinan bertabrakannya mata kuliah tersebut dengan mata kuliah yang lain. Ketiga, hasil percobaan untuk dua semester di tahun akademik tersebut menunjukkan bahwa rata-rata 93% kelas mata kuliah yang ditawarkan di masing-masing semester untuk kedua program studi dapat dijadwalkan tanpa melanggar *hard constraints* dan *soft constraints*. Keempat, rata-rata 7% kelas mata kuliah yang terjadwal, tapi harus melanggar *soft constraints* dikarenakan langkah optimalisasi untuk memenuhi *constraints* jumlah ruang, total slot waktu kuliah, dan permintaan jadwal tertentu oleh dosen tidak tetap.

Terakhir, usulan model solusi belum dapat memberikan pemanfaatan sumber daya, yaitu ruang kuliah yang lebih optimal. Untuk penelitian selanjutnya, peneliti berfokus melakukan percobaan diantaranya, fleksibilitas dalam penentuan slot waktu, tetapi tetap memenuhi *constraints* yang ditetapkan; pengembangan model solusi agar dapat lebih memanfaatkan sumber daya yang ada, terutama ketersediaan ruang kuliah; pembuatan model solusi penjadwalan kedua dengan menerapkan metode penjadwalan lainnya, yaitu *Particle Swarm Optimization*, untuk membandingkan optimalitas penjadwalan mata kuliah yang dihasilkan oleh model solusi pertama ini dengan adanya keterbatasan ruang dan waktu kuliah; serta pengembangan model solusi untuk kasus penjadwalan selain penjadwalan mata kuliah di universitas.

Referensi

[1] E. Burke, D. Elliman, & R. Weare, "A Genetic Algorithm Based University Timetabling System" *In 22nd East-West Conference on Computer Technologies in Education*, pp. 35-40, 1994.

[2] V.V. Peteghem and M. Vanhoucke, "A Genetic Algorithm for Resource-Constrained Project Scheduling Problem", http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.2444&rep=rep1&type=pdf, 2008, retrieved January 3, 2011.

[3] E. Burke, D. Ellman, R. Weare, "The Automated Timetabling of University Exams using a Hybrid Genetic Algorithm" *In Proceedings of the Sixth International Conference on Genetic Algorithms*, 1995.

[4] D. Kordalewski, C. Liu, K. Salvesen, "Solving an Exam Scheduling Problem Using a Genetic Algorithm" *In Proceeding of the Sixth International Conference on Industrial and Engineering Aplications of Artificial Intelligence and Expert Systems*, 1993.

[5] K. Mahar, "Automatic Generation of University Timetables: An Evolutionary Approach" *In IADIS International Conference Applied Computing*, pp. 570-574, 2006.

[6] J.N.D. Gupta, R. S. Sexton, E. A. Tunc, "Selecting Scheduling Heuristics using Neural Networks, " *Informs Journal on Computing,* vol. 12, pp. 150-162, 2000.

[7] M. Malkawi, M.A. Hassan, O.A. Hassan, "A New Exam Scheduling Algorithm Using *Graph Coloring*," *The International Arab*

Journal of Information Technology, vol. 5, pp. 11-68, 2008.

[8] B. Koyuncu & M. Seçir, "Student Time Table By Using *Graph Coloring* Algorithm," *In 5th International Conference on Electrical and Electronics Engineering – ELECO*, 2007.

[9] A. Dandashi & M.A. Mouhamed, "Graph Coloring for Class Scheduling" *In AICCSA '10 Proceeding of the ACS/IEEE International Conference on Computer Systems and Applications – AICCSA*, pp. 1-4, 2010.

[10] M.O. Albertson & J.P. Hutchinson, *Discrete Mathematics with Algorithms, John Wiley & Sons, New Jersey, 1988.*

[11] *J.A. Bondy & U.S.R. Murty, Graph Theory with Applications, Elsevier Sc*ience Publishing Co., Inc., Amsterdam, 1982.

[12] P. H*ajnal & E. Szemerédi, "Brooks coloring in paral*lel," *SIAM Journal on Discrete Mathematics*, vol.3, pp. 74-80, 1990.

[13] D.J.A. Welsh & M.B. Powell, "An upper bound for the chromatic number of a *graph* and its application to timetabling problems," *The Computer Journal,* vol.10, pp. 85–86, 1967.