

PENERAPAN ALGORITMA RUNUT BALIK DALAM KASUS PENJADWALAN KULIAH

Bofandra

Mahasiswa S1 Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Dipatiukur no.29
If16043@students.if.itb.ac.id

ABSTRAK

Masalah penjadwalan kuliah merupakan masalah sangat kompleks yang hingga saat ini masih merupakan sebuah topik yang banyak dibahas dalam berbagai thesis, disertasi, dan karya ilmiah di seluruh penjuru dunia. Inti dari penjadwalan kuliah adalah menjadwalkan sejumlah komponen yang terdiri atas mahasiswa, dosen, ruang, dan waktu dengan sejumlah batasan dan syarat (*constraint*) tertentu. Tulisan ilmiah ini akan mencoba membahas pemecahan masalah penjadwalan kuliah dengan pendekatan ilmu Intelegensia Semu (*Artificial Intelligence*), yakni dengan menggunakan *Constraint Satisfaction Problem*. Penulis mencoba sebuah teknik baru yang merupakan pengembangan dari teknik pencarian solusi dengan *heuristic search* yang dikombinasikan dengan teknik *Smart Backtracking* dan *Look Ahead* yang diutarakan oleh Robert s yang diberi nama *Intelligent Search*.

Kata kunci: *Smart Backtracking*, Pewarnaan Graf, *Heuristic*.

1. PENDAHULUAN

Masalah penjadwalan kuliah terlihat seperti masalah yang biasa yang dapat diselesaikan dengan metoda pemikiran biasa, akan tetapi jika sudah dalam jumlah data yang banyak dan saling berkaitan akan memerlukan banyak waktu untuk menyelesaikannya.

Maka dari itu digunakan masalah pewarnaan dalam graf yang dapat menyelesaikan sebuah penjadwalan. Dalam contoh penjadwalan dalam perkuliahan yang mempunyai *Constraint* yang banyak seperti sistem mata kuliah yang menggunakan sistem paket membuat permasalahan semakin rumit. Jika permasalahan ditambah oleh waktu yang dapat dihadiri oleh sejumlah dosen yang hanya dapat mengajar pada waktu tertentu, kemudian banyaknya mata kuliah yang dapat diajarkan oleh dosen yang sama, kemudian jumlah mahasiswa yang terdaftar dalam suatu kelas, ditambah lagi permasalahan ruangan dan waktu kuliah. Permasalahan sebanyak itu tidak dapat

diselesaikan dengan hanya corat-coret di kertas untuk menentukan pewarnaan dalam graf.

Dengan penggunaan komputer saat ini, dapat digunakan untuk membuat aplikasi komputasi untuk masalah pewarnaan agar memudahkan dalam implementasi. Pada implementasi pewarnaan graf dalam suatu aplikasi penjadwalan diperlukan algoritma yang dapat digunakan dalam permasalahan pewarnaan. Jadi dengan komputasi pada komputer yang cepat saja tidak cukup. Dengan pemanfaatan algoritma yang bagus maka akan terdapat hasil yang lebih baik. Pada analisis kali ini, Penulis menggunakan algoritma runut balik atau yang lebih dikenal dengan algoritma *Backtracking*.

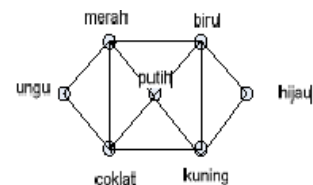
2. Teori

2.1 Pewarnaan Graf dan Penjadwalan

Ada tiga macam pewarnaan graf, yaitu pewarnaan simpul, pewarnaan sisi, dan pewarnaan wilayah (*region*).

Pewarnaan simpul adalah memberi warna pada simpul-simpul suatu graf sedemikian sehingga tidak ada dua simpul bertetangga mempunyai warna yang sama. Kita dapat memberikan sembarang warna pada simpul-simpul asalkan berbeda dengan simpul tetangganya.

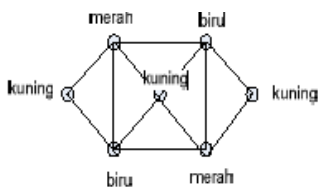
Dibawah ini sebuah graf yang simpul-simpulnya diberi dengan warna merah, biru, hijau, kuning, coklat, putih, dan ungu.



Gambar 1. Graf 1 Dengan Tujuh Warna

Dalam persoalan pewarnaan graf, kita tidak hanya sekedar mewarnai simpul-simpul dengan warna berbeda dari warna simpul tetangganya saja, namun kita juga menginginkan jumlah macam warna yang digunakan sesedikit mungkin. Gambar diatas menggunakan 7 macam warna, sebenarnya kita dapat menggunakan 3 macam

warna saja, yaitu merah, biru dan kuning. Jumlah warna minimum yang dapat digunakan untuk mewarnai simpul disebut bilangan kromatik graf G.



Gambar 2. Graf 1 Dengan Tiga Warna

Persoalan yang mempunyai karakteristik seperti pewarnaan graf adalah persoalan menentukan jadwal ujian. Misalkan terdapat delapan orang mahasiswa (1,2,...,8) dan lima buah mata kuliah yang dapat dipilihnya (A,B,C,D,E).

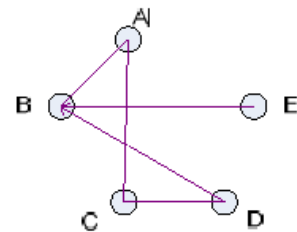
Tabel berikut memperlihatkan matriks lima mata kuliah dan delapan orang mahasiswa. Angka 1 pada elemen (i,j) berarti mahasiswa i memilih mata kuliah j, sedangkan angka 0 menyatakan mahasiswa tidak memilih mata kuliah j.

Tabel Mata-Kuliah Terambil

	A	B	D	E	F
1	0	1	0	0	1
2	0	1	0	1	0
3	0	0	1	1	0
4	1	1	0	0	0
5	0	1	0	1	0
6	0	0	1	1	0
7	1	0	1	0	0
8	0	0	1	1	0

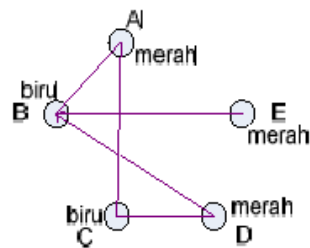
Berdasarkan tabel diatas, admin mata kuliah ingin menentukan jadwal ujian sedemikian sehingga semua mahasiswa dapat mengikuti ujian mata kuliah yang diambilnya tanpa bertabrakan waktunya dengan jadwal ujian kuliah lain yang juga diambilnya. Jika ada mahasiswa yang mengambil dua buah mata kuliah atau lebih, jadwal ujian mata kuliah tersebut harus pada waktu yang tidak bersamaan.

Ujian dua buah mata kuliah dapat dijadwalkan pada waktu yang sama jika tidak ada mahasiswa yang sama yang mengikuti ujian dua mata kuliah itu. Penyelesaian persoalan menentukan jadwal ujian semua mata kuliah sama dengan menentukan bilangan kromatis graf. Kita dapat menggambarkan graf yang menyatakan penjadwalan ujian. Simpul-simpul pada graf menyatakan mata kuliah. Sisi yang menghubungkan dua buah simpul menyatakan ada mahasiswa yang memilih kedua mata kuliah itu.



Gambar 3. Graf Mata Kuliah

Berdasarkan graf tersebut kita menyimpulkan, bahwa apabila terdapat dua buah simpul dihubungkan oleh sisi, maka ujian kedua mata kuliah itu tidak dapat dibuat pada waktu yang sama. Warna-warna yang berbeda dapat diberikan pada simpul graf yang menunjukkan bahwa waktu ujiannya berbeda. Diinginkan jadwal ujiannya sesedikit mungkin untuk memudahkan pelaksanaannya. Jadi kita harus menentukan bilangan kromatis graf. Bilangan kromatik untuk graf jadwal ujian tersebut adalah dua. Jadi, ujian mata kuliah A, E, dan D dapat dilaksanakan bersamaan, sedangkan ujian mata kuliah B dan C dilakukan bersamaan tetapi pada waktu yang berbeda dengan mata kuliah A, E, dan D. Berikut ini graf yang telah diwarnai.



Gambar 3. Graf Mata Kuliah Setelah Pewarnaan

Untuk graf dengan jumlah simpul sedikit, kita dapat menentukan bilangan kromatiknya dengan mudah. Tetapi, untuk graf yang besar, kita perlu membuat program komputer untuk menentukan bilangan kromatik tersebut. Terakhir, dilakukan pencarian apakah masih tersedia ruangan yang kosong pada waktu yang telah dihitung sebelumnya.

2.2 Algoritma Backtracking

Penggunaan metoda backtracking dimaksudkan untuk mengetahui seberapa efektif penggunaannya dalam mengatasi masalah penjadwalan yang bersifat acak, dengan keterkaitan yang rumit.

Algoritma *backtrack* pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950. Dalam perkembangannya beberapa ahli seperti RJ Walker, Golomb, dan Baumert menyajikan uraian umum tentang *backtrack* dan menerapkannya dalam berbagai persoalan dan aplikasi. Algoritma *backtrack* (runut balik) merupakan salah satu

metode pemecahan masalah yang termasuk dalam strategi yang berbasis pencarian pada ruang status. Algoritma *backtrack* bekerja secara rekursif dan melakukan pencarian solusi persoalan secara sistematis pada semua kemungkinan solusi yang ada. Oleh karena algoritma ini berbasis pada algoritma *Depth-First Search* (DFS), maka pencarian solusi dilakukan dengan menelusuri suatu struktur berbentuk pohon berakar secara *preorder*. Proses ini dicirikan dengan ekspansi simpul terdalam lebih dahulu sampai tidak ditemukan lagi suksesor dari suatu simpul.

Dengan cara menelusuri algoritma *backtracking*, dan pengetahuan dari teori algoritma pencarian lainnya, maka akan didapat cara yang efektif dari penelusurannya.

3. Metode

Berdasarkan jumlah dan jenis *constraint* yang ada, Penulis menggunakan pendekatan *heuristic* untuk memecahkan masalah. Kombinasi Kelas - Matakuliah ditetapkan sebagai variabel yang harus diisi dengan nilai-nilai dosen, waktu, dan ruang. Untuk dosen terdapat sejumlah parameter yang harus diperhatikan dalam pengalokasian jadwal, yaitu waktu kesediaan dosen untuk mengajar (tersimpan dalam sebuah tabel yang didapat dari isian dosen itu sendiri), jatah SKS yang merupakan batas maksimum dosen tersebut mengajar dalam 1 minggu (ditentukan oleh ketua jurusan), dan matakuliah yang dapat diajar oleh dosen tersebut. Masalah masih bertambah kompleks dengan adanya sejumlah matakuliah yang harus dialokasikan di ruangan tertentu yang memiliki kriteria tersendiri.

Ide dasar dari *heuristic* yang Penulis kembangkan adalah melakukan pencarian dengan urutan dosen - waktu - ruang. Pasangan kelas - matakuliah telah terlebih dahulu diurutkan sesuai dengan jumlah *conflict possibility* yang ada. Tiap kelas dipasangkan dengan sejumlah dosen yang dapat mengajar matakuliah tersebut, kemudian sejumlah perhitungan *heuristic* dilakukan untuk mencari waktu kuliah yang memenuhi *constraint* dosen dan *constraint* mahasiswa. Terakhir, dilakukan pencarian apakah pada waktu yang telah dihitung sebelumnya masih tersedia ruangan yang kosong.

Proses perhitungan di atas dirancang untuk sedapat mungkin menghindari terjadinya *backtracking* yang akan memperlambat proses. Pada setiap langkah percabangan, dilakukan proses *breadth first search* (BFS) untuk mengambil waktu yang terbaik untuk diproses terlebih dahulu. Kemudian langkah selanjutnya merupakan proses *depth first search* (DFS) dengan pencarian isi variabel pada level di bawahnya.

Meskipun telah diupayakan untuk tidak terjadi, kemungkinan terjadinya *backtracking* tetap masih ada. Untuk itu digunakan sebuah algoritma yang tidak secara otomatis naik satu level seperti yang terjadi pada *search*

tree pada umumnya, akan tetapi program akan mencari pasangan Kelas - Matakuliah yang memiliki *intersection* dosen, paket, atau matakuliah dengan variabel yang menyebabkan terjadinya *backtracking*. Algoritma ini merupakan sebuah bentuk baru dari *Intelligent Backtracking* yang sudah dikenal sebelumnya dalam dunia Intellegensia Semu.

Banyaknya jumlah dan tingkat kompleksitas *constraint* yang ada membuat Penulis harus melakukan klasifikasi *constraint* tersebut dalam kategori *hard constraint* (harus terpenuhi) dan *soft constraint* (diupayakan untuk terpenuhi). Adapun yang Penulis kategorikan sebagai *hard constraint* adalah tidak boleh adanya bentrok waktu dan ruang mengajar dosen dan perkuliahan mahasiswa. Jumlah jam mengajar dosen per hari matakuliah dengan ruangan khusus, serta jatah SKS dosen juga Penulis masukkan ke dalam *hard constraint*.

Urutan jam kuliah dan lokasi kampus Penulis kategorikan sebagai *soft constraint* yang tingkat toleransi kesalahannya dapat diubah-ubah. Hal ini didasarkan bahwa pencarian jadwal kuliah yang baik, urut, dan tidak berpindah kampus dengan tingkat kebenaran 100% akan memperlambat proses hingga mencapai waktu eksponensial berbanding jumlah kelas yang ada.

4. KESIMPULAN

Dengan mengetahui konsep pewarnaan graf maka akan mengerti bagaimana membuat algoritma yang efektif dengan memanfaatkan ilmu algoritma yang ada.

Dari penelitian ini Penulis berkesimpulan bahwa algoritma *heuristic* dengan *smart backtracking* dapat menangani dengan baik untuk masalah penjadwalan yang sebuah kegiatan seperti pada perkuliahan dalam suatu universitas yang mempunyai kelas dan jurusan namun terbatas dalam dosen dan ruang kelas agar bisa didapatkan hasil yang maksimal. Dalam beberapa hal algoritma ini masih memerlukan penyempurnaan seperti efisiensi waktu (menghindari peningkatan *process time* yang eksponensial), dan usaha memenuhi *soft constraint* dengan lebih baik..

REFERENSI

- [1] Munir, Rinaldi. 2006. *Strategi Algoritmik*. Teknik Informatika ITB: Bandung. Nama, "Judul Makalah", Nama jurnal, Volume, Nomor, Tahun terbit, halaman.
- [2] <http://www.robertsetiadi.net/articles/snkk.htm>