

Penerapan Algoritma Greedy untuk Menentukan Penjadwalan Kelas Gedung Labtek V

Albhikautsar Dharma Kesuma - 13511058
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
albhi@students.itb.ac.id

Abstract—Penjadwalan adalah suatu proses menentukan bagaimana menempatkan suatu sumber daya diantara beberapa kemungkinan dari suatu tugas, Pemanfaatan fasilitas yang sudah dimiliki oleh Institut Teknologi Bandung kepada Teknik Informatika pula harus di berdayakan semaksimal mungkin. Makalah ini akan membahas mengenai penjadwalan penggunaan ruangan kuliah pada semester II tahun ajaran 2013-2014

Index Terms—Institut Teknologi Bandung, Teknik Informatika, Penjadwalan

Teknik Informatika Institut Teknologi Bandung yang terletak pada Labtek V memiliki mahasiswa sejumlah sekitar 100 setiap angkatannya, Jadi Teknik Informatika Insitut Teknologi Bandung memiliki mahasiswa bejumlah sekitar 300 orang. Terdapat dua ruangan yang cukup besar yang dapat menampung mahasiswa yang cukup banyak tersebut yang sering dipakai untuk kegiatan perkuliahan yang biasanya pembagian kelas dilakukan berdasarkan NIM dari setiap mahasiswa. Dua ruangan tersebut adalah ruangan 7602 dan 7606.

I. PENDAHULUAN

Accreditation Board for Engineering and Technology adalah sebuah badan yang non-pemerintah yang memberikan akreditasi pada program studi yang bergerak dalam bidang sains terapan, komputasi, dan teknik.

Saat ini ABET tidak hanya memberikan akreditasi untuk Amerika Serikat saja, namun sudah bersifat internasional. Hingga saat ini ABET sudah memberika lebih dari 3100 program studi di lebih 670 Universitas dan lebih dari 24 negara. ABET memberikan akreditasi melalui evaluasi yang dilakukan pada satuan program studi, bukan satu institusi secara keseluruhan. Dengan akreditasi ABET ini, sebuah program studi dari suatu universitas sudah dipastikan memiliki kualitas standard yang bersifat Internasional yang akan didapatkan oleh para mahasiswa dan mahasiswinya.



Logo ABET

Pada tahun 2013, Teknik Informatika Institut Teknologi Bandung melakukan permintaan untuk akreditasi dari ABET dan hasil dari akreditasi ini, hingga makalah ini ditulis masih dalam proses pemberian sertifikasi dari ABET. Namun seperti yang sudah dijelaskan pada paragraf sebelumnya, ABET memiliki beberapa standard yang harus dipenuhi untuk mendapatkan akreditasi tersebut.



Foto Ruang 7602



Foto Ruang 7606

Pada Semester II Tahun Ajaran 2013-2014, Teknik Informatika memiliki empat belas mata kuliah wajib yang harus diadakan dan total SKS yang di adakan adalah 41. Dengan jumlah dosen asumsi sebanyak dua puluh dengan asumsi tidak dosen yang mengajar lebih dari tujuh jam.

Berikut adalah mata kuliah yang akan diajarkan pada semester II tahun ajaran 2013-2014 :

1. Proyek Perangkat Lunak (2011)
2. Intelegensi Buatan (2011)
3. Grafika Komputer (2011)
4. Sistem Paralel dan Terdistribusi (2011)
5. Sistem Informasi (2011)
6. Socio-Informatika dan Profesionalisme (2011)
7. Pemograman Berorientasi Objek (2012)
8. Strategi Algoritma (2012)
9. Teori Bahasa Formal dan Otomata (2012)
10. Sistem Operasi (2012)
11. Basis Data (2012)
12. Rekayasa Perangkat Lunak (2012)
13. Tugas Akhir II (2010)

Pada makalah ini akan dijelaskan mengenai penjadwalan penggunaan ruangan kelas 7602 dan 7606 untuk angkatan 2010, 2011, dan 2012 supaya dapat menampung jumlah mahasiswa yang cukup banyak tersebut.

II. LANDASAN TEORI

2.1 Algoritma Greedy

Algoritma greedy adalah salah satu metode yang populer untuk memecahkan persoalan optimasi. Maksud dari pemecahan persoalan optimasi sendiri adalah mencari solusi paling optimum dari segala kemungkinan yang ada. Ada dua macam persoalan optimasi dalam algoritma greedy ini. Persoalan pertama adalah pemecahan masalah dengan *Maximization* atau memaksimalkan segala kemungkinan yang mungkin terjadi kedepannya. Lalu persoalan kedua adalah pemecahan masalah dengan *Minimization* atau minimalisasi dari segala kemungkinan yang akan terjadi kedepannya.

Sesuai dengan namanya, *greedy*, dalam bahasa Indonesia artinya rakus, tamak, dan loba, maka algoritma ini memiliki prinsip "*Take what you can get now!*". Algoritma Greedy ini membentuk solusi langkah per langkah, pada setiap langkahnya tentu *path* tersebut akan memiliki banyak pilihan dan kemungkinan yang dapat di eksplorasi, dengan algoritma ini keputusan langkah yang diambil berikutnya adalah yang paling menguntungkan pada keadaan sekarang.

Harapan dari pengambilan langkah optimum lokal adalah didapatkannya solusi optimum global (hasil dari keseluruhan langkah) yang optimal pula.

Kekurangan dari algoritma ini adalah tidak adanya penanganan konsekuensi dari setiap langkah yang diambilnya. Karena belum tentu setiap langkah optimum lokal yang diambil akan memberikan solusi optimum global.

2.1.1 Teori Dasar Algoritma Greedy

Dalam mengimplementasikan Algoritma ini, Greedy memiliki beberapa elemen, yaitu :

1. Himpunan Kandidat, C.
2. Himpunan Solusi, S.

3. Fungsi Seleksi (*selection function*)
4. Fungsi Kelayakan (*feasible*)
5. Fungsi Obyektif

Himpunan kandidat merupakan himpunan kandidat solusi yang merepresentasikan opsi yang ditawarkan setiap langkahnya. Himpunan solusi merepresentasikan langkah apa saja yang telah diambil hingga akhir algoritma.

Fungsi seleksi diperuntukan untuk menyeleksi kumpulan pilihan yang ada dan mengembalikan pilihan yang memberikan solusi optimal. Fungsi kelayakan bertindak sebagai pembatas, apakah pilihan yang dipilih layak sesuai ketentuan dari persoalan. Kombinasi dari kedua fungsi ini akan menghasilkan solusi optimum pada setiap langkahnya. Solusi-solusi optimum untuk tersebut akan menjadi hasil dari sebuah fungsi objektif, yaitu persoalan utama yang ingin diselesaikan.

Berikut adalah Skema Umum Algoritma Greedy :

function greedy(input C: himp_kandidat) → himp_kandidat

{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy. Masukan: himpunan kandidat C Keluaran: himp. Solusi yang bertipe himp_kandidat }

Deklarasi

x : kandidat
S : himp_kandidat

Algoritma

S ← { } {inisialisasi S dengan kosong}

while (not SOLUSI(S)) and (C!={}) **do**
 x ← SELEKSI(C) {pilih kandidat dari C}
 C ← C - {x} {elemen C berkurang satu}

if LAYAK(S U {x}) **then**
 S ← S U {x}

endif

endwhile

{SOLUSI(S) atau C kosong}

if SOLUSI(S) **then**
 return S

else

 {tidak ditemukan solusi global}

endif

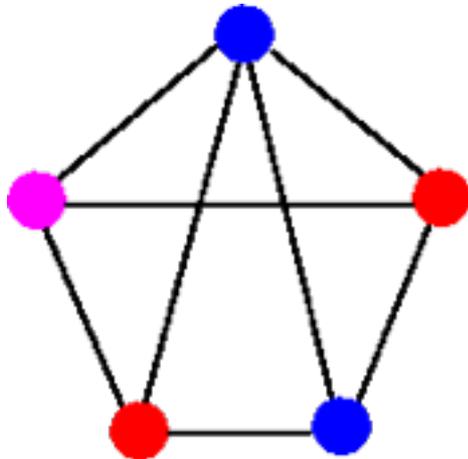
Pertama algoritma greedy akan melakukan iterasi untuk seluruh himpunan kandidat C. Lalu satu-persatu elemen pada himpunan kandidat C akan dilakukan pengecekan oleh fungsi seleksi terlebih dahulu apakah elemen tersebut adalah optimum lokal. Lalu dilakukan pengecekan apakah elemen tersebut oleh fungsi kelayakan. Apabila kandidat tersebut cocok, maka kandidat akan dimasukkan ke dalam

himpunan solusi.

Hal ini akan terus dilakukan hingga seluruh elemen pada himpunan kandidat habis atau solusi yang dituju sudah tercapai atau solusi sudah ditemukan.

2.2 Graph Coloring

Dalam teori graf, pewarnaan graf merupakan kasus khusus dari *graph labeling*. Pewarnaan graf ini adalah penugasan mengenai mewarnai graf sedemikian rupa sehingga titik-titik yang saling berhubungan tidak memiliki warna yang sama. Terdapat dua macam dalam pewarnaan graf yaitu, pewarnaan simpul dan pewarnaan sisi. Pengaplikasian pewarnaan graf biasanya dilakukan untuk pewarnaan peta supaya terlihat perbedaan antara negara dapat terlihat. Selain digunakan untuk mewarnai peta, pewarnaan graf juga bisa dilakukan untuk melakukan penjadwalan. Pada makalah ini, *graph coloring* akan digunakan untuk menentukan penjadwalan mengenai penggunaan kelas supaya tidak ada angkatan yang waktu kuliahnya berhimpitan.



Contoh Pewarnaan Graph

III. METODE PEMECAHAN MASALAH

Algoritma Greedy dapat digunakan dengan sebelumnya mendefinisikan kondisi apa yang dapat menyebabkan situasi menjadi optimum. Pada makalah ini akan dijelaskan dua buah kemungkinan yang dapat menyebabkan situasi menjadi optimum.

Sebelumnya akan dijelaskan batasan mengenai gambaran umum mengenai pembagian kelas yang nantinya akan di implementasikan pada seluruh algoritma *greedy* yang akan dijelaskan pada poin-poin berikutnya. Jadi dalam penjadwalan kelas ini mahasiswa dengan NIM ganjil akan menempati ruangan 7602 dan mahasiswa dengan NIM genap akan menempati ruangan 7606.

3.1. Greedy versi 1

Pemilihan algoritma *greedy* versi 1 ini adalah

penyelesaian *greedy* menggunakan parameter angkatan. Pemilihan ini berdasarkan dengan keadaan yang sekarang sudah dilakukan oleh kebijakan penggunaan kelas di Labtek V ini. Namun pada algoritma *greedy* ini diberikan beberapa batasan melalui fungsi kelayakan yang akan memudahkan baik bagi mahasiswa dan dosen.

function LAYAK(input K : himp_solusi) → boolean

{ *mengembalikan true jika mata kuliah memenuhi syarat-syarat yang diberlakukan* }

function greedy(input C: list of tuple persoalan) → himp_kandidat

{ *Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy dengan* }

Masukan : list of tuple dari seluruh mata kuliah beserta beban SKSnya

Keluaran : himpunan solusi.

Prekondisi : List of Tuple tidak kosong }

Deklarasi

S : himp_kandidat
x : tuple

Algoritma

```
S ← { }  
while (C!={}) do  
  x ← SELEKSI (Head(C))  
  C ← C - {x}  
  if LAYAK(S U {x}) then  
    S ← S U {x}  
  endif  
endwhile
```

```
if SOLUSI(S) then
```

```
  return S
```

```
else  
endif
```

Beberapa batasan yang menjadi fungsi kelayakan dari algoritma *greedy* ini adalah :

- Tidak ada mata kuliah yang berlangsung lebih dari dua jam
- Tidak ada mata kuliah yang berlangsung pada hari yang beriringan
- Angkatan 2011 hanya boleh kuliah dari dari pukul 07.00-13.00
- Angkatan 2012 hanya boleh kuliah dari pukul 13.00-18.00
- Angkatan 2010 mengisi jadwal kosong dengan mengikuti peraturan yang sudah disebutkan diatas

- Pengisian selalu di cek dimulai dari hari senin hingga jumat

Penyelesaian dengan menggunakan greedy versi pertama ini akan dilakukan seleksi pada head tuple. Tuple sudah tersusun sesuai dengan yang sudah dituliskan pada pendahuluan makalah ini. Pada contoh pertama head dari tuple tersebut adalah <Proyek Perangkat Lunak, 2011, 4>. Tuple tersebut akan di cek oleh fungsi kelayakan dengan melihat tahun ajaran dan jumlah SKS. Apabila tahun ajaran 2011 maka akan dilakukan pengisian jadwal dibawah pukul 13.00 dengan melihat hari yang kosong dimulai dari hari senin. Apabila tabel memungkan untuk diisi maka jumlah SKS akan dikurangi 2 lalu apa bila jumlah SKS belum sama dengan nol, Head List akan disimpan lagi ke paling belakang untuk dieksekusi kemudian.

Berikut adalah hasil penjadwalan penggunaan ruangan 7602 dan 7606 dengan menggunakan algoritma Greedy versi 1 :

	Senin	Selasa	Rabu	Kamis	Jumat
07.00-08.00	Proyek Perangkat Lunak	Sistem Terdistribusi	Proyek Perangkat Lunak	Grafika Komputer	Tugas Akhir 2
08.00-09.00	Proyek Perangkat Lunak	Sistem Terdistribusi	Proyek Perangkat Lunak	Sistem Terdistribusi	Tugas Akhir 2
09.00-10.00	Intelegensi Buatan	Sistem Informasi	Intelegensi Buatan	Sistem Informasi	
10.00-11.00	Intelegensi Buatan	Sistem Informasi	Intelegensi Buatan	Socio-Informatika	
11.00-12.00	Grafika Komputer	Socio-Informatika	Tugas Akhir 2		
12.00-13.00	Grafika Komputer	Socio-Informatika	Tugas Akhir 2		
13.00-14.00	Programar AI Berorientasi Objek	Teori Bahasa dan Otomata	Base Data	Programar AI Berorientasi Objek	
14.00-15.00	Programar AI Berorientasi Objek	Teori Bahasa dan Otomata	Base Data	Strategi Algoritma	
15.00-16.00	Strategi Algoritma	Sistem Operasi	Belajar Perangkat Lunak	Teori Bahasa dan Otomata	
16.00-17.00	Strategi Algoritma	Sistem Operasi	Belajar Perangkat Lunak	Sistem Operasi	
17.00-18.00	Base Data				

3.2. Greedy versi 2

Penyelesaian algoritma greedy versi 2 ini menggunakan parameter yang sama dengan greedy versi 1. Namun fungsi kelayakannya yang diubah pada penyelesaian algoritma ini.

function LAYAK(input K : himp_solusi) → boolean

{ mengembalikan true jika mata kuliah memenuhi syarat-syarat yang diberlakukan }

function greedy(input C : list of tuple persoalan) → himp_kandidat

{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy dengan

Masukan : list of tuple dari seluruh mata kuliah beserta beban SKSny

Keluaran : himpunan solusi.

Prekondisi : List of Tuple tidak kosong }

Deklarasi

S : himp_kandidat

x : tuple

Algoritma

```
S ← { }
while (C!={}) do
  x ← SELEKSI (Head(C))
  C ← C - {x}
  if LAYAK(S U {x}) then
    S ← S U {x}
  endif
endwhile
```

```
if SOLUSI(S) then
```

```
  return S
```

```
else
  endif
```

Beberapa batasan yang menjadi fungsi kelayakan dari algoritma greedy versi 2 ini adalah :

- Tidak ada mata kuliah yang berlangsung lebih lama dari 2 jam
- Tidak ada mata kuliah yang berlangsung pada hari yang beriringan
- Pemrosesan kelas tidak boleh satu angkatan beririsan.

Penggunaan algoritma greedy versi 2 ini berjalan sama persis dengan versi pertama. Head dari list tuple akan di proses namun perbedaan disini terletak pada fungsi kelayakan yang tidak memperbolehkan mata kuliah satu angkatan di proses secara beriringan. Contohnya apabila tuple sudah berhasil memproses angkatan 2011. Dan tuple sudah di masukan lagi ke list paling belakang sudah di buang karena jumlah SKSnya sudah 0, maka pemrosesan berikutnya harus melakukan proses untuk mata kuliah untuk angkatan 2012 atau 2010. Begitu pula untuk angkatan 2010 dan 2012. Untuk pemecahan masalah ini dapat juga digunakan graph coloring untuk penyelesaiannya dengan simpul-simpul yang digunakannya adalah jadwal ujian yang beririsan.

Jadi dapat digunakan sebuah titik-titik atau graf yang linear yang nantinya akan dilakukan pewarnaan sehingga tidak ada warna yang boleh sama apabila titik tersebut bersisian.

Berikut adalah hasil penjadwalan penggunaan ruangan 7602 dan 7606 dengan menggunakan algoritma Greedy versi 1 :

	Senin	Selasa	Rabu	Kamis	Jumat
07.00-08.00	Proyek Perangkat Lunak	Grafika Komputer	Basis Data	Perencanaan SD Semestrial Class	Basis Data
08.00-09.00	Proyek Perangkat Lunak	Grafika Komputer	Basis Data	Intelegensi Buitan	
09.00-10.00	Perencanaan SD Semestrial Class	Teori Bahasa dan Komputasi	Socio-Informatika	Intelegensi Buitan	
10.00-11.00	Perencanaan SD Semestrial Class	Teori Bahasa dan Komputasi	Socio-Informatika	Struktur Algoritma	
11.00-12.00	Tugas Akhir 2	Sistem Terdistribusi	Tugas Akhir 2	Grafika Komputer	
12.00-13.00	Tugas Akhir 2	Sistem Terdistribusi	Tugas Akhir 2	Teori Bahasa dan Komputasi	
13.00-14.00	Intelegensi Buitan	Sistem Operasi	Rekamasa Perangkat Lunak	Sistem Terdistribusi	
14.00-15.00	Intelegensi Buitan	Sistem Operasi	Rekamasa Perangkat Lunak	Sistem Operasi	
15.00-16.00	Struktur Algoritma	Sistem Informasi	Proyek Perangkat Lunak	Sistem Informasi	
16.00-17.00	Struktur Algoritma	Sistem Informasi	Proyek Perangkat Lunak		
17.00-18.00	Socio-Informatika				

IV. ANALISIS

Implementasi konsep algoritma greedy pada penjadwalan penggunaan kelas pada gedung Labtek V tidak terdapat hal khusus yang dapat menyulitkan atau bahkan dapat menyebabkan pengaturan kelas tidak dapat terwujud.

Pemrosesan algoritma greedy pada umumnya akan berkurang secara langsung setiap tahapnya. Namun pada penyelesaian persoalan ini dilakukan modifikasi sehingga apabila jumlah SKS belum terpenuhi maka set himpunan kandidat yang merupan tuple yang terdiri tiga parameter itu akan dilakukan pemrosesan ulang yang akan dilakukan di belakang. Namun penyelesaian pada tahap selanjutnya tetap dapat dilakukan dengan algoritma greedy.

Pada kasus 2, penyelesaian penjadwalan dapat juga dilakukan dengan menggunakan Pewarnaan Graf. Hal ini dilakukan karena adanya batasan-batasan tambahan yang dilakukan seperti jadwal kuliah tidak boleh satu angkatan berturut-turut atau harus bergantian.

Hasil penjadwalan pengaturan kelas ini dapat memberikan hasil yang berbeda-beda sesuai dengan batasan-batasan dan fungsi kelayakan yang diberikan. Dapat pula dilakukan perubahan, penambahan, maupun

pengurangan pada fungsi kelayakan ini. Hal ini dapat terlihat dari hasil percobaan dengan menggunakan dua algoritma greedy yang memiliki parameter dan fungsi kelayakan yang sama dapat memberikan hasil penjadwalan yang cukup signifikan berbeda.

Penyelesaian penjadwalan ini tidak hanya dapat diselesaikan dengan menggunakan algoritma greedy dan *graph coloring*. Penyelesaian ini dapat dilakukan dengan menggunakan *Dynamic Programming* untuk permasalahan yang lebih kompleks misalnya batasan waktu dosen untuk mengajar maupun batasan-batasan yang lainnya.

V. KESIMPULAN

Algoritma greedy pada sebagian hal tidak selalu memberikan hasil yang optimum. Namun untuk masalah penjadwalan ini tidak ada hasil yang terbaik karena adanya beberapa batasan yang harus dipenuhi supaya penjadwalan ini dapat terpenuhi. Penggunaan kombinasi dengan menggunakan algoritma lain juga dapat membantu menyelesaikan persoalan ini.

REFERENSI

- [1] www.kfupm.edu.sa diakses pada 18 Desember 2013 pukul 7.25
- [2] <http://www.abet.org/about-abet/> diakses pada 18 Desember 2013 pukul 7.20
- [3] rinaldimunir.wordpress.com diakses pada 18 Desember 2013 pukul 11.28
- [4] www.personal.kent.edu diakses pada 19 Desember 2013 pukul 11.18
- [5] <http://webdocs.cs.ualberta.ca/~joe/Coloring/index.html> diakses pada 19 Desember 2013 pukul 13.15

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2013



Albhikautsar Dharma Kesuma - 13511058