

Analisis Penggunaan Algoritma Backtracking dalam Penjadwalan Kuliah

Farhan Makarim 13515003
Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
Farhan_makarim@student.itb.ac.id

Abstrak—model simulasi menggunakan *backtracking* memberikan pendekatan alternatif untuk memberikan himpunan solusi yang *feasible*. Dalam makalah ini dianalisis dua strategi *backtracking* untuk menentukan penjadwalan kuliah yang efisien. Data penelitian yang sudah ada menunjukkan bahwa *backtracking* adalah sebuah pendekatan yang cukup efektif untuk meningkatkan kualitas penjadwalan ujian.

Kata Kunci—Penjadwalan Ujian, *Backtracking*, *CSP*

I. PENDAHULUAN

Penjadwalan kuliah diperlukan untuk mengatur waktu kerja, sehingga didapatkan jadwal yang seefisien mungkin. Sebuah penjadwalan akan tampak mudah jika komponen yang dijadwalkan dalam jumlah relatif sedikit, namun akan menjadi sebuah masalah yang cukup rumit jika komponen penyusunnya dalam jumlah yang besar.

Pembuatan jadwal kuliah akan selalu muncul karena harus dilakukan setiap pergantian semester. Umumnya jadwal ujian dapat diselesaikan dengan membuat tabel jadwal secara manual. Cara ini membutuhkan waktu yang cukup lama, karena pembuatan jadwal tersebut sangatlah kompleks yang terdiri dari beberapa komponen penyusun seperti mata kuliah, dosen, ruangan dan waktu. Dosen juga dimasukkan kedalam komponen penyusun karena dosen diwajibkan hadir saat kuliah berlangsung. Pada setiap komponen penyusun tersebut banyak terdapat aturan dan batasan-batasan yang telah ditentukan, oleh karena itu diperlukan penjadwalan otomatis yang dapat membuat jadwal dengan cepat, mudah dan tetap memerhatikan aturan-aturan serta batasan yang sudah ditentukan.

Dalam makalah ini, penulis memandang bahwa batasan (*constraint*) merupakan hal terpenting yang harus dipenuhi dalam pembuatan jadwal kuliah. Untuk menyelesaikan masalah pemenuhan batasan dikenal istilah *Constraint Satisfaction Problem* (*CSP*). Terdapat beberapa algoritma yang dapat digunakan untuk menyelesaikan *CSP*, salah satu diantaranya adalah algoritma *backtracking*.

Algoritma *backtracking* adalah suatu algoritma yang berbasis pada *Depth First Search* (*DFS*) untuk mencari solusi persoalan yang lebih efisien. *DFS* akan melakukan pencarian solusi pada suatu node dalam setiap level mulai dari yang paling kiri. Untuk mendapatkan hasil yang optimal dari data

yang dimiliki banyak sekali *constraint* yang dalam makalah ini diselesaikan dengan *CSP*.

CSP bertujuan untuk memperoleh suatu kombinasi variabel-variabel tertentu yang memenuhi *constraint* yang diberikan sehingga dapat dihasilkan solusi yang baik.

II. STUDI LITERATUR

1. Constraint Satisfaction Problem (*CSP*)

CSP merupakan sebuah pendekatan dari masalah yang bersifat matematis dengan tujuan untuk menemukan keadaan atau objek yang memenuhi sejumlah persyaratan atau kriteria. Sebuah *constraint* diartikan sebagai batasan solusi yang memungkinkan dalam sebuah masalah optimasi.

Ada beberapa hal yang perlu diketahui dalam *CSP* yaitu :

- *Initial State* (*) : semua variabel belum di-assign atau *assignment kosong* {}.
- *Successor Function* : sebuah nilai di-assign kedalam sebuah *unassigned* variabel, namun nilai yang di-assign tidak boleh konflik dengan variabel yang sudah di-assign sebelumnya.
- *Goal Test* : semua variabel telah di-assign dan tidak konflik dengan variabel sebelumnya.
- *Path cost* : konstanta *cost* untuk setiap langkah *successor function*.

Dari penjelasan diatas, *unassigned* variabel adalah daftar variabel yang belum memiliki nilai sedangkan *assign* variabel adalah daftar variabel yang sudah memiliki nilai. Pada *CSP* terdapat istilah domain, yaitu daftar nilai yang memungkinkan untuk setiap variabel. Domain dapat berupa integer ataupun string.

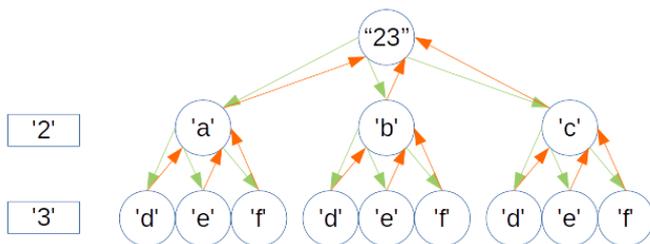
2. Algoritma *Backtracking*

Algoritma *Backtracking* pertama kali diperkenalkan oleh D.H Lehmer, ahli matematika Amerika pada tahun 1950. Algoritma

Backtracking merupakan algoritma yang berbasis DFS untuk mencari solusi persoalan dengan lebih efisien.

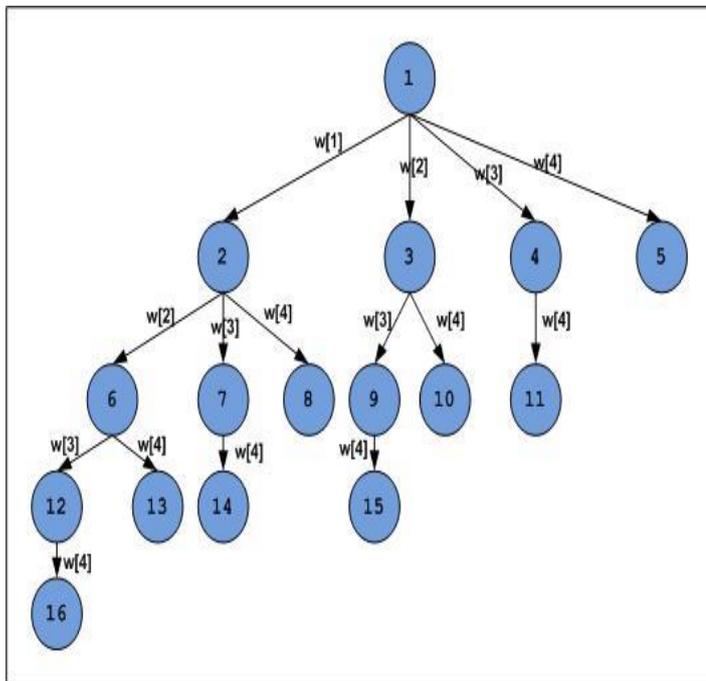
Algoritma *backtracking* melakukan pencarian solusi persoalan secara sistematis pada semua kemungkinan solusi yang ada pada setiap node dengan berbasis pada DFS rekursif.

DFS merupakan suatu metode pencarian yang dilakukan pada satu node dalam setiap level dari yang paling kiri. Jika solusi belum juga ditemukan, maka pencarian dilanjutkan pada node sebelah kanan. Demikian seterusnya sampai ditemukan solusi atau jika menemukan jalan buntu akan *backtrack* ke posisi sebelumnya. Jika solusi telah ditemukan maka pencarian akan berhenti meskipun ada node yang belum ditelusuri.



Proses *backtracking* : Sumber :
<https://rekinyz.files.wordpress.com/2015/02/backtracking2.png>

Akses : Selasa, 16 Mei 2017 pukul 19:31 WIB



III. ANALISIS STUDI KASUS

1. Penjadwalan Kuliah

- Waktu kuliah : waktu perkuliahan adalah hari senin sampai jumat, setiap hari

dibagi menjadi 8 sesi. Setiap sesi kuliah berdurasi 50 menit. Kuliah teori (tatap muka) dialokasikan selama 1 sesi dan praktikum 2 sesi berturut-turut. Namun untuk beberapa mata kuliah, sesi teori dapat dialokasikan 2 sesi sekaligus, sedangkan awal jam perkuliahan dan akhir jam perkuliahan sama untuk setiap harinya kecuali untuk hari jumat. Waktu perkuliahan terbagi menjadi 3 bagian setiap harinya, yaitu:

- ✓ Pagi hari, jam kuliah dimulai pada pukul 08:00 WIB dan berakhir pukul 11:50 WIB
- ✓ Siang hari sampai sore hari dimulai pada pukul 13:00 WIB dan berakhir pukul 16:50 WIB
- ✓ Khusus untuk hari jumat, jam kuliah dimulai pada pukul 08:15 WIB dan berakhir pukul 17:20, sedangkan pada pukul 11:00-12:00 digunakan untuk ibadah shalat jumat.

• Kurikulum

Kurikulum berisi informasi tentang mata kuliah setiap semester, jumlah Sistem Kredit Semester (SKS), dan sistem pengambilan kuliah dengan sistem paket, sehingga mahasiswa harus mengambil seluruh mata kuliah yang telah ditentukan setiap semesternya. Untuk penyusunan jadwal kuliah secara manual, selain jumlah mata kuliah yang terdapat untuk setiap semesternya, perlu diperhatikan juga jumlah SKS dari setiap mata kuliah. Pembagian SKS mata kuliah antara sesi teori dan praktikum untuk setiap minggunya adalah sebagai berikut :

- ✓ Matakuliah 2 SKS, satu jam teori dan dua jam praktikum atau dua jam teori sekaligus
- ✓ Mata kuliah 3 SKS, dua jam teori dan dua jam praktikum atau satu jam teori dan 4 jam praktikum
- ✓ Mata kuliah 4 sks
- ✓ Tugas akhir 4 sks
 - Jumlah Kelas

Pendataan jumlah kelas untuk setiap mata kuliah sangat diperlukan agar tidak terjadi bentrok antara dua kelas yang berbeda terhadap mata kuliah yang sama pada waktu bersamaan. Misalnya, mata kuliah OOP dibuka untuk dua kelas, yaitu kelas A dan kelas B untuk sesi 2, maka mahasiswa yang ada di sesi 1 tidak bisa

mengikuti perkuliahan yang ada di sesi 2 karena akan bertabrakan dengan mata kuliah yang lain

- Pengajar

Setiap pengajar atau dosen bisa mengajar lebih dari satu matakuliah, namun waktu pengajarannya tidak boleh bertabrakan

- Ruangan

Terdapat dua jenis ruangan yaitu ruangan kelas untuk sesi teori dan laboratorium untuk sesi praktikum

- Pengaturan kelas

Langkah selanjutnya dalam pengaturan jadwal kuliah adalah mengatur ruangan kelas atau matakuliah per-semester, mengatur dosen yang mengajar, lama perkuliahan, dan membuka kesempatan kepada pengajar mata kuliah tertentu untuk memilih jam yang diinginkan untuk pengajaran serta waktu dimana dosen tersebut tidak dapat mengajar.

- Pengaturan *constraint*

Ada beberapa batasan yang harus diperhatikan dan dipertimbangkan, jika batasan ini dilanggar akan menyebabkan kesalahan dalam penyusunan jadwal kuliah dan mengakibatkan adanya jadwal kuliah yang konflik. Batasan yang harus diperhatikan antara lain:

- ✓ Mata kuliah yang berbeda tidak diizinkan berjalan bersamaan untuk satu kelas dalam waktu yang sama
- ✓ Seorang dosen tidak diizinkan mengajar pada 2 atau lebih kelas kuliah pada waktu bersamaan
- ✓ Seorang dosen hanya mengajar satu mata kuliah dalam satu kelas dalam satu waktu, satu kelas hanya dijadwalkan untuk satu mata kuliah dalam satu waktu
- ✓ Seorang dosen dapat mengajarkan lebih dari satu matakuliah
- ✓ Seorang dosen tidak ingin mengajar pada satu hari tertentu
- ✓ Terdapat jam-jam tertentu dimana perkuliahan ditiadakan misalnya jika ada meeting, ibadah atau hal lainnya.

- ✓ Sesi teori dilaksanakan sebelum sesi praktikum untuk setiap mata kuliah

- ✓ Jumlah maksimal sesi yang digunakan untuk satu mata kuliah dalam satu hari yaitu tiga sesi, dimana satu sesi untuk kegiatan teori dan dua sesi digunakan sekaligus untuk kegiatan praktikum

- ✓ Pengaturan perbandingan antara sesi teori dan praktikum setiap minggu untuk setiap mata kuliah sesuai dengan jumlah SKS

- ✓ Sesi praktikum untuk mata kuliah yang tiga maupun empat SKS dilakukan berselang satu hari sebelum praktikum terakhir

- ✓ Sesi teori untuk setiap matakuliah dilaksanakan sebanyak satu sesi dalam satu hari kecuali untuk mata kuliah diluar bidang (jurusan) dapat dijadwalkan berturut-turut selama dua sesi dalam satu hari

- ✓ Dua kelas yang mengambil matakuliah yang sama dapat diletakan secara berurut

- Penyusunan dan Evaluasi Jadwal

Dengan memerhatikan hal-hal diatas dalam pembuatan jadwal kuliah secara manual di sebuah politeknik, maka dapat diambil kesimpulan sebagai berikut:

- Resource

- a) Terdapat tiga ruangan kelas yang digunakan untuk kegiatan teori yang ruangan Del 1, Del 2, dan Del 3.

- b) Terdapat tujuh kelas, yaitu kelas 7601,7602,7603,7604,7605,7606 dan 7607.

- c) Terdapat dua ruang laboratorium untuk tingkat satu yaitu LabDas 1 dan LabDas2, dua ruang laboratorium untuk tingkat 2 yaitu IRK dan LabPro dan tiga praktikum untuk tingkat tiga yaitu GAIB, BasDat dan RPL.

- d) 26 Dosen yang aktif mengajar

IV. ANALISIS PERMASALAHAN

Pada pembuatan jadwal kuliah ini, terdapat sejumlah batasan yang harus dipenuhi diman untuk menangani pemenuhan batasan digunakan CSP. Algoritma yang digunakan untuk menyelesaikan CSP ini adalah algoritma *backtracking*. Pada CSP, terdapat beberapa komponen yang ditentukan terlebih dahulu yaitu variabel, domain dan batasan. Pada pembuatan jadwal kuliah, penentuan nilai dari tiap komponen tersebut adalah:

1. Variabel

- Hari, yaitu senin- jumat
- Sesi yaitu 1 s.d 8
- Kelas

Maka variabel yang dibutuhkan untuk pembuatan jadwal kuliah adalah 280 variabel. Perhitungan tersebut diperoleh dari : jumlah hari (5) x jumlah sesi (8) x jumlah kelompok kelas (7)

2. Domain

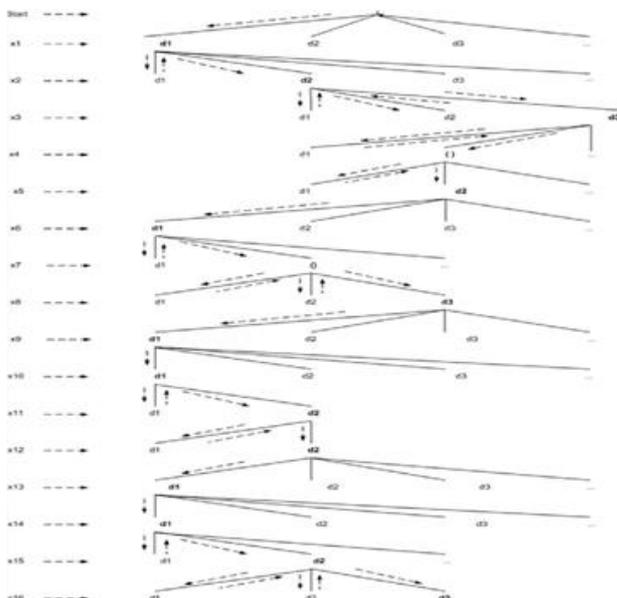
- Mata kuliah
- Ruang
- Jenis sesi (teori atau praktikum)
- Jumlah pertemuan dalam satu minggu untuk mata kuliah tertentu berdasarkan jenis sesi nya
- Lama waktu yang digunakan oleh dosen untuk mengajar

Berikut contoh pohon solusi yang dibentuk untuk menyelesaikan pembuatan jadwal kuliah. Adapun komponen-komponen yang digunakan pada contoh pohon solusi yaitu:

1. Variabel

Pada contoh ini terdapat 16 variabel yang digunakan yaitu:

- Hari, yaitu senin
- Sesi, yaitu sesi 1,2,3 dan sesi 4.



- Kelas berdasarkan kelompok pembagian kelas yaitu 7601,7602 dan 7604

Gambar 2 Alur algoritma runut balik pada penjadwalan kuliah

Bentuk variabel pada contoh ini adalah $x_n = (\text{hari, sesi, kelompok kelas})$

Keterangan :

- a) n adalah urutan variabel $n=1,2,3,\dots,n$
- b) m adalah urutan sesi, $m= 1,2,\dots,8$

berikut adalah variabel-variabel yang digunakan:

$x_1 = (\text{senin, sesi1, 7601}), x_2 = (\text{senin, sesi1, 7602}),$
 $x_3 = (\text{senin, sesi1, 7603}), x_4 = (\text{senin, sesi1, 7604}),$
 $x_5 = (\text{senin, sesi2, 7601}), x_6 = (\text{senin, sesi2, 7602}),$
 $x_7 = (\text{senin, sesi2, 7603}), x_8 = (\text{senin, sesi2, 7604}),$
 $x_9 = (\text{senin, sesi3, 7601}), x_{10} = (\text{senin, sesi3, 7602}),$
 $x_{11} = (\text{senin, sesi3, 7603}), x_{12} = (\text{senin, sesi3, 7604}),$
 $x_{13} = (\text{senin, sesi4, 7601}), x_{14} = (\text{senin, sesi4, 7602}),$
 $x_{15} = (\text{senin, sesi4, 7603}), x_{16} = (\text{senin, sesi4, 7604}).$

2. Domain

Domain yang digunakan pada contoh ini adalah:

- a) Mata kuliah
- b) Ruang
- c) Jenis sesi
- d) Jumlah pertemuan
- e) Lama mengajar

3. Batasan (*constraint*)

- a) Jumlah sesi dalam satu minggu untuk setiap mata kuliah
- b) Batasan yang lainnya yang sudah dijelaskan di bagian sebelumnya

Pada gambar 2 ditunjukkan contoh pohon solusi berdasarkan komponen-komponen pada penjadwalan kuliah.

V. HASIL

No Fungsi	Modul	Fungsi	I/O
B-01	Matakuliah	Penambahan data mata kuliah	T_pelajaran
B-02	Matakuliah	Pengubahan data matakuliah	T_pelajaran
B-03	Matakuliah	Penghapusan data matakuliah	T_pelajaran
B-04	Kelas	Penambahan data kelas	T_kelas

B-05	Kelas	Pengubahan data kelas	T_kelas
B-06	Kelas	Penghapusan data kelas	T_kelas
B-07	Dosen	Penambahan data dosen	T_dosen
B-08	Dosen	Pengubahan data dosen	T_dosen
B-09	Dosen	Penghapusan data dosen	T_dosen
B-10	Ruangan	Penambahan data ruangan	T_classroom
B-11	Ruangan	Pengubahan data ruangan	T_classroom
B-12	Ruangan	Penghapusan data ruangan	T_classroom
B-13	Batasan	Penambahan data batasan	T_relasi
B-14	Batasan	Pengubahan data batasan	T_relasi
B-15	Batasan	Penghapusan data batasan	T_relasi
B-16	Report	Menampilkan data mata kuliah	T_pelajaran
B-17	Report	Menampilkan data kelas	T_kelas
B-18	Report	Menampilkan data dosen	T_dosen
B-19	Report	Menampilkan data ruangan	T_classroom
B-20	Report	Menampilkan data batasan	T_relasi
B-21	-	Generate jadwal kuliah	T_pelajaran, T_kelas, T_dosen, T_classroom, T_relasi / jadwal kuliah

1. Seorang dosen tidak ingin mengajar pada hari tertentu
2. Sesi tertentu dimana perkuliahan ditiadakan
3. Dua kelas yang mengambil mata kuliah yang sama dapat diletakan secara berurut.

VI. Kesimpulan dan Saran

Program penjadwalan yang dibangun menghasilkan jadwal kuliah yang baik, dimana tidak terdapat jadwal yang konflik dan semua fungsi yang ada pada program dapat berjalan dengan baik.

Saran yang dapat diberikan dalam melakukan pengkajian selanjutnya, hendaknya memerhatikan hal-hal berikut :

1. Batasan yang belum bisa diaplikasikan pada aplikasi penjadwalan sebaiknya dapat diterapkan pada pengembangan selanjutnya
2. Sebaiknya jadwal kuliah yang telah terbentuk dapat disajikan kedalam format lain, seperti format pdf dan xls.
3. Pengembangan berikutnya sebaiknya path cost menjadi bahan pertimbangan dalam penyusunan jadwal kuliah misalnya path cost dari konflik yang terjadi antar kelas lebih tinggi daripada masukan dari dosen untuk tidak mengajar pada hari tertentu
4. Aplikasi penjadwalan hanya menggunakan satu basis data, jika hendak membuat jadwal baru, maka data yang ada sebelumnya diubah atau dihapus. Jadwal kuliah yang dibuat tidak disimpan. Sehingga untuk pengembangan selanjutnya, jadwal kuliah yang telah dibuat sebelumnya masih tersimpan dan dapat digunakan kembali, serta data lama pembuatan jadwal kuliah tidak terhapus.

Setelah seluruh informasi mengenai data dan batasannya dimasukkan, maka waktu yang dibutuhkan untuk menghasilkan jadwal kuliah adalah sekitar 0.6-1 detik, pada jadwal kuliah yang dihasilkan tidak terjadi konflik. Pada jadwal kuliah yang dihasilkan terdapat 280 variabel yang merupakan perkalian dari jumlah hari (5), jumlah sesi(8) dan jumlah kelompok kelas (7).

Adapun batasan yang belum bisa ditangani oleh program pengaturan jadwal ini adalah sbb:

UCAPAN TERIMA KASIH

Terimakasih saya ucapkan kepada dosen pengampu mata kuliah IF-2211 Strategi Algoritma yaitu bapak Dr. Ir. Rinaldi Munir MT, ibu Masayu Leyla Khodra ST.,MT dan ibu Dr. Nur Ulfa Maulidevi, ST.,M.Sc semoga materi yang telah diajarkan bisa saya amalkan sebaik mungkin serta semoga bapak dan ibu sekalian mendapat pahala yang berlipat ganda atas ilmu yang telah diajarkan kepada kami.

REFERENCES

- [1] <http://en.wikipedia.org/wiki/Scheduling>, "Backtracking", diakses 17 Maret 2009.
- [2] Russell, Stuart dan Peter Norvig, 2003, "Artificial Intelligence a Modern Approach", Pearson Education, New Jersey.
- [3] Munir, Rinaldi M.T, 2004, "Bahan Kuliah ke-9 IF2251 Strategi Algoritmik Algoritma Runut-balik (Backtracking)", Departemen Teknik Informatika ITB, Bandung.
- [4] Dechter, Rina dan Daniel Frost, "Backtracking Algorithm for Constraint Satisfaction Problem", Department of Information and Computer Science University of

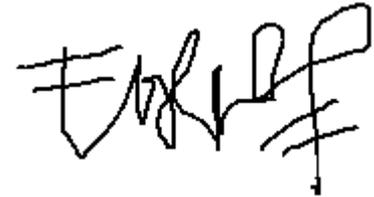
California, Irvine.

- [5] Riki Aditia, Prasodjo, Fachrul dan Ritonga, Irwansyah. 2006. "Pencarian Jalur Dengan Breadth First Search Dan Depth First Search", Sekolah Tinggi Teknologi Telkom, Bandung.

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, 16 Mei 2017



Farhan Makarim 13515003