

IMPLEMENTASI ALGORITMA *INTEGER LINEAR PROGRAMMING* UNTUK SISTEM INFORMASI PENJADWALAN RUANGAN DI FAKULTAS ILMU KOMPUTER UNIVERSITAS INDONESIA

A.R. Hutomo.¹, A. Fitrananda¹, A. Marshadiany.¹, G.P. Prikarti¹, dan E.M. Imah²

¹Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Indonesia, Kampus UI Depok, Jawa Barat, 16424, Indonesia

²Departemen Matematika, FMIPA, Universitas Negeri Surabaya, Kampus Ketintang, Surabaya, 60231

E-mail: ian.hutomo@gmail.com

Abstrak

Permasalahan konflik penjadwalan ruangan (*timetabling*) sering dihadapi hampir sebagian besar institusi akademis di Indonesia, salah satunya di Fakultas Ilmu Komputer Universitas Indonesia (Fasilkom UI). Peningkatan jumlah mahasiswa setiap tahun yang tidak diikuti oleh peningkatan jumlah dan kapasitas kelas menjadi faktor penyebab utama. Selama ini sistem penjadwalan masih dilakukan secara manual, sehingga membutuhkan waktu yang relatif lama dan menyebabkan optimasi pengalokasian kebutuhan ruangan menjadi kurang efisien. Penelitian ini bertujuan untuk menemukan pendekatan yang sesuai dalam menyelesaikan masalah *timetabling* tersebut. Beberapa pendekatan yang dapat digunakan untuk menyelesaikan masalah ini antara lain algoritma *Tabu Search*, *Simulated Annealing*, *Graph Coloring*, dan *Integer Linear Programming* (ILP). Dalam penelitian ini, peneliti menggunakan algoritma ILP karena ILP merupakan model yang paling tepat untuk menyelesaikan masalah *timetabling* di Fasilkom UI. Algoritma ini dapat meminimalkan waktu yang diperlukan untuk melakukan penjadwalan dari sebulan menjadi hitungan menit.

Kata Kunci: *integer linear programming, permasalahan penjadwalan kelas, timetabling*

Abstract

Room scheduling conflict issues (*timetabling*) are facing most of the academic institutions in Indonesia, one is in the Faculty of Computer Science (Fasilkom) Universitas Indonesia (UI). In the number of students each year followed by no increase in the number and capacity of the class became the main factor. During this scheduling system is still done manually so it takes a relatively long time so that the optimization is less efficient allocation of space requirements. This study aims to find an appropriate approach in solving the *timetabling* problem. Several approaches can be used to solve these problems include *Tabu Search* algorithm, *Simulated Annealing*, *Graph Coloring*, and *Integer Linear Programming* (ILP). In this study we used the ILP algorithm for ILP is the most appropriate model to solve the *timetabling* problem in Fasilkom UI. This algorithm can minimize the time required to perform the scheduling of a month becomes a matter of minutes.

Keywords: *integer linear programming, timetabling, university classroom scheduling problem*

1. Pendahuluan

Ruangan merupakan infrastruktur yang memegang peranan krusial dalam menunjang keberlangsungan proses belajar mengajar di institusi pendidikan. Masalah tersedianya ruangan biasa dikenal dengan *University Class Scheduling Problem* (UCSP) / *timetabling problem*, praktis berdampak pada terhambatnya kegiatan institusi, terutama kegiatan akademis. UCSP muncul sebagai respons atas kurang optimalnya pemetaan antara sumber daya yang tersedia terhadap batasan yang ada sehingga dibutuhkan suatu *system* penjadwalan yang baik.

Fakultas Ilmu Komputer Universitas Indonesia (Fasilkom UI) memiliki sejumlah ruangan untuk menunjang kegiatan setiap *civitas* akademisi, baik ruang kelas, aula, maupun laboratorium. Pengelolaan ruangan selama ini, telah dikoordinasikan dengan baik oleh sekretariat fakultas, namun dilakukan secara manual. Seiring dengan peningkatan jumlah mahasiswa dan staf dari tahun ke tahun, kebutuhan akan ruangan pun semakin meningkat. Peningkatan kebutuhan ruangan ini kemudian berimbas pada meningkatnya kompleksitas dalam pengalokasian ruangan sehingga metode manual tidak cukup optimal dalam mengakomodasi kebutuhan tersebut.

Beberapa peneliti telah mencoba memodelkan UCSP dengan beberapa algoritma penjadwalan yaitu, *Genetic Algorithm* (GA), *simulated annealing*, dan algoritma genetika *hybrid* (*Tabu Search*) [1]. Ketiga algoritma tersebut menggunakan pendekatan *heuristik* yang berdasarkan pada teknik pencarian lokal sehingga tidak menjamin akan menghasilkan solusi yang optimal. Algoritma tersebut kurang tepat untuk permasalahan UCSP di Fasilkom UI, karena solusi yang dibutuhkan bersifat optimum global. Salah satu algoritma yang dapat diterapkan untuk memecahkan masalah ini adalah *Integer Linear Programming* (ILP), yaitu suatu pendekatan *kombinatorial* yang dapat melakukan analisis pada seluruh *search space* yang terdefinisi dalam ruang lingkup masalah. Algoritma ILP dapat mengolah data dalam jumlah besar dan *constraint* dalam jumlah banyak.

Dalam tulisan ilmiah ini akan dijabarkan langkah-langkah dan rumus yang telah ditetapkan dalam metode ILP. Metode ILP ini diharapkan dapat diterapkan di Fasilkom UI dan menjadi solusi yang terbaik untuk memecahkan masalah *conflict scheduling*. Secara spesifik tujuan dari penelitian ini antara lain: melakukan pembuktian ilmiah non-eksperimental dengan melakukan analisis kualitatif berdasarkan langkah-langkah dan pembahasan rumus yang terdapat di dalamnya untuk memaparkan penggunaan metode ILP dalam memecahkan masalah *class scheduling*; memberikan alternatif solusi yang tidak hanya memecahkan masalah konflik *class scheduling*, tetapi juga yang efektif, efisien, dan dapat menangani *constraint* dalam jumlah besar; serta dapat dilakukan tindak lanjut untuk menerapkan metode ILP secara riil dalam sistem *class scheduling* Fasilkom UI.

UCSP dalam penelitian ini terbatas pada permasalahan konflik *class scheduling* di Fasilkom UI, domain permasalahan memiliki *constraint* dan parameter yang sudah dipastikan, bukan yang bersifat mendadak seperti kelas pengganti atau yang sebelumnya belum terdapat pada jadwal. Pengangkatan masalah tersebut dimaksudkan bahwa penetapan jadwal rutin pada awal periode akademik yang tidak efektif dapat berakibat “domino” kepada kebutuhan perkuliahan yang mendadak dengan sulitnya mendapatkan ruangan. Dengan menyelesaikan masalah konflik jadwal dan membuat penjadwalan yang efektif menggunakan ILP diharapkan saat masa yang akan datang terdapat kebutuhan mendadak akan mudah untuk mendapatkan kebutuhan ruangan karena jadwal telah dibuat efektif, sehingga solusi dengan metode ILP akan secara tidak langsung memecahkan masalah tersebut.

Berdasarkan tipe metodologi yang akan dilakukan, penelitian tidak akan dilakukan secara eksperimental dengan membuat sistem secara langsung dan diuji coba dalam waktu dekat, melainkan dengan menggunakan gabungan teori dan penjabaran matematis dengan rumus dan langkah-langkah serta dengan melakukan refleksi dan analisis secara langsung terhadap kondisi yang ada di Fasilkom UI saat ini. Namun untuk kelanjutannya, dapat dilakukan tindak lanjut untuk menerapkan metode ILP secara riil dalam sistem *class scheduling* Fasilkom UI.

Sebelum membahas tentang ILP, terlebih dahulu peneliti menjelaskan tentang beberapa istilah terkait masalah optimasi penjadwalan ruangan dan beberapa metode yang pernah digunakan dalam mengatasi masalah serupa.

University classroom scheduling problem (UCSP) merupakan masalah mengenai penjadwalan ruangan kelas di institusi pendidikan. UCSP timbul akibat keterbatasan ruangan yang dihadapkan dengan banyaknya mata kuliah yang harus diadakan dan jumlah murid yang tidak seimbang. Penjadwalan adalah alokasi sumber daya dari waktu ke waktu untuk melakukan sejumlah tugas. Sumber daya yang dimaksud adalah dosen, ruangan, mata kuliah, dan mahasiswa. Tugas didefinisikan sebagai kegiatan-kegiatan universitas. Masalah mendasar dari penjadwalan ruang kelas adalah menetapkan kegiatan-kegiatan universitas seperti perkuliahan terhadap berbagai sumber daya terbatas [2].

Penjadwalan kegiatan terhadap sumber daya tersebut harus dilakukan dengan meminimalkan jumlah pelanggaran terhadap *constraint* tertentu. Secara sederhana, *constraint* ini dapat dimisalkan dengan tidak adanya dosen, mata kuliah, atau ruangan yang digunakan pada lebih dari satu jadwal pada setiap periode. *Constraint* ini kemudian digolongkan menjadi *hard constraint* yaitu *constraint* yang wajib dipenuhi, *soft constraint* yaitu *constraint* yang sebaiknya dipenuhi [3]. Bentuk *hard constraint* adalah tidak ada orang yang berada pada dua tempat atau waktu secara bersamaan, atau total jadwal mata kuliah harus kurang atau sama dengan jumlah ruangan yang dimiliki. Sebagai contoh, bentuk *soft constraint* adalah: seorang dosen sebaiknya tidak memiliki jadwal kuliah malam setiap hari, atau seorang mahasiswa sebaiknya tidak memiliki lebih dari empat kelas dalam satu hari. Seiring dengan meningkatnya jumlah mahasiswa di universitas, maka jumlah kelas yang ditawarkan juga menjadi hal yang penting untuk dipertimbangkan. Terlebih, setiap mata kuliah akan memiliki jumlah peserta yang berbeda-beda, begitu pula kapasitas dari ruang kelas.

Salah satu permasalahan dari UCSP adalah bagaimana melakukan optimisasi dan utilisasi ruangan secara efektif dan efisien [4]. Dari permasalahan ini, termasuk didalamnya adalah masalah penjadwalan ujian, penggunaan ruangan untuk seminar dan acara lainnya, penjadwalan *shift* karyawan, dan sebagainya. Permasalahan ini juga merupakan *non-polynomial-time hard (NP-hard) problem*, yang berarti jumlah perhitungan yang dibutuhkan untuk mencari solusi optimal akan meningkat secara eksponensial seiring dengan meningkatnya ukuran masalah [5].

Constraint Satisfaction Problem (CSP) merupakan sebuah permasalahan umum dengan fitur-fitur unik yang dapat dieksploitasi untuk menemukan sebuah solusi. Permasalahan tersebut dimodelkan dengan variabel-variabel terbatas di mana setiap variabel terhubung dengan domain terbatas dan satu *set constraint* yang membatasi nilai-nilai yang bisa dipilih oleh variabel [6]. Tugasnya berfokus pada bagaimana memberikan nilai yang sesuai kepada setiap variabel untuk memenuhi seluruh *constraint* yang ada. CSP seringkali diasosiasikan dengan teori graf seperti *graph colouring* dan *scene labelling*.

Pada *Tabu Search Algorithm*, pemecahan masalah dilakukan dengan mencoba berbagai macam kemungkinan yang memenuhi kriteria penilaian dari *constraint* yang ada [7]. Suatu nilai yang tidak memenuhi kriteria solusi akan dimasukkan dalam status *tabu*, yang berarti nilai tersebut tidak dipertimbangkan sebagai solusi yang optimal. Kemungkinan-kemungkinan solusi tersebut diukur dengan menerapkan serangkaian *if* dan *else*. Keunikan dari *Tabu Search Algorithm* ialah apabila suatu solusi dianggap tidak memuaskan, maka akan dicari solusi yang lainnya di dalam wilayah tetangganya. Apabila sudah terdapat solusi yang dianggap cukup optimal, maka akan tetap dilakukan pencarian karena ada kemungkinan *system* mendapatkan solusi yang lebih optimal. Solusi optimal yang sebelumnya dimasukkan ke dalam *explicit memory* untuk menampung sementara yang selanjutnya dapat digantikan nilainya dengan solusi yang lebih optimal [8].

Seperti halnya algoritma dalam pemecahan masalah UCSP, *Simulated Annealing Algorithm* juga memecahkan masalah *soft constraint* dan *hard constraint*. Langkah awal pemecahan setelah menentukan *soft constraint* dan *hard constraint* hampir sama dengan *Tabu Search*, yaitu dengan menerapkan metode iteratif untuk mencoba kemungkinan dari *neighborhood*-nya dengan metode *simple searching*, *swapping*, dan kombinasi keduanya [9]. Perbedaananya terletak pada proses selanjutnya yaitu *cost calculation*. *Cost calculation* merupakan karakteristik dari

algoritma ini dan bertujuan untuk mencari pengaruh dan fungsi dari kedua *soft constraint* dan *hard constraint*. Langkah terakhir dari *Simulated Annealing Algorithm* adalah *cooling schedule*. *Cooling schedule* bertujuan untuk memberikan solusi yang terbaik berdasarkan *cost calculation* dengan waktu eksekusi yang dapat diterima.

Graph Coloring Algorithm. Pada *graph coloring algorithm* untuk memecahkan CSP memiliki langkah awal yang sama, yaitu dengan menentukan *constraint-constraint* yang diinginkan. Setelah itu *constraint* pada setiap mata kuliah yang akan dijadwalkan (*course*) direpresentasikan dalam *vertices* dan *edges*. *Vertices* merepresentasikan *courses* dan *edge* merepresentasikan setiap *course* yang masih bermasalah, atau konflik. Setiap *edge* yang ada menghubungkan *constraint* yang ada pada setiap *course*-nya, misalnya *constraint* dua *courses* untuk tidak dijadwalkan pada waktu yang bersamaan.

Pada *graph coloring* juga dikenal *greedy graph coloring*. Pada *greedy graph coloring* dilakukan analisis pada setiap *vertices*-nya apakah sudah sesuai dengan *constraint* yang ada. Dengan mendaftarkan pada warna yang sudah ada pada *vertices* yang sudah sesuai dengan *constraint* yang ada. Dengan demikian *greedy graph coloring* memiliki konsep untuk menggunakan warna seminimum mungkin.

Integer Linear Programming (ILP) merupakan sebuah metode matematis untuk memaksimalkan *profit* dan meminimalisasi *cost* berdasarkan sebuah model matematika yang melibatkan variabel-variabel yang bertipe *integer* yang direpresentasikan dalam suatu bentuk hubungan yang bersifat linier [10-12]. Penyelesaian persamaan matematis biasanya menghasilkan keluaran berupa bilangan pecahan, jika sistem mengharapkan keluaran bilangan *integer*, maka ILP merupakan algoritma paling tepat karena jika menggunakan pembulatan seringkali hasil yang diperoleh bukan solusi optimal. Berdasarkan jenis dan jumlah *integer*-nya, ILP dapat dibedakan menjadi tiga buah kategori. Pertama, *pure ILP* atau ILP murni, di mana keseluruhan variabel yang bermain di dalam model matematika tersebut berupa *integer* [13]. Kedua, *mixed ILP (MILP)*, di mana hanya sebagian saja yang berupa *integer*. Kategori terakhir yaitu *zero-one ILP*, di mana variabel-variabel *integer* yang ada bersifat *binary*, yang mana hanya dapat menerima nilai satu (*yes*) atau nol (*no*).

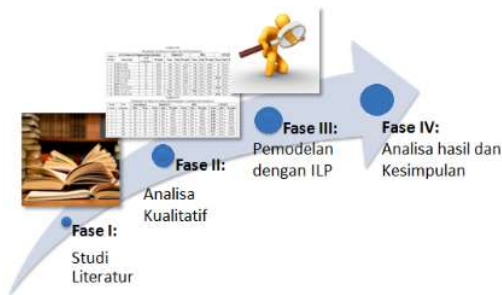
ILP banyak digunakan untuk menentukan kondisi maksimum maupun minimum dari sebuah model matematis yang melibatkan banyak

variabel-variabel yang menjadi syarat dan batasan dalam menghitung kondisi maksimum maupun minimum tersebut [14].

Penghitungan ILP sendiri dimulai dengan menentukan asumsi, *decision variable*, *problem constraint*, dan fungsi objektif sebelum akhirnya ditemukan solusi optimalnya [15]. Mengenai penghitungan ILP ini akan dijelaskan secara mendalam pada metodologi penelitian.

2. Metodologi

Penelitian ini terdiri atas empat fase: fase pertama mencakup studi literatur, fase kedua mencakup analisis kualitatif, fase ketiga pemodelan dengan metode ILP, dan fase keempat mencakup analisis untuk penarikan kesimpulan. Metodologi penelitian ini terangkum pada gambar 1.



Gambar 1. Tahapan metodologi penelitian.

Fase awal penelitian dilakukan studi literatur. Pada fase ini dilakukan tinjauan pustaka berupa buku, jurnal ilmiah internasional, maupun hasil konferensi ilmiah internasional untuk memperoleh informasi lebih dalam dan lebih rinci terkait masalah pada *class scheduling*. Dilakukan pula analisis artikel, berita terbaru untuk mengetahui update dan informasi tambahan mengenai penggunaan metode *class scheduling* dari berbagai algoritma dan penerapannya di berbagai institusi pendidikan untuk melihat adanya keterkaitan antara metode ILP dan metode lainnya.

Setelah dilakukan studi literatur, fase selanjutnya dilakukan analisis kualitatif dari keempat algoritma tersebut tertuang dalam tabel I. Dari keempat algoritma yang disebutkan dalam tabel I. Algoritma ILP merupakan algoritma yang paling baik untuk diterapkan di Fasilkom UI, dengan pertimbangan sebagai berikut: memiliki kelebihan untuk menangani *constraint* dalam jumlah yang besar, seperti yang dimiliki oleh Fasilkom UI. Selain itu Fasilkom UI juga memiliki kemungkinan besar untuk bertambahnya parameter-parameter yang ada setiap tahunnya,

baik parameter jumlah mahasiswa, mata kuliah, dosen, maupun ruangan; menutupi kekurangan yang dimiliki oleh algoritma lain dengan kekurangan dari ILP yang masih dapat diterima dan di-handle; berdasarkan studi kasus dari berbagai universitas, ILP merupakan metode yang terbaik dan terbukti memecahkan masalah UCSP di universitas-universitas dunia yang memiliki masalah UCSP yang serupa dengan Fasilkom UI.

Fase ketiga dari penelitian ini yaitu pemodelan dengan metode ILP. Dalam fase ini dilakukan penjabaran perhitungan dengan pendekatan rumus dan langkah-langkah dari metode ILP. Langkah-langkah yang dilakukan adalah dengan membuat pemodelan masalah secara matematis pada lima langkah awal yang dilanjutkan dengan implementasi atau perhitungan sebagai langkah terakhir dari metode ILP meliputi 6 hal. Pertama, menentukan asumsi. Awal dari pemodelan ini dimulai dengan menentukan asumsi yang berlaku dalam hal *class scheduling*. Asumsi yang diidentifikasi merupakan pra kondisi yang telah dilakukan sebelum melakukan perhitungan atau penerapan algoritma. Kedua, menentukan *decision variable*. Pada tahap ini ditentukan *decision variable* yang berlaku secara matematis. Seluruh jenis *decision variable* akan diidentifikasikan dalam bentuk 0-1.

Ketiga, menentukan model parameter. Pada tahap ini mendaftarkan semua parameter untuk memodelkan *timetabling problem*. Keempat, menentukan *problem constraint*. Pada tahap ini ditetapkan batasan-batasan masalah pada *class scheduling*. Terdapat dua jenis *problem constraint* yang harus ditentukan: *hard constraint* dan *soft constraint* [7]. Pemodelan matematis kemudian dilakukan untuk kedua *constraint* tersebut. Kelima, menentukan fungsi objektif. Dalam tahap ini ditentukan fungsi objektif dalam bentuk matematis dengan menggabungkan semua *constraint* yang ada. Fungsi objektif yang ingin dicapai dalam penelitian ini adalah meminimumkan semua *constraint*. Keenam, mengimplementasikan penghitungan. Setelah seluruh tahap dan komponen telah ditentukan dan dimodelkan dengan matematis, maka selanjutnya dilakukan perhitungan untuk mendapatkan jadwal yang diinginkan dan optimal.

Fase terakhir dilakukan analisis hasil dan penarikan kesimpulan. Pada fase ini telah dapat dilihat hasil dari penjabaran dan implementasi perhitungan rumus dengan menggunakan ILP sehingga dapat dibandingkan dengan melakukan *class scheduling* secara manual. Dari tahap ini, dilakukan analisis perbandingan dan penarikan kesimpulan mengenai pengaruh serta keunggulan dari penggunaan metode ILP untuk menyelesaikan

masalah *class scheduling* beserta faktor-faktor pendukung dari segi organisasi.

3. Hasil dan Pembahasan

Langkah pertama dari pemodelan dan analisis data dengan *integer linear programming* dilakukan dengan menentukan asumsi. Asumsi-asumsi yang digunakan untuk penelitian ini diantaranya, masa registrasi mata kuliah pada SIAK sudah selesai namun belum mencakup masa *add and drop*; mata kuliah yang akan dibuka telah ditentukan; jumlah maksimum mahasiswa untuk setiap mata kuliah telah ditentukan; serta dosen untuk setiap mata kuliah telah ditentukan.

Langkah kedua dilakukan dengan menentukan *decision variable*. Variabel-variabel yang digunakan dalam penelitian ini didefinisikan sebagai berikut:

Variabel $i = \{1, 2, 3, \dots, Q\}$ di mana i merupakan indeks dari periode mata kuliah yang dilaksanakan.

Variabel $j = \{1, 2, 3, \dots, N\}$ di mana j merupakan indeks mata kuliah yang terdapat pada periode akademik yang dipetakan dengan N merepresentasikan total mata kuliah yang akan dimasukkan dalam jadwal.

Variabel $k = \{1, 2, 3, \dots, M\}$ di mana k merupakan indeks dari ruangan kelas yang terdapat pada Fasilkom UI dan yang dapat maupun akan digunakan sebagai ruang kelas dengan M merepresentasikan total keseluruhan ruangan kelas yang dapat digunakan untuk kegiatan belajar mengajar.

x_{ijk} bernilai 1 jika mata kuliah j dijadwalkan pada periode i di slot waktu k . x_{ijk} bernilai 0 jika sebaliknya.

Langkah ketiga yaitu menentukan model parameter. Setelah menentukan variabel keputusan, seluruh parameter kemudian didaftarkan sebagai berikut:

a : himpunan kelompok mahasiswa
 B_k : kapasitas untuk ruangan k
 C : himpunan mata kuliah
 C_l : himpunan mata kuliah yang diajar oleh dosen l
 $C.A_h$: himpunan mata kuliah yang diambil oleh kelompok mahasiswa h
 D_j : durasi untuk mata kuliah j
 E_j : ruangan untuk mata kuliah j
 I : himpunan slot mata kuliah $\{I_1, \dots, I_{55}\}$
 I_{senin} : $\{1, 2, 3, \dots, 11\}$
 I_{selasa} : $\{12, 13, \dots, 22\}$
 I_{rabu} : $\{23, 24, \dots, 33\}$ - $\{I_r\}$
 I_{kamis} : $\{34, 35, \dots, 44\}$
 I_{jumat} : $\{45, 46, \dots, 55\}$ - $\{I_j\}$
 I_r : $\{28\}$
 I_j : $\{50, 51\}$

I_l : $\{5, 16, 27, 38, 49\}$
 M : jumlah ruangan kelas = 20
 N : jumlah mata kuliah = 177
 Q : jumlah slot mata kuliah = 52
 T : himpunan dosen, $|T| = 80$
 P_j : bobot untuk mata kuliah j
 R : jumlah kelompok mahasiswa
 q : jumlah dosen yang mengajar

Berdasarkan data jadwal mata kuliah semester genap 2010/2011 yang telah diperoleh dari sekretariat, maka elemen-elemen dari himpunan-himpunan di atas dapat didaftarkan ke dalam tabel II, tabel III, dan tabel IV [16].

Langkah keempat, menentukan *problem constraint*. *Problem constraint* dibagi menjadi dua kategori utama *hard constraint* dan *soft constraint*. *Hard constraint* merupakan batasan wajib yang mutlak untuk dipenuhi dan menjadi peraturan atau standar. *Hard constraint* pada penelitian ini mencakup:

Suatu kelompok mahasiswa a tidak dapat mengikuti mata kuliah pada periode waktu i yang bersamaan, maka persamaan 1:

$$\sum_{j \in C.A_h} \sum_{k=1} x_{ijk} \leq 1, \quad \forall i \in Q \text{ dan } \forall h \in p \quad (1)$$

Seorang dosen t tidak dapat mengajar lebih dari satu mata kuliah pada periode waktu i yang sama, maka persamaan 2:

$$\sum_{j \in C_t} \sum_{k=1}^M x_{ijk} \leq 1, \quad \forall i \in Q, \forall t \in q, \text{ dan } \forall a \in p \quad (2)$$

Sebuah ruangan tidak dapat digunakan untuk lebih dari satu mata kuliah pada suatu periode waktu i yang sama, maka persamaan 3:

$$\sum_{j=1}^N x_{ijk} \leq 1, \quad \forall i \in Q, \forall k \in M \quad (3)$$

Setiap mata kuliah harus memiliki total alokasi waktu sesuai dengan bobot SKS, maka persamaan 4:

$$\sum_{i=1}^Q \sum_{k=1}^M x_{ijk} \leq \frac{D_j}{N}, \quad \forall j \in N \quad (4)$$

Tidak ada ruangan yang bisa digunakan melebihi kapasitasnya;

Suatu mata kuliah tidak dijadwalkan lebih dari satu kali dalam sehari maka persamaan 5:

$$\sum_{t=1-D_{j+1}}^i x_{tjk} = D_j, \quad \forall j \in N, \forall k \in M, \forall i \in Q \quad (5)$$

TABEL I
ANALISIS KUALITATIF ALGORITMA

	Kelebihan	Kelemahan
Tabu Search	- Efisien dengan menggunakan analisis <i>neighborhood</i>	- Membutuhkan waktu yang relatif lama karena menggunakan metode induktif - Kurang <i>satisfiable</i> dengan <i>constraint</i> yang besar - Membutuhkan proses yang cukup panjang dengan metode <i>if-else</i>
Simulated Annealing	- Efisien dengan menggunakan analisis <i>neighbor-hood</i> (hampir sama seperti <i>tabu search</i>)	- Belum menjamin solusi yang optimal meskipun dengan <i>cost</i> yang optimal, ataupun - Kalkulasi <i>cost</i> belum tentu optimal meskipun solusinya ditemukan [4]
Graph Coloring	- Memiliki penggunaan memori yang minimum dengan menggunakan <i>greedy graph</i> - Memiliki efisiensi yang tinggi	- Kurang dapat diandalkan dengan data dan <i>constraint</i> yang sangat besar
Integer Linear Programming	- Dapat digunakan dalam <i>constraint</i> dan data yang sangat besar - Dapat dimodifikasi sesuai dengan kebutuhan organisasi - Terbukti lebih unggul disbanding-kan <i>tabu search</i> dan <i>graph coloring</i> [4]	- Membutuhkan parameter yang <i>fix</i> dan tidak berubah

Sedangkan itu, *soft constraint* merupakan batasan yang tidak mutlak namun sangat disarankan dan diusahakan untuk dapat dipenuhi. *Soft constraint* dalam penelitian ini mencakup: Tidak ada jadwal kuliah pada waktu istirahat pukul 12.00 hingga 13.00, maka persamaan 6:

$$\sum_{j=i}^N \sum_{k=1}^M \sum_{1 \in I_l} x_{ijk} = f_{L_2}(x), \quad (6)$$

I_l = Himpunan waktu makan siang

Tidak ada jadwal kuliah pada hari Rabu pukul 13.00 hingga 14.00 karena dialokasikan untuk Seminar Reboan, sehingga persamaan 7:

$$\sum_{j=1}^N \sum_{k=1}^M \sum_{1 \in I_j} x_{ijk} = f_{R_3}(x), \quad (7)$$

I_r = Himpunan waktu seminar Reboan

TABEL II
HIMPUNAN KELOMPOK MAHASISWA

Notasi elemen	Nilai elemen
a_1	IK-2
a_2	IK-4
a_3	IK-6
a_4	IK-8
a_5	SI-2
a_6	SI-4
a_7	SI-6
a_8	SI-8
a_9	KI-2
a_{10}	KI-4
a_{11}	MIK
a_{12}	Eks-1
a_{13}	Eks-2
a_{14}	Eks-3
a_{15}	Eks-4
a_{16}	Eks-5

TABEL III
KAPASITAS RUANGAN

Notasi elemen	Nilai elemen (dalam satuan kursi)
B_1 (R. 1221)	40
B_2 (R. 2102/Aula)	126
B_3 (R. 2301)	60
B_4 (R. 2302)	70
B_5 (R. 2303)	70
B_6 (R. 2304/2305)	120
B_7 (R. 2306)	70
B_8 (R. 2307)	65
B_9 (R. 2401)	40
B_{10} (R. 2402/2403)	140
B_{11} (R. 2404)	80
B_{12} (R. 2405)	70
B_{13} (R. 2406)	70
B_{14} (R. 2601)	70
B_{15} (R. 2602)	180
B_{16} (R. 2603)	40
B_{17} (R. 3111)	34
B_{18} (R. 3112)	34
B_{19} (R. 3113)	34
B_{20} (R. 3224)	40

Tidak ada jadwal kuliah pada hari Jum'at pukul 11.00 hingga 13.00 karena dialokasikan untuk shalat Jum'at, maka persamaan 8:

$$\sum_{j=i}^N \sum_{k=1}^M \sum_{1 \in I_j} x_{ijk} = f_{J_1}(x), \quad (8)$$

I_j = Himpunan waktu Sholat Jum'at

Langkah kelima, menentukan fungsi objektif, dengan menggunakan rumus persamaan 9

$$\min f(x) = a_1 f_{J_1}(x) + a_2 f_{L_2}(x) + a_3 f_{R_3}(x) \quad (9)$$

Langkah terakhir yaitu dengan mengimplementasikan penghitungan. Berdasarkan semua *constraint* dan fungsi objektif yang telah didefinisikan di atas, kemudian penghitungan dilakukan dengan menggunakan *software* Lindo.

TABEL IV
HIMPUNAN DOSEN

Notasi elemen (T ke-...)	Nilai elemen
1	Aniati Murni
2	Abdul Muthalib
3	Achmad Nizar H.
4	Ade Azurat
5	Adhi Yuniarto
6	Agus Kurniawan
7	Alex
8	Ari Saptawijaya
9	Arif Budiman
10	Arli Setiawati
11	Barokah Widodo
12	Bayu Distiawan
13	Belawati Widjaja
14	Benny Ranti
15	Bob Hardian
16	Budi Yuwono
17	Chan Basaruddin
18	Clara Vania
19	Dadan Hardianto
20	Dana Indra
21	Dina Chahyati
22	Djauhari Sulichah
23	Dosen FE 1
24	Dosen FE 2
25	Dosen FE 3
26	Dosen Luar 1
27	Dosen Luar 10
28	Dosen Luar 2
29	Dosen Luar 3
30	Dosen Luar 4
31	Dosen Luar 5
32	Dosen Luar 6
33	Dosen Luar 7
34	Dosen Luar 8
35	Dosen Luar 9
36	Eko Budiardjo
37	Erdefi Rakun
38	Hendriawan
39	Heri Kurniawan
40	Heru Suhartanto
41	Iik Wilarso
42	Ika Alfina
43	Indra Budi
44	Ito Wasito
45	Ivan Fanany
46	Ivano Aviandi
47	Kasfu Hammi
48	Kasiyah Junus
49	Kurniawati Azizah
50	Manusun G S Aritonang
51	Metti Zakaria
52	Mirna Adriani
53	Petrus Mursanto
54	Puspa Indah
55	Putu Wuri Handayani
56	Rahmad Azhari
57	Rahmat Samik Ibrahim

Notasi elemen (T ke-...)	Nilai elemen
58	Rahmat Widyanto
59	Ricky Roland
60	Rifki Shihab
61	Riri Satria
62	Rizal Fathoni Aji
63	Ronald Simamora
64	Ruli Manurung
65	Samuel Louvan
66	Satrio Baskoro Y.
67	Setiadi Yazid
68	Siti Aminah
69	Tisha Melia
70	Toemin Masoem
71	Wahyu Catur Wibowo
72	Widijanto S. N
73	Wisnu Jatmiko
74	Wulung Pambuko
75	Yohanes Stefanus
76	Yova Ruldeviyani
77	Yudho Giri Sucahyo
78	Yugo Kartono Isal
79	Zaimal Hasibuan
80	Zaki Rahman

Setelah dilakukan penghitungan dengan menggunakan *software* Lindo diperoleh hasil bahwa permodelan ILP dapat mengefisiensikan waktu yang diperlukan dalam penyusunan jadwal ruangan hanya dalam 2 jam atau setara dengan efisiensi sebesar 97% bila dibandingkan dengan cara penjadwalan manual yang membutuhkan waktu hingga 3 hari untuk menyelesaikan draft pertama jadwal kuliah (97% diperoleh dari $(70/72) \times 100\%$).

4. Kesimpulan

Kesimpulan yang dapat diperoleh dari penelitian ini yaitu algoritma ILP merupakan pendekatan terbaik untuk memodelkan *timetabling problem* bila dibandingkan dengan algoritma *Tabu Search*, *Simulated Annealing* dan *Graph Coloring*. Selain itu, permodelan dengan menggunakan algoritma ILP pada masalah *timetabling* di Fasilkom UI memberikan efisiensi sebesar 97% bila dibandingkan dengan sistem penjadwalan secara manual.

Referensi

- [1] J.W. Herrmann & C.Y. Lee, "Solving A Class Scheduling Problem with a Genetic Algorithm," *ORSA Journal on Computing*, vol. 7, pp 443-452, 1995.
- [2] A. Wasfy & F.A. Aloul, "Solving the university class scheduling problem using advanced ILP techniques" *presented at the 4th IEEE GCC Conference*, Bahrain, 2007.
- [3] E. Tsang, "A Glimpse of Constraint Satisfaction," *Journal Artificial Intelligence*

- Review Kluwer Academic Publishers, vol. 13, pp. 215-227, 1999.
- [4] A.A. Salman' & S.A. Hamdan, "Solving Department's Course-scheduling Problem using Differential Evolution" *International Conference on Methods and Models in Computer Science*, 2009.
- [5] M.M. Aldasht, M.H. Saheb, I. Najjar, M.H. Tamimi, & T.O. Takturi, "University Course Scheduling Using Parallel Multi-Objective Evolutionary Algorithms," *Journal of Theoretical and Applied Information Technology*, vol.22, pp. 129-136, 2010.
- [6] D. Mairiza & D. T. Soraya, "Pendekatan Constraint Satisfaction dalam Perkuliahan di Fakultas Ilmu Komputer Universitas Indonesia," *Jurnal Ilmu Komputer dan Informasi*, vol. 1, pp. 1-7, 2008.
- [7] A. Gunawan, K.M. Ng, & K.L. Poh, "Solving the Teacher Assignment-Course Scheduling Problem by a Hybrid Algorithm," *World Academy of Science, Engineering and Technology*, vol. 33, pp. 259-164, 2007.
- [8] A.P. Punnen & Y.P. Aneja, "A Tabu Search Algorithm for The Resource-constrained Assignment Problem," *The Journal of Operational Research Society*, vol. 46, pp. 214-220, 1995.
- [9] C.L. Monna & C.N. Porrs, "On the Complexity of Scheduling With Batch Setup Times," *The Journal of Operational Research Society*, vol. 37, pp. 798-804, 1989.
- [10] C.W. Chang & J.G. Waight, "A mixed integer linear programming based hydro unit commitment" *In Power Engineering Society Summer Meeting, 1999, IEEE* , pp.924-928, 1999.
- [11] I. Kadayif, M. Kandemir, & U. Sezer, "An integer linear programming based approach for parallelizing applications in on-chip multiprocessors" *presented at 39th Design Automation Conference*, pp. 703-708, 2002.
- [12] A. Kobetski & M. Fabian, "Scheduling of discrete event systems using mixed integer linear programming" *In Discrete Event Systems, 2006 8th International Workshop*, pp.76-81, 2006.
- [13] F.R. Fogle & J.N. Lovett Jr., "An improved exploratory search technique for pure integer linear programming problems," *Systems, Man, and Cybernetics, 1991. "Decision Aiding for Complex Systems" IEEE International Conference*, vol.1, pp. 573-577, 1991.
- [14] S. Papadomanolakis & A. Ailamaki, "An Integer Linear Programming Approach to Database Design," *Data Engineering Workshop, IEEE 23rd International Conference*, pp. 442-449, 2007.
- [15] S. Ribić & S. Konjicija, "A two phase integer linear programming approach to solving the school timetable problem" *In Information Technology Interfaces (ITI), 2010 32nd International Conference*, pp. 651-656, 2010.
- [16] Final Jadwal Perkuliahan Program S1Semester Genap 2010/2011, Fakultas Ilmu Komputer Universitas Indonesia.