JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 9
No. 2
Agustus
2023

Algoritma Penanganan *Constraint* pada Persoalan Penjadwalan Perkuliahan Universitas di Lingkungan Pendidikan Tinggi Keagamaan Islam (PTKI)

Fatchurrochman[#1], Arif Nur Afandi[#2], M Zainal Arifin[#3], Wayan Firdaus Mahmudy[*4]

[#]*Program Studi Teknik Elektro dan Informatika, Universitas Negeri Malang*
Jalan Semarang No 5 Malang
[1]fatchurrochman.2205349@students.um.ac.id
[2]an.afandi@um.ac.id
[3]arifin.mzainal@um.ac.id

[*]*Fakultas Ilmu Komputer Universitas Brawijaya*
Jalan Veteran No 8 Malang
[4]wayanfm@ub.ac.id

***Abstrak*— Penjadwalan perkuliahan di universitas adalah kegitan rutin yang membutuhkan waktu relatif lama untuk menyelesaikannya jika dikerjakan secara manual. Waktu yang dibutuhkan akan semakin besar ketika semakin banyak constraint yang dipertimbangkan. Mekanisme penanganannya akan berbeda di tiap universitas karena mungkin mereka mempunyai constraint yang unik. Dalam paper ini dipaparkan berbagai algoritma penanganan untuk tiap jenis constraint dalam persoalan penjadwalan perkuliahan. Algoritma penjadwalan perkuliahan otomatis yang digunakan dalam penelitian ini adalah sequential search yang bekerja dengan cara mencari slot waktu yang masih belum dipergunakan untuk ditempati oleh kelas perkuliahan. Bila slot waktu telah dipergunakan maka sistem akan mencari slot waktu yang lain secara berurutan. Uji coba dilakukan di program studi Teknik Informatika UIN Malang pada semester Ganjil tahun akademik 2021/2022. Hasilnya menunjukkan bahwa dengan 10 constraint, sistem yang dibangun dapat menjadwalkan 190 kelas perkuliahan secara otomatis dan 21 kelas perkuliahan dijadwalkan secara interaktif. Dengan sistem yang diajukan maka seluruh kelas perkuliahan sebanyak 211 dapat terjadwal meskipun ada pelanggaran soft constraint oleh 17 kelas perkuliahan.**

Kata kunci*— Penjadwalan Perkuliahan Universitas, *Constraint*, *Seaquential Search

I. PENDAHULUAN

Ada dua tema besar dalam penelitian tentang penjadwalan perkuliahan universitas. Pertama adalah penelitian yang berkaitan dengan algoritma penyelesaian masalah penjadwalan perkuliahan [1] [2] [3] [4], dan yang kedua adalah penelitian yang berkaitan dengan penerapan hasil penelitian tersebut secara praktis di universitas secara nyata [5] [6] [7]. Penelitian pertama berkaitan dengan algoritma penjadwalan perkuliahan universitas yang diharapkan dapat menyusun jadwal perkuliahan dalam waktu yang singkat dan memenuhi *constraint* yang telah ditetapkan. Sementara itu penelitian kedua berupaya menjembatani kesenjangan berbagai riset tentang penjadwalan perkuliahan dengan penerapannya secara nyata di universitas.

Meskipun terkait erat dengan algoritma, paper ini berkaitan dengan persoalan kedua yaitu tentang praktek penjadwalan perkuliahan universitas di dunia nyata. Hal ini menjadi penting karena ternyata masih banyak universitas yang melakukan penjadwalan perkuliahannya secara manual [8]. Salah satu sebabnya adalah karena sangat sulit menemukan model yang berlaku secara umum [6]. Setiap universitas memiliki karakteristik unik yang pada akhirnya menjadikan *constraint* yang tidak sama dengan *constraint* pada berbagai penelitian yang telah dilakukan. Adanya *constraint* yang tidak dapat dipenuhi ini yang menyebabkan sistem penjadwalan tidak digunakan dan akhirnya penjadwalan dilakukan secara manual.

Penelitian ini berisi algoritma untuk menangani *constraint* dalam persoalan penjadwalan perkuliahan di universitas. Algoritma yang dibangun diharapkan dapat digunakan oleh universitas ketika mengimplementasikan sistem pendjadwalan perkuliahan secara nyata. Hal ini akan memperbesar kemungkinan bahwa sistem penjadwalan perkuliahan akan berjalan sesuai dengan kebutuhan universitas. Dan pada akhirnya diharapkan akan meningkatkan efisiensi pengelolaan sumberdaya universitas.

Submitted 05-05-2023; Revised 16-08-2023; Accepted 24-08-2023

Kegunaan lain dari algoritma yang dipaparkan dalam paper ini adalah bahwa berbagai algoritma tersebut dapat digunakan untuk mendapatkan nilai objektif bila persoalan penjadwalan perkuliahan ini diselesaikan dengan menggunakan pendekatan meta-heuristik [9] [10] atau berbasis riset operasi [11] [12]. Metode meta-heuristik banyak digunakan untuk berbagai masalah penjadwalan karena kemampuanya menanganai *constraint* yang kompleks [13] [14]. Misalnya untuk mengetahui berapa banyak pelanggaran *constraint* ruang yang terjadi pada sebuah solusi yang diperoleh dengan algoritma genetika atau *simulated annealing*. Berdasar nilai obektif tersebut akan dicari solusi yang paling optimal.

Penelitian ini lebih ditekankan pada identifikasi *constraint* yang digunakan di universitas dan penanganannya berbasis aturan. Pengembangan penelitian selanjutnya akan memanfaatkan pendekatan meta-heuristik, riset operasi, atau gabungan keduanya.

II. PENJADWALAN PERKULIAHAN

Penjadwalan perkuliahan adalah bagian dari kegiatan perencanaan dalam proses belajar mengajar di universitas. Perencanaan yang baik akan berpengaruh pada kelancaran proses belajar mengajar sehingga tujuan pembelajaran dapat dicapai. Penjadwalan adalah menempatkan sejumlah pertemuan dalam waktu [15]. Penjadwalan perkuliahan adalah penjadwalan mingguan untuk menempatkan matakuliah pada satu set program universitas dengan meminimalkan tumpang tindih antar matakuliah yang memiliki mahasiswa yang sama [16].

Saat ini penjadwalan perkuliahan dilakukan oleh program studi. Sebelumnya pihak universitas telah membagi ruang berdasar fakultas, dan fakultas membaginya untuk program studi. Sistem penjadwalan terkomputerisasi telah digunakan untuk input data satu persatu oleh program studi dan akan menampilkan notifikasi bila ada jadwal yang melanggar *constraint* ruang atau melanggar *constraint* dosen. Belum ada sistem penjadwalan perkuliahan otomatis di tingkat universitas. Kondisi ini menyebabkan proses penjadwalan membutuhkan waktu yang lama dan ada potensi penggunaan ruang yang kurang optimal.

A. Beberapa Istilah yang berkaitan dengan Persoalan Penjadwalan perkuliahan

Ada beberapa istilah yang perlu dijelaskan sebelumnya agar terdapat kesamaan persepsi tentang bagaimana algoritma dan sistem bekerja.

Kelas perkuliahan adalah istilah yang digunakan untuk untuk menunjukkan informasi yang terdiri dari data nama matakuliah, kelas, dan dosen pengampu. Dalam kondisi nyata, kelas perkuliahan ini adalah plot mengajar yang ditentukan oleh program studi. Kelas perkuliahan ini sangat penting dalam sistem yang dibangun karena kelas perkuliahan ini yang menjadi dasar pembentukan jadwal kuliah secara otomatis.

Jadwal perkuliahan adalah kelas perkuliahan yang dilengkapi dengan data hari, slot waktu, dan ruang. *Slot waktu* adalah waktu yang dipergunakan oleh kelas perkuliahan sesuai dengan besarnya SKS. Satu slot waktu adalah satu SKS.

B. Constraint

Constraint adalah berbagai hal yang dipertimbangkan dalam proses penyusunan jadwal perkuliahan. *Hard constraint* adalah *constraint* yang harus dipenuhi oleh jadwal perkuliahan. Pelanggaran *hard constraint* akan menyebabkan jadwal menjadi tidak layak untuk diterapkan di dunia nyata. *Soft constraint* adalah *constraint* yang pemenuhannya akan meningkatkan kualitas dari jadwal perkuliahan yang disusun. Pelanggaran *soft constraint* masih dapat diterima asalkan jumlahnya diupayakan sekecil mungkin. Tabel I menampilkan berbagai *constraint* yang digunakan dalam paper ini.

TABEL I
CONSTRAINT PENJADWALAN PERKULIAHAN DI PERGURUAN TINGGI KEAGAMAAN ISLAM

No	Constraint		
	Nama Constraint	**Keterangan**	**Jenis Constraint**
1	Waktu Belajar	Waktu belajar mulai 06.30 sampai 14.00 karena ruang kuliah digunakan untuk pembelajaran bahasa Arab	Hard
2	Dosen	Dosen tidak boleh mengajar di dua tempat yang berbeda pada waktu yang sama	Hard
3	Ruang	Ruang kuliah tidak boeh digunakan oleh dua kelas perkuliahan pada waktu yang sama	Hard
4	Shalat Dhuhur	Perkuliahan tidak boleh dilaksanakan pada saat waktu shalat dhuhur dan istirahat makan siang	Soft
5	Shalat Jum'at	Perkuliahan tidak boleh dilaksanakan pada waktu shalat jum'at	Hard
6	Preferensi Mengajar	Dosen dapat menyampaikan hari tertentu untuk mengajar	Soft
7	Matakuliah Paket Semester I	Mahasiswa semester I mengikuti matakuliah paket.	Hard
8.	Kapasitas Ruang	Kelas perkuliahan harus dijadwalkan pada ruang yang tidak melebihi kapasitas ruang	Hard
9	SKS Praktikum	Matakuliah praktikum memiliki 1 SKS tapi membutuhkan 2 slot waktu	Hard
10	Ruang Praktikum	Matakuliah praktikum membutuhkan ruang tertentu	Hard

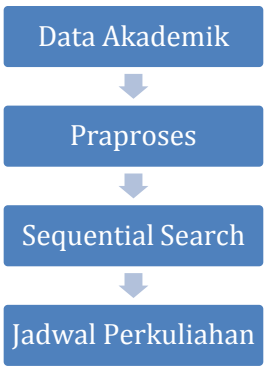
Constraint pada tabel I adalah *constraint* yang dipergunakan di UIN Maulana Malik Ibrahim Malang, salah satu PTKI di lingkungan Kementrian Agama. Secara umum berbagai *constraint* tersebut juga berlaku di berbagai universitas meskipun masing-masing universitas dapat memiliki *constraint* yang unik sesuai dengan kebijakan universitas.

III. Algoritma penanganan Constraint dalam penjadwalan perkuliahan

Sistem penjdwalan perkuliahan universitas dimulai dengan menyiapkan data akademik seperti matakuliah, dosen dan ruang. Beberapa kegiatan praproses diperlukan agar sistem penjadwalan dapat memenuhi berbagai *constraint* yang telah ditetapkan. Setelah semua data dan praproses siap maka proses penjadwalan perkuliahan otomatis akan dilaksanakan (Gambar 1).

Penjadwalan perkuliahan otomatis ini menggunakan algoritma *sequential search*. Cara kerja algoritma adalah dengan memberikan ruang, waktu, dan hari pada kelas perkuliahan yang akan dijadwalkan. Bila ruang, hari dan jam yang diberikan tersebut tidak melanggar berbagai *constraint* yang telah ditetapkan maka jadwal disimpan. Bila melanggar salah satu *constraint* maka akan dicari ruang, waktu, dan hari yang lain [17].

Output dari sistem penjadwalan perkuliahan otomatis ini adalah jadwal perkuliahan yang tidak melanggar *constraint* yang telah ditetapkan.



Gambar. 1 Alur proses penjadwalan perkuliahan otomatis

Agar sistem dapat bekerja maka setiap *constraint* harus ditangani dengan algoritma tertentu. Pada bagian ini akan diuraikan secara lengkap algoritma penanganan berbagai *constraint* yang ada pada tabel I.

A. Algoritma Penanganan Constraint Waktu Belajar

Dalam penelitian ini waktu perkuliahan ditentukan oleh universitas mulai dari jam 06.30 sampai jam 14.00. Di atas jam 14.00 ruang kuliah akan dipergunakan untuk pembelajaran bahasa Arab bagi mahasiswa semester I dan semester II. Waktu belajar ini harus dilaksanakan sesuai dengan ketentuan sehingga termasuk dalam *hard constraint*.

Constraint ini ditangani dengan membagi waktu belajar dari 06.30 sampai 14.00 menjadi slot waktu dalam range 50 menit. Slot waktu tersebut kemudian digunakan untuk membagi slot waktu berdasarkan besarnya SKS seperti ditampilkan pada tabel II. Ketika ada kelas perkuliahan yang akan dijadwalkan maka akan didapatkan terlebih dulu besar SKS nya, dan berdasarkan SKS ini maka akan diperoleh range waktu yang mungkin digunakan untuk menjadwalkan kelas perkuliahan tersebut. Selanjutnya proses penjadwalan dilanjutkan dengan pemeriksaan *constraint* yang lain.

Algoritma 1 Penangan *Constraint* Waktu Belajar	
1	Ambil data kelas_perkuliahan
2	Mendapatkan data ruang yang tersedia
3	Dapatkan informasi dari database SKS kelas perkuliahan tersebut
4	Berdasar SKS tersebut, diperoleh range waktu yang memungkinkan untuk menjadwalkan kelas perkuliahan tersebut
5	Mendapatkan Preferensi hari mengajar
6	Proses Penjadwalan menggunakan *Sequential Search*
7	Bila ruang, slot waktu, dan hari belum digunakan maka jadwal disimpan

Tabel II hanya menampilkan sebagian data, dalam kondisi nyata besarnya SKS dapat dikembangkan sesuai dengan ketentuan program studi dan universitas.

TABEL I
PENGATURAN JAM KULIAH

No	Jam Kuliah		
	Kode Jam	Range	SKS
1	101	06.30 - 07.20 (1 SKS)	1
2	102	07.20 - 08.10 (1 SKS)	1
3	103	08.10 - 09.00 (1 SKS)	1
4	104	09.00 - 09.50 (1 SKS)	1
5	105	09.50 - 10.40 (1 SKS)	1
6	106	10.40 - 11.30 (1 SKS)	1
7	107	11.30 - 12.20 (1 SKS)	1
8	108	12.20 - 13.10 (1 SKS)	1
9	109	13.10 - 14.00 (1 SKS)	1
10	201	06.30 - 08.10 (2 SKS)	2
11	202	07.20 - 09.00 (2 SKS)	2
12	203	08.10 - 09.50 (2 SKS)	2
13	204	09.00 - 10.40 (2 SKS)	2
14	205	09.50 - 11.30 (2 SKS)	2
15	206	10.40 - 12.20 (2 SKS)	2

B. Algoritma Penanganan Constraint Dosen

Constraint dosen termasuk dalam *hard constraint* dalam persoalan penjadwalan perkuliahan universitas. Bila *constraint* dosen ini dilanggar maka jadwal yang tersusun menjadi tidak layak karena tidak dapat dilaksanakan di dunia nyata.

Constraint dosen berarti bahwa seorang dosen tidak boleh mengajar di ruang yang berbeda pada waktu yang sama. Yang dimaksud dengan waktu yang sama adalah waktu yang beririsan atau menjadi bagian antara satu waktu tertentu dengan waktu yang lain. Detil kondisi yang disebut dengan waktu yang sama sehingga menyebabkan terlanggarnya *constraint* dosen, yaitu :

- Pada hari yang sama seorang dosen dijadwalkan mengajar pada jam yang tepat sama dengan kelas

perkuliahan yang telah terjadwal untuk dosen bersangkutan.
- Pada hari yang sama seorang dosen dijadwalkan pada jam yang berada pada range kelas perkuliahan yang telah terjadwal untuk dosen bersangkutan.
- Pada hari yang sama seorang dosen dijadwalkan mengajar pada range waktu yang melebihi range jam dari kelas perkuliahan yang telah terjadwal untuk dosen bersangkutan.

Dari kondisi di atas maka dapat disusun algoritma untuk menangani *constraint* dosen. Algoritma bekerja secara sekuensial dengan memeriksa apakah ruang, jam, dan hari yang diberikan pada sebuah kelas perkuliahan berada pada tiga kondisi di atas dibandingkan dengan jadwal yang telah ada. Algoritma akan memberikan nilai *true* bila kondisi di atas terjadi, yaitu dosen yang bersangkutan ternyata telah terjadwal pada waktu yang sama, dan nilai *false* jika tidak terjadi. Nilai *true* ini kemudian dapat diterjemahkan sebagai penolakan terhadap waktu yang ditetapkan dan segera dicari waktu yang lain. Nilai *false* berarti jadwal tidak berada pada kondisi tersebut sehingga jadwal dapat disimpan.

Algoritma 2 Penanganan *Constraint* Dosen

```
1   test : boolean // bernilai true bila terjadi pelanggaran
2   //Ambil jadwal yang telah tersusun
3   N : int // banyaknya jadwal
4   h : String // hari di jadwal yang telah tersusun
5   jm : long // jam mulai di jadwal yang telah tersusun
6   ja : long // jam akhir di jadwal yang telah tersusun
7   k : String // kode dosen di jadwal yang telah tersusun
8   //Ambil data kelas_perkuliahan
9   kodemk : String // kode matakuliah
10  kelas : String //  kelas
11  kodedosen : String // kode dosen
12  Berikan ruang, jam, dan hari
13  hari : String // hari dijadwalkan
14  jammulai : long // jam mulai
15  jamakhir : long // jam berakhir
16  for (i ← 0 to N)
17    if (hari == h dan jammulai >= jm dan jammulai < ja
    dan
18    kodedosen == k)
19      test ← true
20      break
      else if (hari == h dan jammulai < ja dan jamakhir > jm
21    dan kodedosen == k)
22      test ← true
23      break
      else if (hari == h dan jm >= jammulai dan ja <=
24  jamakhir
25    dan kodedosen == k)
26      test ← true
27      break
      end if
    end for
```

C. Algoritma Penanganan Constraint Ruang

Constraint ruang juga termasuk dalam hard constraint. *Constraint* ruang berarti bahwa sebuah ruang tidak boleh digunakan oleh dua atau lebih kelas perkuliahan pada waktu yang sama. Pada dasarnya algoritma penanganan *constraint* ruang sama dengan *constraint* dosen hanya saja objek pelanggarannya adalah ruang. Dengan demikian terlanggarnya constraint ruang terjadi pada kondisi :
- Pada hari yang sama satu ruang dijadwalkan pada range yang telah terjadwal untuk ruang bersangkutan.
- Pada hari yang sama satu ruang dijadwalkan pada range waktu yang melebihi range jam dari yang telah terjadwal untuk ruang bersangkutan.
- Pada hari yang sama satu ruang dijadwalkan pada jam yang tepat sama dengan yang telah terjadwal untuk ruang bersangkutan.

Langkah awal algoritma adalah memberikan ruang, jam, dan hari pada kelas perkuliahan yang akan dijadwalkan. Algoritma lalu akan melakukan pemeriksaan pada jadwal yang telah ada, apakah jadwal tersebut berada pada tiga kondisi di atas. Jika berada pada kondisi di atas maka akan diberikan nilai *true* yang menunjukkan adanya pelanggaran sehingga jadwal akan ditolak dan akan diberikan ruang, jam dam hari yang lain. Bila bernilai *false* maka jadwal akan disimpan.

Algoritma 3 Penanganan *Constraint* Ruang

```
1   test : boolean // bernilai true bila terjadi pelanggaran
2   //Ambil jadwal yang telah tersusun
3   N : int // banyaknya jadwal
4   h : String // hari di jadwal yang telah tersusun
5   jm : long // jam mulai di jadwal yang telah tersusun
6   ja : long // jam akhir di jadwal yang telah tersusun
7   r : String // kode ruang di jadwal yang telah tersusun

8   Ambil data kelas_perkuliahan
9   kodemk : String // kode matakuliah
10  kelas : String //  kelas
11  k : String // kode dosen
12  Berikan ruang, jam, dan hari pada kelas perkuliahan
    tersebut
13  ruang : String // ruang yang akan dijadwalkan
14  hari : String // hari akan dijadwalkan
15  jammulai : long // jam mulai yang akan dijadwalkan
15  jamakhir : long // jam berakhir yang akan
17  for (i ← 0 to N)
18    if (hari == h dan jammulai >= jm dan jammulai < ja
    dan
19    ruang == r)
20      test ← true
21      break
      else if (hari == h dan jammulai < ja dan jamakhir > jm
22    dan ruang == r)
23      test ← true
24      break
      else if (hari == h dan jm >= jammulai dan ja <=
25  jamakhir
26    dan ruang == r)
27      test ← true
28      break
      end if
    end for
```

D. Algoritma Penanganan Constraint Shalat Dhuhur

Constraint waktu shalat dhuhur adalah termasuk dalam *soft constraint* dalam sistem penjadwalan perkuliahan yang dibangun. Ini berarti bahwa jadwal kuliah yang melanggar waktu shalat dhuhur masih bisa dilaksanakan di dunia nyata tetapi kualitasnya berkurang karena tidak sesuai dengan ketentuan yang diharapkan. Untuk wilayah Indonesia bagian barat, shalat dhuhur dilaksanakan sekitar jam 12.00. Waktu shalat ini berubah-ubah disekitar jam tersebut sesuai dengan pergerakan matahari.

Untuk menyelesaikan *constraint* ini maka perlu diperhatikan pengaturan slot waktu yang telah ditentukan oleh universitas. Tabel menunjukkan pembagian slot waktu yang digunakan dalam sistem sesuai dengan ketentuan universitas. Karena matakuliah memiliki SKS yang berbeda maka waktu kuliah yang mengandung slot waktu tersebut perlu diidentifikasi sehingga tidak digunakan dalam penjadwalan. Tabel mengatur slot waktu untuk 1 SKS sampai 4 SKS. Bila diperlukan besarnya SKS dapat ditambahkan dan diatur dengan cara yang sama.

Waktu shalat dhuhur berada pada slot waktu antara jam 11.30 sampai jam 12.20. Identifikasi dilakukan dengan melihat tabel *JamKuliah* dimana diketahui bahwa slot waktu yang berkaitan dengan jam 11.30 – 12.20 ada pada kode jam 107 untuk 1 SKS, kode jam 106 dan 107 untuk 2 SKS, serta kode jam 305, 306, dan 307 untuk 3 SKS. Berdasar data tersebut maka akan diperoleh range waktu yang tidak melanggar waktu shalat dhuhur. Range waktu ini dipergunakan dalam proses penjadwalan perkuliahan otomatis secara sekuensial yang digunakan dalam penelitian ini. Maka algoritma yang disusun untuk menangani constraint shalat dhuhur ini merupakan bagian dari praproses dalam sistem penjadwalan perkuliahan yang dikembangkan.

Algoritma 4 *Constraint* Penanganan Waktu Shalat Dhuhur

```
1  Identifikasi waktu yang berkaitan dengan slot waktu
   shalat dhuhur
2  batas1 ← 107
3  batas21 ← 206
4  batas22 ← 207
5  batas31 ← 305
6  batas32 ← 306
7  batas33 ← 307
8  Pilih range waktu sesuai dengan besarnya SKS dan tidak
   mengandung slot waktu batas di atas
9  rangeJam : Array bertipe String
10 Menggunakan rangeJam sebagai slot waktu yang valid
   dalam penjadwalan secara sequential search
```

E. Algoritma Penanganan Constraint Waktu Shalat Jum'at

Constraint waktu shalat jum'at merupakan *hard constraint*. Ini berbeda dengan *constraint* waktu shalat dhuhur yang merupakan *soft constraint* dengan pertimbangan bahwa shalat jum'at waktunya tertentu dan ada khutbah yang wajib diikuti, sementara shalat dhuhur memiliki range waktu yang lebih panjang.

Constraint shalat jumat memiliki durasi yang lebih panjang dibandingkan dengan salat dhuhur. Constraint shalat jumat hanya berlaku untuk hari jumat dari jam 11.30 sampai 13.10 yang setara dengan dua slot waktu. Agar sistem dapat mengidentifikasi waktu shalat jum'at ini maka perlu dilakukan praproses dengan mengidentifikasi slot waktu yang beririsan dengan waktu shalat jum'at ini. Identifikasi dilakukan mulai dari slot waktu 1 SKS sampai 3 SKS (Tabel III).

Setelah identifikasi dilakukan maka Langkah selanjutnya adalah melaksanakan proses penjadwalan secara sekuensial dimana bila sebuah kelas perkuliahan terjadwal hari jumat dan berada pada slot waktu shalat jumat maka sitem tidak akan memproses lebih lanjut jadwal tersebut tetapi bila selain slot waktu tersebut maka jadwal akan dilakukan pemeriksaan *constraint* yang lain.

TABEL II
SLOT WAKTU TERKAIT SHALAT JUMAT

No	Slot Waktu		
	1 SKS	**2 SKS**	**3 SKS**
1	10.40 - 11.30	09.50 - 11.30	09.00 - 11.30
2	11.30 - 12.20	10.40 - 12.20	09.50 - 12.20
3	12.20 - 13.10	11.30 - 13.10	10.40 - 13.10
4		12.20 - 14.00	11.30 - 14.00
5			12.20 - 14.50

Algoritma 5 Penanganan *Constraint* Waktu Shalat Jum'at

```
1  for i ← 0 to seluruh ruang
2    for j ← to seluruh range jam
3      for k ← seluruh hari
4        if (hari ← jumat dan range slot waktu jumat 1 SKS)
5        else if (hari ← jumat dan range slot waktu jumat 2
           SKS)
6        else if (hari ← jumat dan range slot waktu jumat 3
           SKS)
7        else
8          Pemeriksaan constraint yang lain
9        End if
10     End for
11   End for
12 End for
```

F. Algoritma Penanganan Constraint Preferensi Mengajar

Preferensi mengajar adalah hari mengajar yang lebih disukai oleh dosen untuk mengajar dibanding hari yang lain. Selain mengajar, kegiatan dosen adalah melaksanakan penelitian dan melakukan pengabdian masyarakat. Pemenuhan preferensi mengajar diharapkan dapat memberikan kenyamanan bagi dosen dalam mengajar dan melaksanakan tugas-tugas yang lain. Constraint preferensi mengajar ini merupakan *soft constraint* sehingga dapat dilanggar bila keadaan tidak memungkinkan.

Constraint ini ditangani dengan mencatat preferensi hari mengajar tiap dosen dan menggunakan data itu sebagai hari dalam penentuan jadwal dosen bersangkutan. Data ini disimpan dalam tabel Preferensi Mengajar seperti ditampilkan pada tabel IV. Data preferensi mengajar ini

akan diambil dari database ketika ada kelas perkuliahan yang akan dijadwalkan dan akan digunakan untuk menentukan hari pada saat pemberian ruang dan slot waktu. Dengan cara ini maka dapat dipastikan setiap dosen akan mengajar sesuai dengan preferensi hari yang telah dipilih. Kemungkinan lain adalah kelas perkuliahan tersebut tidak terjadwal bila pada hari yang ditentukan ternyata dosen telah terjadwal atau ruang telah digunakan.

TABEL IV
REFERENSI MENGAJAR

No	Preferensi Mengajar			
	Semester	**Tahun**	**Kose Dosen**	**Preferensi**
1	1	2122	65001	1
2	1	2122	65001	2
3	1	2122	65001	3
4	1	2122	65001	4
5	1	2122	65001	5

Algoritma 6 Penanganan *Constraint* Preferensi Hari Mengajar

1 Ambil data kelas_perkuliahan
2 Dapatkan informasi dari database apakah preferensi hari mengajar untuk dosen berangkutan
3 Lakukan perulangan mulai dari ruang pertama sampai ruang terakhir
4 Lakukan perulangan sesuai preferensi hari
5 Lakukan perulangan sesuai dengan slot waktu
6 Beri ruang, jam, dan hari pada data kelas_perkuliahan
7 Proses Penjadwalan menggunakan *Sequential Search*

G. Algoritma Penanganan Constraint Matakuliah Paket untuk Mahasiswa Semester I

Mahasiswa semester II keatas memilih sendiri kelas perkuliahan yang akan diikuti. Program studi menyedaiakan kelas perkuliahan beserta jadwalnya sesuai dengan kurikulum dan mahasiswa bebas memilih kelas perkuliahan tersebut.

Mahasiswa semester satu tidak memilih matakuliah sendiri seperti mahasiswa semester dua keatas. Program studi membagi mahasiswa semester satu menjadi beberapa kelompok mahasiswa. Kelompok mahasiswa ini kemudian diberi label kelas, misalnya kelas A, B, C dan seterusnya sesuai dengan banyaknya mahasiswa. Misalnya jumlah mahasiswa baru sebanyak 120 dan setiap kelas terdiri dari 30 mahasiswa maka terdapat 4 kelas yang diberi nama kelas A, B, C dan D. Maka setiap kelas ini tidak boleh dijadwalkan pada waktu yang sama untuk matakuliah yang berbeda karena pesertanya adalah mahasiswa yang sama. Inilah yang disebut dengan *constraint* matakuliah paket semester I.

Untuk menangani constraint mahasiswa semester I ini maka perlu diidentifikasi terlebih dulu apakah sebuah kelas perkuliahan adalah matakuliah untuk mahasiswa semester I. identifikasi dapat dilakukan dengan memberi tanda pada tabel Matakuliah_Pemasaran untuk matakuliah yang dipasarkan bagi mahasiswa semester I (Tabel V).

TABEL V
MATAKULIAH SEMESTER 1

No	Matakuliah Semester I			
	Kode Matakuliah	**Tahun**	**Smt**	**Semester Matakuliah**
1	1565002	2122	1	1
2	1565003	2122	1	1
3	1565004	2122	1	1
4	1565031	2122	1	1
5	20000011A01	2122	1	1
6	20000011A03	2122	1	1
7	20000011A04	2122	1	1
8	20000011A05	2122	1	1
9	20000011A10	2122	1	1
10	20060511C01	2122	1	1

Dalam tabel V, terdapat kolom Semester Matakuliah yang menunjukkan bahwa matakuliah tertentu ditujukan kepada mahasiswa semester I. Didasarkan pada informasi tersebut maka kelas perkuliahan yang akan dijadwalkan perlu dilakukan pemeriksaan apakah jadwal yang sedang disusun tidak bersamaan waktunya dengan kelas yang sama. Bila waktunya tidak bersamaan dengan jadwal yang telah ada maka jadwal akan disimpan.

Algoritma 7 Penanganan *Constraint* Matakuliah Paket untuk Mahasiswa Semester I

1 Ambil data kelas_perkuliahan
2 Periksa apakah kelas perkuliahan tersebut adalah matakuliah semester 1
3 Jika kelas perkuliahan tersebut adalah matakuliah semester 1 maka saat penentuan hari dan slot waktu harus diperiksa apakah slot waktu tersebut telah digunakan oleh kelas yang sama. Jika belum digunakan maka jadwal bisa disimpan jika sudah digunakan maka dicarikan slot waktu yang lain.

H. Algoritma Penanganan Constraint Kapasitas Ruang

Sebuah kelas perkuliahan memiliki kapasitas kelas, yaitu banyaknya mahasiswa yang dapat mengambil kelas perkuliahan tersebut. Kapasitas kelas ini dimaksudkan agar tidak terjadi penumpukan di kelas tertentu dan juga mempertimbangkan kapasistas ruang kelas. Sebuah kelas perkuliahan tidak boleh dijadwalkan pada ruang yang kapasitasnya lebih kecil dari jumlah mahasiswa pada kelas perkuliahan tersebut.

Data kapasitas kelas perkuliahan ditentukan oleh program studi pada saat proses penentuan dosen pengampu matakuliah dan data kapasitas ruang ada pada tabel Ruang. Pada saat penentuan ruang untuk sebuah kelas perkuliahan akan dilakukan pemeriksaan apakah daya tampung ruang lebih besar atau sama dengan kapasitas kelas perkuliahan. Bila kapasitas kelas perkuliahan lebih kecil atau sama dengan kapasitas ruang maka jadwal dapat diterima bila tidak maka akan dicarikan ruang, slot waktu dan hari yang lain.

Algoritma 8 Penanganan *Constraint* Kapasitas Ruang	
1	test : boolean // bernilai true bila terjadi pelanggaran
2	Ambil data kapasistas kelas dari kelas perkuliahan
2	Berikan ruang, jam, dan hari
3	Ambil data kapasitas ruang dari tabel Ruang
4	if kapasistas kelas > kapasitas ruang
5	test ← true
6	Cari ruang, jam, dan hari yang lain
7	end if

I. Algoritma Penanganan Constraint SKS Praktikum

Untuk menangani constraint ini perlu dilakukan pemeriksaan apakah sebuah kelas perkuliahan merupakan matakuliah praktikum atau bukan. Bila kelas perkuliahan ini merupakan matakuliah praktikum maka diberi tanda 1 dan bila bukan matakuliah praktikum maka tandanya bukan 1. Tanda ini diletakkan pada tabel Matakuliah. Pada field Mtk_Jenis. Tanda ini akan digunakan untuk menentukan besarnya SKS, yaitu bila Mtk_Jenis = 1 maka akan diberikan 2 SKS dan bila tidak ada tanda 1 maka besarnya SKS sesuai dengan field Mtk_SKS.

Algoritma 9 Penanganan *Constraint* SKS Praktikum	
1	Ambil data kelas_perkuliahan
2	Dapatkan informasi dari database apakah jenis dari matakuliah ini
3	Bila jenis = 1 maka
4	SKS ← 2
5	Bila jenis ≠ 1 maka
	SKS ← sesuai SKS di database
6	Beri ruang, jam, dan hari pada kelas_perkuliahan tersebut
7	Proses pemeriksaan *constraint* lain

J. Algoritma Penanganan Constraint Ruang Praktikum

Matakuliah praktikum memutuhkan ruang khusus yang sesuai dengan peralatan atau software yang disediakan oleh laboratorium. Penentuan ruang laboratorium untuk matakuliah tertentu ditentukan oleh program studi sebelum proses penjadwalan dilaksanakan.

Algoritma 10 Penanganan Constraint Ruang Praktikum	
1	Ambil data kelas_perkuliahan
2	Beri ruang, jam, dan hari pada data kelas_perkuliahan sesuai data SKS
3	Dapatkan informasi dari database apakah perlu ruang khusus
4	Bila ruang perlu ruang khusus maka
5	ruang = ruang_khusus
6	Proses Penjadwalan menggunakan *Sequential Search*
7	Bila ruang pada jadwal = ruang_khusus
8	Jadwal disimpan

K. Hasil dan Diskusi

Uji coba yang dilakukan pada data akademik di program studi Teknik Informatika UIN Maulana Malik Ibrahim Malang semester Ganjil tahun akademik 2021/2022. Terdapat 211 kelas perkuliahan yang harus dijadwalkan. Uji coba dilaksanakan pada spesifikasi komputer Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz dengan RAM 8 GB.

Hasil uji coba sistem yang telah dibangun menunjukkan bahwa dari 211 kelas perkuliahan yang harus dijadwalkan, sebanyak 190 (90%) kelas perkuliahan terjadwal tanpa melanggar constraint yang telah ditetapkan. Terdapat 21 (10%) kelas perkuliahan belum terjadwal. Dari 21 kelas perkuliahan yang belum terjadwal tersebut ada 4 kelas perkuliahan yang memiliki memiliki 2 SKS dan 17 kelas perkuliahan memiliki 3 SKS. Waktu yang diperlukan untuk proses penjadwalan rata-rata adalah 35,32 detik.

Dua puluh satu kelas perkuliahan yang tidak terjadwal disebabkan oleh dua hal, yaitu slot waktu yang sudah habis terpakai oleh kelas perkuliahan yang lain, atau slot waktu yang tersedia berada pada ruang yang kapasitasnya tidak mencukupi untuk menampung peserta kuliah. Tabel VI menunjukkan bahwa di ruang kelas, semua slot telah digunakan sehingga nilainya 0 karena sudah tidak ada slot yang tersisa. Beberapa ruang laboratorium masih memiliki beberapa slot waktu tetapi kapasitasnya lebih kecil dari kapasitas kelas perkuliahan sehingga tidak dapat dipergunakan.

Masih ada kemungkinan bahwa lebih banyak kelas perkuliahan yang dapat dijadwalkan mengingat bahwa algoritma *sequential search* yang digunakan dalam paper ini bekerja ke arah depan, tanpa ada peninjauan kembali untuk mengetahui apakah jadwal yang tersusun merupakan jadwal yang optimal. Algoritma ini juga belum ada proses pertukaran jadwal, sehingga slot waktu yang telah ditempati tidak dapat diganti dengan kelas perkuliahan yang lain.

TABEL VI
SISA SLOT WAKTU

No	Ruang	Sisa Slot	
		2 SKS	3 SKS
1	B.306	0	0
2	B.307	0	0
3	B.314	0	0
4	B.315	0	0
5	B.316	0	0
6	B.317	0	0
7	B.318	0	0
8	D.222	0	0
9	LAB. PROGRAMMING	5	0
10	LAB. MOBILE PROGRAMMING	6	0
11	LAB. DATABASE	3	0
12	LAB. INTELLIGENT	6	0
13	LAB. DIGITAL DAN ROBOTIK	6	0
14	LAB. COMPUTER NETWORK	6	0
15	LAB. MULTIMEDIA	6	0
16	LAB. SOFTWARE ENGINEERING	12	6

Dalam kondisi nyata, semua kelas perkuliahan harus terjadwal karena telah ditetapkan oleh program studi. Kelas perkuliahan yang tidak terjadwal menyebabkan sistem menurun kualitasnya. Agar hal ini tidak terjadi, maka perbaikan dapat dilakukan menggunakan aplikasi

penjadwalan perkuliahan interaktif [18]. Dengan cara ini maka seluruh kelas perkuliahan dapat dijadwalkan meskipun ada 17 kelas perkuliahan yang melanggar *soft constraint*.

IV. KESIMPULAN

Dengan menggunakan algoritma yang telah disusun maka sistem dapat menjadwalkan 90% dari seluruh kelas perkuliahan. Sebanyak 10% dijadwalkan secara interaktif untuk menjamin bahwa semua kelas perkuliahan terjadwal. Dari penjadwalan interaktif tersebut terdapat 17 kelas perkuliahan yang melanggar *soft constraint*.

Penambahan *constraint* dapat dilakukan sesuai dengan kebutuhan universitas. *Constraint* akan menjadi semakin luas ketika semua harapan pemangku kepentingan dipertimbangkan dalam penyusunan jadwal perkuliahan universitas.

Adanya kelas perkuliahan yang belum terjadwal dan kelas perkuliahan yang terjadwal tetapi melanggar *soft constraint* menunjukkan adanya kebutuhan metode penyusunan jadwal yang lebih baik sehingga diharapkan seluruh kelas perkuliahan dapat terjadwal secara otomatis dengan pelanggaran *soft constraint* yang kecil atau bahkan tanpa pelanggaran sama sekali.

Penelitian selanjutnya diharapkan dapat menampung seluas mungkin *constraint* dalam penjadwalan perkuliahan dan eksplorasi berbagai pendekatan untuk menyelesaikan *constraint* tersebut.

REFERENSI

[1] A. E. Phillips, H. Waterer, M. Ehrgott, and D. M. Ryan, "Integer programming methods for large-scale practical classroom assignment problems," *Comput. Oper. Res.*, vol. 53, pp. 42–53, Jan. 2015, doi: 10.1016/j.cor.2014.07.012.

[2] K. Sylejmani, E. Gashi, and A. Ymeri, "Simulated annealing with penalization for university course timetabling," *J. Sched.*, Jul. 2022, doi: 10.1007/s10951-022-00747-5.

[3] G. Alnowaini and A. A. Aljomai, "Genetic Algorithm For Solving University Course Timetabling Problem Using Dynamic Chromosomes," in *2021 International Conference of Technology, Science and Administration (ICTSA)*, Mar. 2021, pp. 1–6. doi: 10.1109/ICTSA52017.2021.9406539.

[4] D. Wahyuningsih, "Rancangan Sistem Penjadwalan Akademik Menggunakan Algoritma Max Min Ant System (Studi Kasus: STMIK Atma Luhur Pangkalpinang)," *J. Edukasi Dan Penelit. Inform. JEPIN*, vol. 1, no. 2, Nov. 2015, doi: 10.26418/jp.v1i2.11654.

[5] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, "Practices in timetabling in higher education institutions: a systematic review," *Ann. Oper. Res.*, vol. 275, no. 1, pp. 145–160, Apr. 2019, doi: 10.1007/s10479-017-2688-8.

[6] B. McCollum, "A Perspective on Bridging the Gap Between Theory and Practice in University Timetabling," in *Practice and Theory of Automated Timetabling VI*, E. K. Burke and H. Rudová, Eds., in Lecture Notes in Computer Science, vol. 3867. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 3–23. doi: 10.1007/978-3-540-77345-0_1.

[7] T. Müller and H. Rudová, "Real-life curriculum-based timetabling with elective courses and course sections," *Ann. Oper. Res.*, vol. 239, no. 1, pp. 153–170, Apr. 2016, doi: 10.1007/s10479-014-1643-1.

[8] M. Lindahl, *Strategic, Tactical and Operational University Timetabling*. DTU Management, 2017.

[9] I. Balan, "A New Genetic Approach for Course Timetabling Problem," *J. Appl. Comput. Sci. Math.*, vol. 15, no. 1, pp. 9–14, 2021, doi: 10.4316/JACSM.202101001.

[10] S. L. Goh, G. Kendall, and N. R. Sabar, "Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem," *J. Oper. Res. Soc.*, vol. 70, no. 6, pp. 873–888, Jun. 2019, doi: 10.1080/01605682.2018.1468862.

[11] D. S. Holm, R. Ø. Mikkelsen, M. Sørensen, and T. J. R. Stidsen, "A graph-based MIP formulation of the International Timetabling Competition 2019," *J. Sched.*, vol. 25, no. 4, pp. 405–428, Aug. 2022, doi: 10.1007/s10951-022-00724-y.

[12] E. Rappos, E. Thiémard, S. Robert, and J.-F. Hêche, "A mixed-integer programming approach for solving university course timetabling problems," *J. Sched.*, vol. 25, no. 4, pp. 391–404, Aug. 2022, doi: 10.1007/s10951-021-00715-5.

[13] R. E. Febrita and W. F. Mahmudy, "Modified genetic algorithm for high school time-table scheduling with fuzzy time window," in *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, Nov. 2017, pp. 88–92. doi: 10.1109/SIET.2017.8304115.

[14] R. Rody, W. F. Mahmudy, and I. P. Tama, "Using Guided Initial Chromosome of Genetic Algorithm for Scheduling Production-Distribution System," *J. Inf. Technol. Comput. Sci.*, vol. 4, no. 1, pp. 26–32, Jun. 2019, doi: 10.25126/jitecs.20194195.

[15] E. Burke, K. Jackson, J. H. Kingston, and R. Weare, "Automated University Timetabling: The State of the Art," *Comput. J.*, vol. 40, no. 9, pp. 565–571, Jan. 1997, doi: 10.1093/comjnl/40.9.565.

[16] S. Ceschia, L. Di Gaspero, and A. Schaerf, "Educational timetabling: Problems, benchmarks, and state-of-the-art results," *Eur. J. Oper. Res.*, Jul. 2022, doi: 10.1016/j.ejor.2022.07.011.

[17] Z. Abidin, "Solving University Course Timetabling Problem (UCTP) Using Depth First Search (DFS) Algorithm and Rule Base Systems," *Int. J. Eng.*.

[18] A. N. Afandi and M. Z. Arifin, "Sistem Penjadwalan Perkuliahan Universitas Otomatis Menggunakan Algoritma Sequential Search," 2022.