

# Metode Simulated Annealing untuk Optimasi Penjadwalan Ujian Perguruan Tinggi

<sup>1</sup>Wiktasari, <sup>2</sup>Sirli Fahriah

<sup>1,2</sup>Program Studi Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Semarang  
E-mail : <sup>1</sup>wiktasari@polines.ac.id, <sup>2</sup>sirlifahriah94@gmail.com

## Abstrak

Penjadwalan ujian (*timetabling*) merupakan penugasan mata kuliah yang diujikan yang diikuti oleh mahasiswa pada slot waktu dan ruang yang tersedia dengan melibatkan batasan tertentu. *Simulated annealing* merupakan salah satu metode *heuristic* yang dapat digunakan sebagai metode pencarian dan memberikan solusi yang dapat diterima (*objective function*) dengan hasil yang baik. Pada penelitian ini membahas tentang penjadwalan ujian pada perguruan tinggi menggunakan metode *simulated annealing* dengan menggunakan lima variabel data yaitu mata kuliah yang diujikan (X1), mahasiswa (X2), slot waktu yang terdiri dari hari (X3) dan waktu periode (X4) dan variabel ruang (X5). Penelitian ini memiliki dua *objectivefunction* yang akan dihasilkan, pertama adalah penugasan mata kuliah yang diujikan yang diikuti oleh mahasiswa pada slot hari dan waktu yang tersedia, kedua penugasan hasil optimasi *objective function* 1 pada ruang yang tersedia. *Objective function* dihitung dengan memperhatikan batasan yang terlibat untuk menghasilkan solusi yang optimal. Penelitian ini melakukan uji coba terhadap metode *simulated annealing* dengan menghasilkan rata-rata varian sebesar 84,994% data dapat mencapai solusi dengan standar deviasi sebesar 1.0267. Pada penelitian ini diberikan metode solusi dalam penggunaan ruang pencarian yang tersisa untuk dapat digunakan kembali oleh data yang belum teralokasikan.

**Kata kunci :** batasan, heuristic, objective function, penjadwalan, timetabling, simulated annealing.

## Abstract

*Scheduling examinations (schedules) is the assignment of tested courses that are followed by students in available time and space slots with certain limitations. Annealing simulation is a heuristic method that can be used as a search method and provides an acceptable solution (objective function) with good results. In this study, it discusses the scheduling of exams in universities using the annealing simulation method using five data variables, namely the subjects tested (X1), students (X2), time slots consisting of days (X3) and time periods (X4) and variables. space (X5). This study has two objective functions that will be generated, first is the assignment of tested subjects which are followed by students in the available day and time slots, the second is the assignment of optimization results for objective function 1 in the available space. The objective function is calculated by taking into account the constraints involved to produce an optimal solution. Research that conducted trials on the annealing simulation method by producing an average variant of 84.994% data can reach a solution with a standard deviation of 1.0267. This research provides a solution method in using the remaining search space to be reused by unallocated data*

**Keywords :** constraint, heuristic, objective function timetabling, simulated annealing.

## I. PENDAHULUAN

Permasalahan penjadwalan pada perguruan tinggi merupakan permasalahan mendapat perhatian beberapa dekade terakhir karena dianggap sulit. Permasalahan penjadwalan pada perguruan tinggi dibagi menjadi dua yaitu penjadwalan mata kuliah dan ujian [10]. Permasalahan penjadwalan ujian membutuhkan penempatan ujian yang diambil oleh mahasiswa kedalam slot ruang dan waktu dengan memenuhi batasan-batasan tertentu yang tidak bisa dilanggar dan disebut dengan *hard constraint*. *Hard constraint* misalnya adalah mahasiswa tidak bisa mengikuti lebih dari satu ujian dalam waktu yang

sama, kapasitas ketersediaan ruang dan urutan ujian yang harus dilaksanakan [1]. Permasalahan penjadwalan ujian dapat dijelaskan sebagai kumpulan ujian yang terjadwal dengan mempertimbangkan *hard* dan *soft constraint* [2].

Kendala dalam penjadwalan ujian dapat mencakup permasalahan yang luas. Permasalahan yang dihadapi dapat berupa kapasitas yang terbatas, jadwal mahasiswa yang bentrok, jumlah maksimal ujian yang diadakan dalam sehari dan jumlah maksimal ujian yang dilaksanakan mahasiswa dalam sehari [3].

Penjadwalan ujian universitas dapat diperlakukan sebagai permasalahan optimasi yang

bertujuan untuk optimasi *objective function*. *Objective function* dapat berupa pemenuhan semua batasan yang ada baik berupa sumber daya yang terbatas, slot waktu yang sedikit dan lain sebagainya [4]. Salah satu metode yang digunakan untuk permasalahan optimasi adalah *simulated annealing* (SA). Algoritma SA telah digunakan untuk permasalahan optimasi untuk tujuan optimasi. Optimasi adalah metode yang dapat digunakan untuk meminimalisasi *cost* yang diperlukan dalam menemukan solusi [5].

Metode yang dipilih *simulated annealing* dapat diterapkan pada berbagai jenis masalah *timetabling* atau penjadwalan dengan menggunakan berbagai macam *hard constraint* dan *soft constraint* didapatkan *fitness function* yang baik atau optimum [6].

Manfaat yang diharapkan dari hasil penelitian ini adalah dapat memaksimalkan penggunaan sumber daya yang terlibat dalam proses penjadwalan sehingga menghasilkan jadwal ujian lebih baik. Penelitian ini menggunakan data ujian prodi Teknik Informatika Polines memiliki pertimbangan penempatan ujian yang diambil oleh mahasiswa dengan batasan mahasiswa tidak dapat menghadiri lebih dari satu ujian dalam satu waktu sesuai dengan slot ruang yang tersedia.

## II. METODE PENELITIAN

Penjadwalan merupakan alokasi sumber daya yang tersedia dari waktu ke waktu dalam menjalankan tugas tertentu untuk mencapai tujuan tertentu. Tujuan penyelesaian penjadwalan ujian adalah untuk menempatkan mahasiswa yang mengikuti ujian pada sumber daya yang diperlukan seperti slot hari, slot waktu dan ruang [7].

### 2. 1. Model Permasalahan Penjadwalan Ujian

Penjadwalan mata kuliah perguruan tinggi merupakan penugasan himpunan mata kuliah kedalam slot waktu yang tersedia dengan memperhatikan batasan. Optimasi himpunan dengan beberapa kriteria variabel sebagai input akan menghasilkan keputusan dengan memperhatikan batasan yang terkait untuk menghasilkan jadwal ujian [7].

Tujuan yang diharapkan dari penjadwalan adalah dapat menjadwalkan semua mata kuliah yang diujikan kedalam slot waktu yang tersedia dengan menggunakan ruang kelas yang tersedia dengan memperhatikan semua *hard constraint*.

### 2. 2. Formulasi Simulated Annealing

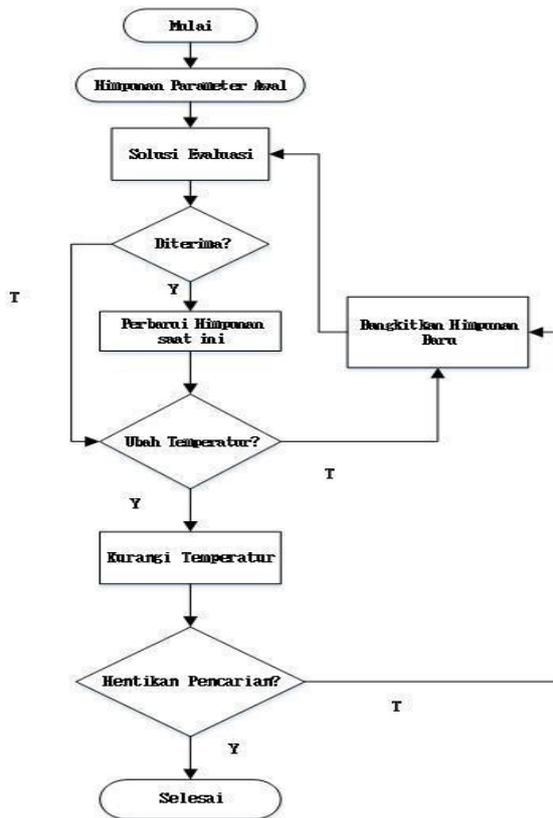
Terdapat beberapa kebutuhan dalam formulasi optimasi metode SA yaitu :

1. Melakukan spesifikasi nilai tunggal dari *objective function* baik dalam bentuk ekspresi tertutup maupun sebuah prosedur untuk komputasi.
2. Mendeskripsikan variabel (argumen dari *objective function*) dan daerah dimana akan dilakukan pencarian solusi atau yang disebut dengan *constraint*. Variabel yang digunakan adalah variabel input yang terkait dengan variabel dalam pengambilan keputusan.
3. Mendefinisikan tetangga dari variabel. Tetangga ini harus bernilai kecil dibanding dengan nilai *objective function* dari tetangga-tetangga terdekatnya.
4. Membuat sebuah prosedur yang menghasilkan langkah-langkah *pseudo-random* untuk melewati tetangga.
5. Membuat kriteria perhentian langkah-langkah *random* yang dibuat pada langkah 4. Langkah perhentian dibedakan menjadi dua kasus, yaitu:
  - a. Nilai optimal *objective function* dan lokasi yang akan ditentukan telah diketahui sebelumnya.
  - b. Baik nilai optimal maupun lokasinya tidak diketahui.

### 2. 3. Algoritma Simulated Annealing

*Simulated Annealing* merupakan metode pencarian global untuk menyelesaikan permasalahan optimalisasi. Metode ini digunakan untuk mendapatkan solusi dari hasil optimalisasi sumber daya yang terkait dengan pencarian. *Simulated Annealing* menghasilkan hasil acak dari solusi yang diharapkan, sehingga metode ini dapat menghasilkan solusi dengan beberapa kemungkinan [8].

*Simulated Annealing* memiliki kemampuan untuk keluar dari lokal optimum berdasarkan aturan penerimaan solusi kandidat. Aturannya adalah jika nilai *objective function* pada solusi sekarang lebih kecil dibanding solusi dulu, maka solusi sekarang diterima. T adalah parameter pengendali temperatur [9]. Parameter T merupakan pengendali ruang pencarian atau ruang penempatan partikel pada proses pendinginan. Flowchart algoritma SA dapat dilihat pada Gambar [5] :



Gambar 1. Flowchart Algoritma SA

1. Populasi Awal yaitu menentukan definisi dari nilai parameter yang akan digunakan.
2. Temperatur Awal yaitu temperatur atau suhu pada SA digunakan sebagai parameter pengendali yang harus didefinisikan secara hati-hati pada saat penentuan populasi awal.
3. Mekanisme Gangguan yaitu metode yang berfungsi untuk membuat solusi baru yang berasal dari solusi sekarang yaitu melakukan eksplorasi tetangga dari solusi sekarang dan melakukan perubahan pada solusi sekarang. Pada umumnya SA digunakan dalam optimasi bilangan integer untuk permasalahan kombinatorial.
4. *Objective Function* merupakan ekspresi yang berhubungan dengan parameter dengan beberapa property untuk diminimasi atau dimaksimalkan.
5. Jadwal Pendinginan, jadwal pendinginan pada umumnya digunakan dalam aturan geometrik untuk variasi temperatur :  
 $T_{i+1} = sT_i$   
 dengan  $s < 1$ .
6. Kriteria Akhir, terdapat beberapa metode untuk pengendali perhentian dari sebuah algoritma, yaitu:
  - a. Jumlah maksimum perulangan
  - b. Nilai minimum temperatur
  - c. Nilai minimum *objective function*
  - d. Nilai minimum kecepatan penerimaan.

Ada 6 tahapan algoritma SA yang digunakan untuk menyelesaikan permasalahan optimasi [11]

1. Memilih titik awal  $X_0$  secara acak atau ditentukan terlebih dahulu berdasar pengetahuan dari permasalahan yang dihadapi dan menentukan nilai yang sesuai dengan *objective function*  $\phi(X_0)$ ;
2. Memilih langkah acak  $s \in X$ ;
3. Pada titik  $X_k$  hitung perubahan sementara nilai *objective function*  
 $\Delta f \leftarrow f(s') - f(s)$ ;
4. Memeriksa kriteria perhentian, jika memenuhi berhenti, sebaliknya;
5. Untuk  $\Delta f \leq 0$  tentukan  $X_{k+1} + \Delta f$ , kembali kelangkah (2);
6. Untuk  $\Delta f > 0$  periksa  $\exp(-\beta\Delta f)$  pilih nilai acak  $\rho$ ,  $0 < \rho < 1$ , dari beberapa spesifikasi distribusi pada  $[0,1]$  kemudian bandingkan :
  - a. Jika  $\exp(-\beta\Delta f) \leq \rho$  kembali ke langkah (5)
  - b. Jika  $\exp(-\beta\Delta f) > \rho$  kembali ke langkah (2)

#### 2. 4. Constraint dalam Penjadwalan Ujian

Dalam penjadwalan ujian terdapat beberapa *hard constraint* dan *soft constraint* yaitu [3]:

##### 1. *Hard constraint*

- a. Jumlah mahasiswa yang ujian dalam sehari.
- b. Jumlah ujian yang diadakan dalam sehari.
- c. Setiap mahasiswa ujian maksimal satu kali dalam sehari.
- d. Kapasitas ruang dan waktu ujian.
- e. Ujian dengan mata kuliah yang sama harus diadakan dalam waktu yang sama

##### 2. *Soft constraint*

- a. Ujian tidak dapat diadakan pada waktu tertentu
- b. Dosen pengampu harus menjadi pengawas ujian.

#### 2. 5. Pernyataan dan Formulasi Permasalahan

##### 2. 5. 1. Himpunan

Himpunan yang digunakan dalam pemodelan untuk permasalahan penjadwalan ujian terdiri dari himpunan ujian yang terjadwalkan, mahasiswa, slot hari, slot waktu per hari dan slot ruang dapat dinotasikan sebagai berikut di bawah ini [3]:

- $I$  himpunan ujian terjadwalkan,  $i', i \in \{1, \dots, I\}$
- $J$  himpunan hari,  $j \in \{1, \dots, J\}$
- $K$  himpunan slot waktu per hari,  $k \in \{1, \dots, K\}$
- $E$  himpunan slot ruang,  $e \in \{1, \dots, E\}$
- $L$  himpunan mahasiswa,  $l \in \{1, \dots, L\}$

2. 5. 2. *Parameter Input*

Parameter input yang digunakan dalam pemodelan untuk permasalahan penjadwalan dapat ditunjukkan sebagai berikut :

- $C$  jumlah ruang kelas yang tersedia
- $H_i$  jumlah slot waktu yang dibutuhkan per ujian  $i(i \in I)$
- $LT_{ijk}$  jumlah minimum mahasiswa ujian dalam sehari dalam slot waktu yang tersedia
- $UT_{ijk}$  jumlah maksimum mahasiswa ujian dalam sehari dalam slot waktu yang tersedia

2. 5. 3. *Variabel Keputusan*

Variabel keputusan yang digunakan dalam pemodelan untuk permasalahan penjadwalan dapat ditunjukkan sebagai berikut :

- $X_{lijk} = 1$  jika mahasiswa  $l$  mengikuti ujian  $i$  pada hari  $j$  dan slot waktu  $k$ ; sebaliknya 0 maka  $(\forall l \in L, \forall i \in I, \forall j \in J, \forall k \in K)$
- $Y_{ijke} = 1$  jika ujian  $i$  diadakan pada hari  $j$ , slot waktu  $k$  dan ruang  $e$ , sebaliknya 0 maka  $(\forall i \in I, \forall j \in J, \forall k \in K, \forall e \in E)$
- $U_{lijk} = 1$  jika mahasiswa  $l$  mengikuti ujian  $i$  pada hari  $j$  mulai pada slot waktu  $k$ ; sebaliknya 0 maka  $(\forall l \in L, \forall i \in I, \forall j \in J, \forall k \in K)$
- $P_{li} = 1$  jika mahasiswa  $l$  mengikuti ujian  $i$ ; 0 maka  $(\forall l \in L, \forall i \in I)$
- $L_l$  jumlah ujian yang diikuti oleh mahasiswa  $l(i \in I)$
- $V_l$  jumlah ujian dalam sehari  $\forall i \in I$

2. 5. 4. *Objective Function :*

$$[\text{Mahasiswa} - \text{Ujian}] \text{Max} = \sum_i \sum_j \sum_k X_{lijk} \dots \dots \dots (1)$$

$$[\text{Plotting} - \text{Ujian}] \text{Max} = \sum_j \sum_k \sum_e Y_{ijke} \dots \dots \dots (2)$$

$$\sum_{i \in I} X_{lijk} \leq 1 (l \in L, j \in J, k \in K) \dots \dots \dots (3)$$

$$\sum_j \sum_k Y_{ijke} \leq C (k \in K) \dots \dots \dots (4)$$

$$\sum_{i \in I} Y_{ijke} \leq 1 (j \in J, k \in K, e \in E) \dots \dots \dots (5)$$

$$\sum_{k \in K} Y_{ijke} = X_{lijk} \dots \dots \dots (6)$$

$$X_{lijk} = 0 (l \in L, i \notin I, j \in J, k \in K) \dots \dots \dots (7)$$

$$\sum_{j \in J} \sum_{k \in K} \sum_{e \in E} Y_{ijke} = L_l (i \in I) \dots \dots \dots (8)$$

$$\sum_{i \in I} X_{lijk} \leq V_l (l \in L, j \in J, k \in K) \dots \dots \dots (9)$$

$$Y_{ijke} \in \{0,1\} (i \in I, j \in J, k \in K, e \in E) \dots \dots (10)$$

$$X_{lijk} \in \{0,1\} (l \in L, i \in I, j \in J, k \in K) \dots \dots (11)$$

$$LT_{ijk} \leq \sum_{i \in I} X_{lijk} \leq UT_{ijk} \dots \dots \dots (12)$$

$$L_l \in Z^+ (l \in L) \dots \dots \dots (13)$$

$$V_l \in Z^+ (l \in L) \dots \dots \dots (14)$$

Persamaan 1 dan 2 menunjukkan *objective function* yang akan dimaksimasi, merupakan mahasiswa mengikuti ujian pada slot waktu dan hari yang tersedia untuk kemudian dilakukan penugasan ujian pada slot waktu, hari dan ruang kelas yang tersedia. Persamaan 3 memastikan mahasiswa hanya melaksanakan satu ujian pada tiap satu slot waktu. Persamaan 4 mencegah ujian yang dijadwalkan pada satu hari menggunakan slot waktu yang tersedia melebihi jumlah kelas yang tersedia. Persamaan 5 membatasi satu ruang hanya digunakan oleh satu ujian dalam satu slot waktu. Persamaan 6 penjumlahan slot waktu mahasiswa ujian dalam sehari. Persamaan 7 memastikan mahasiswa tidak mengikuti ujian yang bukan mata kuliahnya. Persamaan 8 menghitung jumlah ujian yang diikuti mahasiswa. Persamaan 9 menentukan jumlah ujian yang diikuti mahasiswa dalam sehari. Persamaan 10 dan 11 mendeklarasikan variabel keputusan sebagai bilangan biner. Persamaan 12 menentukan jumlah maksimal dan minimal ujian yang dapat diikuti mahasiswa dalam sehari. Persamaan 13 dan 14 menentukan agar nilai  $L_l$  dan  $V_l$  dan negatif.

Algoritma SA digambarkan dalam bentuk pseudocode yang terlihat sebagai berikut:

```

Algorithm
Input : baca table yang memuat jumlah dosen,
jumlah mata kuliah,
T0 temperatur awal
(1) Set T = T0 (set Li = |I| dan JKl =
|L| x |M| x |C|)
(2) Set x = solusi random
(3) Set F1 = objective function 1
for i = 0 to max_iter
tentukan Pli
kondisi persamaan 3
kondisi persamaan 7
kondisi persamaan 12
if xlijk ≥ 1 then
Solusi ditolak
else
Solusi diterima (F'i)
end for
(4) Set F2 = objective function 2
for i = 0 to max_iter
tentukan Xijlm → F'1 + LMi
kondisi persamaan 4
    
```

```

        kondisi persamaan 5
    if  $Y_{ijke} \geq 1$  then
        solusi ditolak ( $x$  dan  $F_2$ )
    else
        solusi diterima
    ( $x'$  dan  $F_2'$ )
    end for
    
```

Jadwal ujian yang baru

Model permasalahan untuk penjadwalan melibatkan beberapa parameter seperti yang terlihat pada Gambar Himpunan dalam pemodelan penjadwalan terdiri dari himpunan dosen, himpunan mata kuliah, himpunan mahasiswa, himpunan hari, himpunan slot waktu dan himpunan mata kuliah dengan dosen pengampu.

### III. HASIL DAN PEMBAHASAN

Ujian dilaksanakan selama 9 hari dengan 2 slot waktu dalam sehari, mata kuliah yang diujikan ada 18, jumlah ujian yang akan dilaksanakan sebanyak 75 kali ujian (mata kuliah dikalikan dengan kelas mahasiswa yang mengambil mata kuliah yang bersangkutan) dan jumlah ruang 5. Kelas merupakan himpunan mahasiswa seluruh angkatan.

#### 3.1. Pengujian Objective Function 1

Objective function 1, persamaan 1 berfungsi untuk penugasan ujian mahasiswa ke dalam slot waktu yang tersedia. Dalam penugasan ini memastikan mahasiswa mengikuti ujian sesuai dengan slot hari dan waktu yang tersedia. Dalam penugasan ini melibatkan empat batasan (*hard constraint*) yaitu memastikan mahasiswa hanya melaksanakan satu ujian pada tiap slot waktu,

TABEL 2. PENEMPATAN MATA KULIAH UJIAN PADA SLOT HARI DAN WAKTU

Mhs	MK		Hari dan Slot Waktu																	
			S		S		R		K		J		S		S		R		K	
Angk 1	5	7	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Angk 2	4	9																		
Angk 3	2	2																		
		18	9 x 2																	

Pada gambar 2 hasil dari penugasan mata kuliah yang diujikan yang diikuti oleh mahasiswa menghasilkan semua mata kuliah yang diujikan yang dapat dialokasikan semua dengan memenuhi tiga batasan. Batasan yang terpenuhi yaitu memastikan mahasiswa hanya melaksanakan satu ujian pada tiap satu slot waktu, persamaan 3,

persamaan 3, memastikan mahasiswa tidak mengikuti ujian yang bukan mata kuliahnya persamaan 7, jumlah minimal (0) dan jumlah maksimal (1) mahasiswa mengikuti ujian dalam sehari, persamaan 12 dan ujian dengan makul yg sama dilaksanakan pada slot waktu dan hari yang sama.

TABEL 1 PARAMETER OBJECTIVVE FUNCTION 1

Parameter	Nilai
Total Iterasi	$ L x I $
Temperatur awal $T_0$	[1,18]
Cooling factor $\alpha$	{0,1}
Temperatur akhir $T_n$	0

Objective function 1 merupakan penugasan ujian yang diikuti oleh mahasiswa ke dalam slot hari dan waktu yang tersedia. Mata kuliah yang diujikan dianggap sebagai himpunan mahasiswa dan mata kuliah. Dengan menggunakan batasan ujian dengan mata kuliah yang sama diujikan pada slot waktu dan hari yang sama, jadi ujian dengan mata kuliah yang sama dianggap sebagai satu himpunan sehingga berjumlah 18 mata kuliah yang diujikan. Dalam perhitungan ini mata kuliah yang memiliki lebih dari satu kelas dianggap satu kesatuan sehingga disini hanya menghitung semua mata kuliah yang diikuti semua angkatan yaitu sebanyak 18. Mata kuliah yang diujikan ini akan ditempatkan pada hari dan slot waktu yang tersedia (9 x 2). Terdapat 9 hari untuk pelaksanaan ujian dengan slot waktu masing-masing hari adalah 2. Vektor mata kuliah yaitu himpunan mata kuliah yang diujikan sebanyak 18 yang akan ditempatkan pada matriks hari dan slot waktu dapat dilihat pada tabel berikut.

memastikan mahasiswa tidak mengikuti ujian yang bukan mata kuliahnya, persamaan 7, dan ujian dengan makul yg sama dilaksanakan pada slot waktu dan hari yang sama, akan tetapi untuk batasan mahasiswa hanya mengikuti satu ujian dalam satu hari, persamaan 12, masih terdapat tiga angkatan kelas yang tidak memenuhi. Optimasi ini menghasilkan nilai 83,33%.

```
In [49]: runfile('F:/POLINES/KEGIATAN/LATIHAN/PYTHON/penjadwalan/of1.py', wdir='F:/POLINES/KEGIATAN/LATIHAN/PYTHON/penjadwalan')
[['2-PW' '2-BD']
 ['2-BI' '2-JK']
 ['3-PG' '1-AP']
 ['2-PB' '3-IOT']
 ['2-ARKOM' '1-MD']
 ['2-BING' '1-PTI']
 ['2-KWIN' '1-SO']
 ['1-DP' '2-KIS']
 ['1-PT' '1-DGD']]
```

Gambar 2 Hasil Pengujian

3.2. Pengujian Objective Function 2

Objective function 2 persamaan 2 berfungsi untuk melakukan plotting ujian kedalam slot waktu, hari dan ruang yang tersedia. Dalam penugasaan ini melibatkan dua batasan (hard constraint) yaitu mencegah ujian yang dijadwalkan pada satu hari menggunakan slot waktu yang tersedia melebihi jumlah kelas yang tersedia, persamaan 4 dan membatasi satu ruang hanya digunakan oleh satu ujian dalam satu slot waktu persamaan 5.

TABEL 3 PARAMETER OBJECTIVE FUNCTION 2

Parameter	Nilai
Total Iterasi	$ I x E $
Temperatur awal $T_0$	$[1,75]$
Cooling factor $\alpha$	$\{0,1\}$
Temperatur akhir $T_n$	0

Mata kuliah yang diujikan telah ditempatkan pada slot hari dan waktu pada langkah pencarian objective function 1, maka langkah selajutnya adalah penempatan pada slot ruang yang tersedia. Terdapat lima ruang yang tersedia untuk 75 mata kuliah yang diujikan. Himpunan mahasiswa pada angkatan 1 memiliki 5 kelas, angkatan 2 memiliki 4 kelas dan angkatan 3 memiliki 2 kelas, maka penempatan ujian mata kuliah yang diikuti mahasiswa dengan jumlah maksimal 5 kelas pada 5 ruang akan mendapatkan optimasi sebanyak 100%.

3.3. Analisa Metode dengan Data Sampel

Tabel menunjukkan uji coba metode pada beberapa kali percobaan. Data utama berupa data mata kuliah yang diujikan dan data mahasiswa untuk mendapatkan rata-rata jadwal ujian optimal yang bisa dilakukan.

TABEL 4. UJI COBA METODE DENGAN DATA SAMPEL

No	Objective Function 1									Objective Function 2								
	Dataset						SA			Data set						SA		
	Data		Optimasi (%)				$T_0$	$\alpha$	$T_n$	Data			Optimasi (%)		$T_0$	$\alpha$	$T_n$	
	MK Ujian + Mahasiswa Hari + waktu	Pers. 3	Pers. 7	Pers. 12	Pers. 13	Mahasiswa Ujian per Slot Waktu				Hari + slot waktu	Jumlah Ruang	Pers. 4	Pers. 5					
1.	18	9x2	100	100	83,33	100	[1,18]	{0.83}	3	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
2.	18	9x2	100	100	83,33	100	[1,18]	{0.83}	3	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
3.	18	9x2	100	100	88,88	100	[1,18]	{0.88}	2	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
4.	18	9x2	100	100	88,88	100	[1,18]	{0.88}	2	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
5.	18	9x2	100	100	77,77	100	[1,18]	{0.77}	4	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
6.	18	9x2	100	100	83,33	100	[1,18]	{0.83}	3	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
7.	18	9x2	100	100	77,77	100	[1,18]	{0.77}	4	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
8.	18	9x2	100	100	88,88	100	[1,18]	{0.88}	2	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
9.	18	9x2	100	100	83,33	100	[1,18]	{0.83}	3	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
10.	18	9x2	100	100	88,88	100	[1,18]	{0.88}	2	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
11.	18	9x2	100	100	88,88	100	[1,18]	{0.88}	2	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
12.	18	9x2	100	100	83,33	100	[1,18]	{0.83}	3	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
13.	18	9x2	100	100	88,88	100	[1,18]	{0.88}	2	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
14.	18	9x2	100	100	83,33	100	[1,18]	{0.83}	3	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
15.	18	9x2	100	100	72,22	100	[1,18]	{0.72}	5	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
16.	18	9x2	100	100	94,44	100	[1,18]	{0.94}	1	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	
17.	18	9x2	100	100	88,88	100	[1,18]	{0.88}	2	$\leq 5$	9x2	5	100	100	[1,75]	{0.1}	0	

18.	18	9 x 2	100	100	88,88	100	[1,18]	{0.88}	2	≤ 5	9 x 2	5	100	100	[1,75]	{0.1}	0
19.	18	9 x 2	100	100	83,33	100	[1,18]	{0.83}	3	≤ 5	9 x 2	5	100	100	[1,75]	{0.1}	0
20.	18	9 x 2	100	100	83,33	100	[1,18]	{0.83}	3	≤ 5	9 x 2	5	100	100	[1,75]	{0.1}	0

Dari percobaan untuk *objective function* 1 terlihat bahwa mata kuliah dapat teralokasi sebanyak 13-17 dari 18 himpunan mata kuliah yang diujikan dengan kelompok angkatan mahasiswa. Temperatur awal adalah [1,18] yang merupakan jumlah mata kuliah yang diujikan dipasangkan dengan kelompok angkatan mahasiswa. Memiliki faktor pendingin  $\alpha = 0,18$ . Dengan beberapa kali percobaan maka didapat optimasi masih terkendala pada batasan bahwa

mahasiswa hanya boleh ujian satu kali dalam sehari. Dalam percobaan ini mata kuliah yang diujikan dapat teralokasikan sebanyak 13 hingga 17 artinya pada setiap percobaan ada mahasiswa yang mengikuti ujian lebih dari satu kali dalam satu hari.

TABEL 5. ANALISA DENGAN DATA SAMPEL

No	Dataset				Optimasi SA		Kesimpulan-Solusi
	himpunan MK ujian	Hari/Waktu Periode	Jumlah kelas ujian (mahasiswa x MK)	Jumlah Ruang	Plotting dgn Batasan		
					MK Ujian + mahasiswa	Ruang	
1.	18	9 x 2	75	5	15	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
2.	18	9 x 2	75	5	15	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
3.	18	9 x 2	75	5	16	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
4.	18	9 x 2	75	5	16	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
5.	18	9 x 2	75	5	14	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
6.	18	9 x 2	75	5	15	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
7.	18	9 x 2	75	5	14	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
8.	18	9 x 2	75	5	16	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
9.	18	9 x 2	75	5	15	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
10.	18	9 x 2	75	5	16	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
11.	18	9 x 2	75	5	16	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari
12.	18	9 x 2	75	5	15	75	Mengulangi iterasi untuk mahasiswa yang mengikuti ujian lebih dari satu dalam sehari

Dari percobaan yang telah dilakukan beberapa kali percobaan maka didapat prosentase optimasi untuk jadwal kuliah memiliki nilai yang stabil yaitu 84,994%. Dengan optimasi data mata kuliah yang diujikan kedalam slot hari dan waktu. Dimana mata kuliah yang diujikan adalah jumlah mata kuliah yang diikuti oleh semua angkatan. Nilai rata-rata optimasi yang didapatkan adalah 84,994% dimana batasan ebrupa mahasiswa tidak

boleh ujian lebih dari satu dalam satu hari masih bernilai sekitar 22,22%.

Untuk *objective function* 1, hasil yang optimal didapatkan dengan memperhatikan keseimbangan jumlah mata kuliah maksimal yang boleh diikuti oleh mahasiswa dalam sehari. Solusinya adalah penambahan slot waktu dalam sehari, sehingga mahasiswa tidak ujian secara berurutan dan terdapat jeda waktu.

Tabel 6. Penyimpangan Hasil Percobaan dengan SA kuliah, maka pelanggaran terhadap batasan jumlah maksimum dosen mengampu mata kuliah akan terjadi. Untuk *objective function* 2, hasil percobaan yang didapat dapat memenuhi keseluruhan batasan yang ditentukan. Terdapat lima ruang yang tersedia untuk 75 mata kuliah yang diujikan. Himpunan mahasiswa pada angkatan 1 memiliki 5 kelas, angkatan 2 memiliki 4 kelas dan angkatan 3 memiliki 2 kelas, maka penempatan ujian mata kuliah yang diikuti mahasiswa dengan jumlah maksimal 5 kelas pada

5 ruang akan mendapatkan optimasi sebanyak 100%.

### 3.4. Analisa Penyimpangan Hasil Percobaan Metode SA

Percobaan dengan data real dengan menggunakan sebanyak 18 data mata kuliah yang diujikan 9 data hari dan 2 slot waktu menghasilkan variasi terhadap prosentase optimasi. Dari percobaan yang dilakukan sebanyak 20 kali seperti yang terlihat pada tabel 5 dilakukan perhitungan simpangan baku atau standar deviasi. Standar deviasi dari percobaan didapatkan nilai 1,0267.

TABEL 6. PENYIMPANGAN HASIL PERCOBAAN DENGAN SA

No	Percobaan ke-	Jumlah Data teroptimasi	Prosentase (X)	$X^2$
1	1	15	83,33	6943,8889
2	2	15	83,33	6943,8889
3	3	16	88,88	7899,6544
4	4	16	88,88	7899,6544
5	5	14	77,77	6048,1529
6	6	15	83,33	6943,8889
7	7	14	77,77	6048,1529
8	8	16	88,88	7899,6544
9	9	15	83,33	6943,8889
10	10	16	88,88	7899,6544
11	11	15	88,88	7899,6544
12	12	15	83,33	6943,8889
13	13	16	88,88	7899,6544
14	14	15	83,33	6943,8889
15	15	13	72,22	5215,7284
16	16	17	94,44	7899,6544
17	17	16	88,88	7899,6544
18	18	16	88,88	7899,6544
19	19	15	83,33	6943,8889
20	20	15	83,33	6943,8889
Jumlah			1.699,88	144.979,3342
Rata-rata			84,994	

$$\begin{aligned}
s &= \sqrt{\frac{\sum X^2 - \frac{(\sum X)^2}{n}}{n-1}} \\
&= \sqrt{\frac{144.979,3342 - \frac{(1.699,88)^2}{20}}{20-1}} \\
&= \sqrt{\frac{144.979,3342 - 144.479,6007}{19}} \\
&= \sqrt{\frac{499,7335}{19}} \\
&= \sqrt{26,30} \\
&= 1,0267
\end{aligned}$$

Dari hasil perhitungan didapat nilai standar deviasi sebesar 1.0267, nilai ini menunjukkan keberagaman hasil percobaan dengan beberapa kali percobaan dengan nilai berkisar 1 merupakan nilai yang tidak begitu besar sehingga dapat disimpulkan bahwa metode yang digunakan memiliki kestabilan dalam melakukan optimasi untuk menghasilkan jadwal ujian.

#### IV. KESIMPULAN

Berdasarkan hasil penelitian optimasi jadwal ujian dengan menggunakan *Simulated Annealing* dengan menggunakan data mata kuliah yang diujikan yaitu 18 dikali dengan jumlah himpunan (3 kelas angkatan) mahasiswa yang mengambil mata kuliah sebanyak 75, kelas mahasiswa sebanyak 11, data hari sebanyak 9, data waktu periode per hari sebanyak 2 dan data ruang sebanyak 2 diperoleh hasil rata-rata varian alokasi mata kuliah pada slot waktu yang tersedia adalah 84,994 dengan standar deviasi sebesar 1.0267. penambahan slot waktu dalam sehari, sehingga mahasiswa tidak ujian secara berurutan dan terdapat jeda waktu.

#### DAFTAR PUSTAKA

- [1] T. Arbaoui, J. P. Boufflet and A. Moukrim, "A matheuristic fo exam timetabling," *IFAC-PapersOnline*, vol. 49, pp. 1289-1294, 2016.
- [2] T. Arbaoui, J. P. Boufflet and A. Moukrim, "Lower bounds and compact mathematical formulations for spacing soft constraints for university examination timetabling problems," *Computers and Operations Research*, vol. 106, pp. 133-142, 2019.
- [3] O. A. Kasm, B. Mohandes, A. Diabat and S. El Khatib, ""Exam timetabling with allowable conflicts within a time window"," *Computers and Industrial Engineering*, vol. 127, pp. 263-273, 2019.
- [4] N. Leite, C. M. Fernandes, F. Melicio and A. C. Rosa, "A cellular memetic algorithm for the examination timetabling problem," *Computers and Operations Research*, vol. 94, pp. 118-138, 2018.
- [5] P. Grabusts, J. Musatovs and V. Golenkov, "The application of simulated annealing method for optional route detection between objects," *Procedia Computer Science*, vol. 149, pp. 95-101, 2018.
- [6] C. Ramesh, R. Kamalakannan, R. Karthik and C. Pavin, "A lot streaming based flow shop schedulling problem using simulated annealing algorithm," *Materials Today*, vol. xx, p. xxx, 2020.
- [7] N. Leite, F. Melicio and A. C. Rosa, "A fast simulated annealing algorithm for examination timetabling problem," *Expert Systems With Applications*, vol. 122, pp. 137-151, 2019.
- [8] Z. Li, W. Wei, K. Hu, H. Chen, Y. Wang, Q. Liu and S. Liu, "Simulated annealing wrapped generic ensemble fault diagnostic strategy for VwRF system," *Energy & Buildings*, vol. 224, pp. 110-281, 2020.
- [9] T. Konishi, H. Kojima, H. Nakagawa and T. Tsuchiya, "Using simulated annealing for locating array construction," *Information and Software Technology*, vol. 126, pp. 106-346, 2020.
- [10] E. M. Giftci and V. Ozkur, "Optimising flight connection times in airline bank structure through Simulated Annealing and Tabu Search algorithm," *Journal of Air Transport Management*, vol. 87, pp. 101-858, 2020.
- [11] S. Turk, M. Deveci, E. Ozcan, F. Catinez and R. John, "Interval type-2 fuzzy sets improved by Simulated Annealing for locating the electric charging stations," *Information Sciences*, vol. 547, pp. 641-666, 2021.