# BAB III
# METODOLOGI PENELITIAN

Bab ini membahas metodologi penelitian yang terdiri dari deskripsi masalah, tahapan penelitian, model optimasi, dan metode penyelesaiannya.

## 3.1 Deskripsi Masalah

Penelitian ini meneliti masalah penjadwalan perkuliahan di Prodi Matematika dan Prodi Pendidikan Matematika S1 FPMIPA UPI semester ganjil tahun akademik 2022-2023. Penjadwalan perkuliahan ini disusun dengan memperhatikan komponen mata kuliah, dosen pertama dan kedua, kelas, ruangan, dan waktu perkuliahan. Penyusunan penjadwalan perkuliahan di Prodi Matematika dan Prodi Pendidikan Matematika S1 FPMIPA UPI menggunakan sistem yang telah dibangun oleh universitas. Namun dari jadwal tersebut masih terdapat masalah yaitu bahwa dosen kedua tidak terdeteksi oleh sistem. Hal ini berakibat adanya bentrok pada jadwal mengajar dosen.

Penelitian ini meneliti masalah penyusunan jadwal perkuliahan di Prodi Matematika dan Prodi Pendidikan Matematika S1 FPMIPA UPI. Penyusunan jadwal tersebut akan mengikutsertakan dosen kedua mata kuliah. Dalam penelitian ini, masalah penjadwalan perkuliahan akan diselesaikan dengan menggunakan metode *Simulated Annealing*.

## 3.2 Tahapan Penelitian

Tahapan penelitian yang dilakukan adalah sebagai berikut:

1. Studi Pustaka

   Pada tahap ini dilakukan studi pustaka dengan cara mempelajari konsep serta teori-teori mengenai masalah penjadwalan perkuliahan, model *Simulated Annealing* dan algoritmanya, bersumber pada berbagai literatur baik jurnal, buku, dan karya tulis lainnya.
2. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah data yang diperoleh dari Prodi Matematika dan Prodi Pendidikan Matematika S1 FPMIPA UPI semester ganjil tahun akademik 2022-2023 dengan cara wawancara. Data tersebut terdiri dari data mata kuliah, dosen, kelas, ruangan, hari, dan slot waktu.

3. Pemodelan

   Pada tahap ini dibangun model optimisasi dari masalah penjadwalan mata kuliah dengan terlebih dahulu mendefinisikan himpunan, parameter, dan variabel keputusan dari model optimisasi serta *soft constraint* dan *hard constraint*.

4. Penyelesaian Model

   Pada tahapan ini, model matematika akan diselesaikan dengan menerapkan Algoritma *Simulated Annealing*.

5. Validasi

   Model dan penyelesaian masalah penjadwalan akan divalidasi dengan membandingkan solusi dari kasus yang sudah diketahui solusinya. Validasi dilakukan untuk memeriksa apakah model dan teknik penyelesaian ada kesalahan atau tidak. Jika diperoleh solusi yang sama maka tahapan akan dilanjutkan ke implementasi. Jika solusi berbeda maka tahapan akan diulang dari pemodelan.

6. Implementasi

   Setelah model dan teknik penyelesaian valid, selanjutnya model dan teknik penyelesaian tersebut akan diimplementasikan pada penyelesaian masalah penjadwalan di Prodi Matematika dan Prodi Pendidikan Matematika S1 FPMIPA UPI. Selanjutnya akan dianalisis kinerja dari Algoritma *Simulated Annealing* dalam penyusunan jadwal perkuliahan.

7. Penarikan Kesimpulan

   Penarikan kesimpulan akan dilakukan berdasarkan hasil implementasi.

## 3.3 Model Optimasi Masalah Penjadwalan Perkuliahan

Model optimisasi dari masalah penjadwalan di Prodi Matematika dan Prodi Pendidikan Matematika S1 FPMIPA UPI dibangun dengan menggunakan asumsi-asumsi berikut: 1. Setiap dosen bersedia mengajar pada semua waktu yang tersedia;

2. Penugasan dosen pada mata kuliah diketahui;
3. Mata kuliah yang dijadwalkan tidak termasuk mata kuliah praktikum;
4. Slot waktu dan ruangan yang tersedia cukup untuk menjadwalkan seluruh mata kuliah;
5. Satu slot waktu mewakili satu SKS.

Tahapan pertama pemodelan adalah mendefinisikan himpunan, parameter, dan variabel keputusan. Berikut adalah pendefinisian himpunan, parameter, dan variabel keputusan yang akan digunakan pada model optimisasi.

**1. Himpunan**

- I: Himpunan dosen dengan indeks $i$
- J: Himpunan kelas dengan indeks $j$
- K: Himpunan mata kuliah dengan indeks $k$
- $K_1$: Himpunan mata kuliah dengan bobot 4 SKS dengan indeks $k_1$
- L: Himpunan ruangan dengan indeks $l$
- M: Himpunan hari dengan indeks $m$
- N: Himpunan jam (slot waktu) SKS dengan indeks $n$

**2. Parameter**

$$d_{jl} = \begin{cases} 1, & \text{jika ruangan } l \text{ dapat digunakan oleh kelas } j \\ 0, & \text{yang lainnya.} \end{cases}$$

$$q_{im} = \begin{cases} 1, & \text{jika dosen } i \text{ maksimal mengajar tiga mata kuliah pada hari } m \\ 0, & \text{yang lainnya} \end{cases}$$

$$s_{im} = \begin{cases} 1, & \text{jika dosen } i \text{ maksimal mengajar dua mata kuliah pada hari } m \\ 0, & \text{yang lainnya} \end{cases}$$

$$w_{jm} = \begin{cases} 1, & \text{jika kelas } j \text{ maksimal belajar tiga mata kuliah pada hari } m \\ 0, & \text{yang lainnya} \end{cases}$$

$$z_{jm} = \begin{cases} 1, \text{jika kelas } j \text{ maksimal belajar dua mata kuliah berurutan pada} \\ \quad \text{hari } m \\ 0, \text{yang lainnya} \end{cases}$$

**3. Variabel Keputusan**

Variabel keputusan dari model optimasi masalah penjadwalan mata kuliah didefinisikan untuk mengidentifikasi apakah dosen dipasangkan pada suatu mata kuliah pada suatu kelas di suatu ruangan pada hari dan jam tertentu. Variabel keputusan tersebut didefinisikan sebagai:

$$x_{ijklmn} = \begin{cases} 1, \text{jika dosen } i \text{ mengajar kelas } j \text{ pada mata kuliah } k \\ \quad \text{di ruangan } l \text{ pada hari } m \text{ dan jam ke } n \\ 0, \text{yang lainnya.} \end{cases}$$

Kendala model optimasi masalah penjadwalan perkuliahan mewakili *hard constraint*, yaitu kendala yang harus dipenuhi atau tidak boleh dilanggar pada saat penyusunan jadwal perkuliahan. Kendala-kendala tersebut adalah:

1. Setiap dosen tidak boleh mengajar lebih dari satu kelas dalam satu mata kuliah di ruangan pada hari dan jam yang sama.

$$\sum_{j \in J} \sum_{k \in K} \sum_{l \in L} x_{ijklmn} \leq 1, \forall i \in I, m \in M, n \in N$$

2. Setiap kelas tidak boleh dijadwalkan lebih dari satu mata kuliah pada hari dan jam yang sama.

$$\sum_{i \in I} \sum_{k \in K} \sum_{I \in L} x_{ijklmn} \leq 1, \forall j \in J, m \in M, n \in N$$

3. Setiap mata kuliah diajarkan oleh minimal satu dosen.

$$\sum_{i \in I} \sum_{j \in J} \sum_{I \in L} x_{ijklmn} \geq 1, \forall k \in K, m \in M, n \in N$$

4. Mata kuliah yang berbobot empat SKS dilaksanakan dua kali pertemuan dalam satu minggu.

$$\sum_{i \in I} \sum_{I \in L} \sum_{m \in M} \sum_{n \in N} x_{ijklmn} = 2, \forall k \in K_1, j \in J,$$

5. Tidak boleh ada mata kuliah pada hari Jumat pukul 11.10-12.00.

$$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{l\in L} x_{ijklmn} = 0, untuk\ m = 5, n = 6$$

6. Setiap ruangan tidak boleh dijadwalkan lebih dari satu mata kuliah pada hari dan jam yang sama.

$$\sum_{I\in I}\sum_{j\in J}\sum_{k\in K} x_{ijklmn} \leq 1, \forall l \in L, m \in M, n \in N$$

7. Kapasitas setiap ruangan harus memenuhi banyak mahasiswa dalam kelas.

$$d_{jl} \cdot \sum_{i\in I}\sum_{k\in K}\sum_{m\in M}\sum_{n\in N} x_{ijklmn} - \sum_{i\in I}\sum_{k\in K}\sum_{m\in M}\sum_{n\in N} x_{ijklmn} = 0, \forall\, j \in J, l \in L$$

Fungsi tujuan model optimasi masalah penjadwalan perkuliahan didefinisikan untuk memenuhi *soft constraint*, yaitu kendala yang sedapat mungkin dipenuhi, tetapi masih bisa dilanggar. Kendala-kendala tersebut adalah:

1. Setiap dosen maksimal mengajar tiga mata kuliah dalam satu hari.

   Misalkan

$$p_{im} = \sum_{j\in J}\sum_{k\in K}\sum_{l\in L}\sum_{n\in N} x_{ijklmn}, untuk\ i \in I, m \in M$$

   dengan $p_{im}$ adalah variabel untuk banyak mata kuliah yang diajarkan oleh dosen *i* pada hari ke *m*. Misalkan $q_{im}$ merupakan parameter kebenaran $dosen$ $i$ paling banyak mengajar tiga mata kuliah pada hari m, dan diekspresikan sebagai

$$q_{im} = \begin{cases} 1, & \text{jika } p_{im} \leq 3 \\ 0, & \text{lainnya.} \end{cases}$$

   Maka

$$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{l\in L}\sum_{m\in M}\sum_{n\in N} q_{im} \cdot x_{ijklmn}$$

2. Setiap dosen mengajar paling banyak dua mata kuliah berurutan dalam satu hari.

   Misalkan

$$r_{im} = \sum_{j\in J} \sum_{k\in K} \sum_{l\in L} \sum_{n\in N} x_{ijklmn}, untuk\ i \in I, m \in M$$

di mana $r_{im}$ adalah variabel untuk banyak mata kuliah yang diajarkan oleh dosen $i$ pada hari ke $m$. Misalkan $s_{im}$ hanya akan bernilai benar jika dosen $i$ paling banyak mengajar dua mata kuliah berurutan pada hari $m$, maka secara logika, $s_{im}$ dapat bernilai benar apabila banyak mata kuliah yang diajarkan oleh dosen $i$ pada hari ke $m$ paling sedikit 2. Maka $s_{im}$ dituliskan sebagai berikut:

$$s_{im} = \begin{cases} 1, & jika\ r_{im} \leq 2 \\ 0, & lainnya \end{cases}$$

Dengan demikian, kendala kedua dituliskan sebagai:

$$\sum_{i\in I} \sum_{j\in J} \sum_{k\in K} \sum_{l\in L} \sum_{m\in M} \sum_{n\in N} s_{im} \cdot x_{ijklmn}$$

3. Setiap mata kuliah yang dilaksanakan dalam dua pertemuan per minggu jarak idealnya minimal dua hari.

   Misalkan

$$T = \{t \in E \mid t = 1,2,3\}$$

   Maka

$$\sum_{i\in I} \sum_{j\in J} \sum_{k\in K_1} \sum_{l\in L} \sum_{m\in M} \sum_{n\in N} x_{ijklmn} \cdot x_{ijkl(m+2)n}$$

4. Setiap kelas belajar paling banyak tiga mata kuliah dalam satu hari.

   Misalkan

$$v_{jm} = \sum_{i\in I} \sum_{k\in K} \sum_{l\in L} \sum_{n\in N} x_{ijklmn}, untuk\ j \in J, m \in M$$

   di mana $v_{jm}$ adalah variabel untuk banyak mata kuliah yang dipelajari oleh kelas $j$ pada hari ke $m$. Misalkan $w_{jm}$ merupakan parameter kebenaran kelas $i$ paling banyak belajar tiga mata kuliah pada hari $m$, maka

$$w_{jm} = \begin{cases} 1, & \text{jika } v_{jm} \leq 3 \\ 0, & \text{lainnya.} \end{cases}$$

Maka kendala ketiga dituliskan sebagai

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} \sum_{m \in M} \sum_{n \in N} w_{jm} \cdot x_{ijklmn}$$

5. Setiap kelas paling banyak belajar dua mata kuliah berurutan dalam satu hari. Misalkan

$$y_{jm} = \sum_{i \in I} \sum_{k \in K} \sum_{l \in L} \sum_{n \in N} x_{ijklmn}\,, untuk\ j \in J, m \in M$$

dengan $y_{jm}$ adalah variabel untuk banyak mata kuliah yang dipelajari oleh kelas $j$ pada hari ke $m$. Misalkan $z_{jm}$ hanya akan bernilai benar apabila kelas $i$ paling banyak belajar dua mata kuliah berturut-turut pada hari $m$, maka secara logika, $z_{jm}$ dapat bernilai benar apabila banyak mata kuliah yang dipelajari oleh kelas j pada hari ke m paling sedikit dua sehingga $z_{jm}$ dapat dituliskan sebagai berikut.

$$z_{jm} = \begin{cases} 1, & \text{jika } y_{jm} \leq 2 \\ 0, & \text{lainnya.} \end{cases}$$

Maka kendala di atas ditulis sebagai

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} \sum_{m \in M} \sum_{n \in N} z_{jm} \cdot x_{ijklmn}$$

Jadi, fungsi tujuan dari model optimisasi adalah untuk memaksimumkan pemenuhan *soft constraints*. Fungsi tujuan ini dituliskan sebagai:

Memaksimumkan:

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} \sum_{m \in M} \sum_{n \in N} (q_{im} + s_{im} + w_{jm} + z_{jm}) \cdot x_{ijklmn}$$

Adapun batasan variabel dari model optimisasi adalah sebagai berikut:

$$x_{ijklmn} \in \{0,1\}, \forall i \in \mathrm{I}, j \in J, k \in K, l \in L, m \in M, n \in N$$

Selengkapnya, model optimasi masalah penjadwalan perkuliahan di Prodi Matematika dan Prodi Pendidikan Matematika S1 FPMIPA UPI adalah sebagai berikut:

**Memaksimumkan**:

$$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{l\in L}\sum_{m\in M}\sum_{n\in N}(q_{im}+s_{im}+w_{jm}+z_{jm})\cdot x_{ijklmn}$$

$$+\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{l\in L}\sum_{m\in M}\sum_{n\in N}x_{ijklmn}\cdot x_{ijkl(m+2)n} \qquad (3.1)$$

**Terhadap:**

$$\sum_{j\in J}\sum_{k\in K}\sum_{l\in L}x_{ijklmn}\leq 1, \forall i\in I, m\in M, n\in N$$

$$\sum_{i\in I}\sum_{k\in K}\sum_{I\in L}x_{ijklmn}\leq 1, \forall j\in J, m\in M, n\in N$$

$$\sum_{i\in I}\sum_{j\in J}\sum_{I\in L}x_{ijklmn}\geq 1, \forall k\in K, m\in M, n\in N$$

$$\sum_{i\in I}\sum_{I\in L}\sum_{m\in M}\sum_{n\in N}x_{ijklmn}=2, \forall k\in K_1, j\in J,$$

$$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{l\in L}x_{ijklmn}=0, untuk\ m=5, n=6$$

$$\sum_{I\in I}\sum_{j\in J}\sum_{k\in K}x_{ijklmn}\leq 1, \forall l\in\ L, m\in M, n\in N$$

$$d_{jl}\cdot\sum_{i\in I}\sum_{k\in K}\sum_{m\in M}\sum_{n\in N}x_{ijklmn}-\sum_{i\in I}\sum_{k\in K}\sum_{m\in M}\sum_{n\in N}x_{ijklmn}=0, \forall\, j\in J, l\in L$$

$$x_{ijklmn}\in\{0,1\}, \forall i\in \mathrm{I}, j\in J, k\in K, l\in L, m\in M, n\in N.$$

Pada sub bab selanjutnya akan dibahas teknik penyelesaian dari model optimisasi di atas.

## 3.4 Penyelesaian Model menggunakan Algoritma *Simulated Annealing*

Pada penelitian ini, masalah penjadwalan mata kuliah akan diselesaikan dengan menggunakan Algoritma *Simulated Annealing*. Dalam optimasi kombinatorial, di mana ruang lingkup pencarian terlalu luas dan sulit untuk mengidentifikasi solusi yang tepat, maka diperlukan metode *Simulated Annealing*. Terdapat empat hal yang perlu diperhatikan dalam penggunaan metode *Simulated Annealing*, yaitu representasi akurat dari konfigurasi dalam suatu permasalahan, proses modifikasi terdapat elemen-elemen konfigurasi, fungsi evaluasi atau fungsi objektif yang dapat menyatakan baik atau buruknya solusi terhadap permasalahan, dan jadwal penurunan suhu dalam proses annealing dan berapa lama proses dilakukan.

Sebelum Algoritma *Simulated Annealing* diproses, pasangan mata kuliah, dosen, dan kelas harus ditentukan karena jumlah kegiatan perkuliahan ditentukan oleh pasangan mata kuliah, dosen, dan kelas. Dengan demikian, hanya ruangan, hari, dan waktu yang ditentukan menggunakan Algoritma *Simulated Annealing*. Kemudian akan digunakan Algoritma *Simulated Annealing* untuk menyelesaikan masalah penjadwalan perkuliahan.

Tahapan penyusunan jadwal mata kuliah dengan menggunakan Algoritma *Simulated Annealing* adalah sebagai berikut (Hakim & Hasibuan, 2021):

**1. Inisiasi Parameter Input**

Algoritma *Simulated Annealing* bekerja dengan dipengaruhi oleh beberapa parameter yang harus ditetapkan sebelum algoritma dijalankan. Parameter-parameter tersebut adalah sebagai berikut (Silitonga & Apdillah, 2017):

a. Temperatur awal ($T_a$)

Temperatur awal adalah penanda awal iterasi. Suhu awal perlu diatur cukup tinggi sehingga solusi yang buruk memiliki kemungkinan lebih besar untuk diterima. tetapi tidak terlalu tinggi karena akan memperpanjang waktu komputasi;

b. Temperatur akhir ($T_f$)

Temperatur akhir adalah penanda batas akhir iterasi yang sudah dihentikan. Suhu akhir harus diatur dengan tepat mendekati nol agar algoritma tidak berjalan sangat lama;

c. Faktor reduksi suhu ($\alpha$)

Faktor reduksi suhu menyatakan persentase yang digunakan untuk menurunkan suhu secara bertahap dan terkendali;

d. Angka replikasi (*Nrep*)

Angka replikasi adalah angka yang menunjukkan berapa kali *loop* yang harus dilakukan sebelum menurunkan suhu.

Adapun rentang nilai yang dapat digunakan sebagai parameter input dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Rentang nilai parameter input

| **Parameter Input** | **Rentang Nilai** |
|---|---|
| Suhu Awal ($T_\alpha$) | > 0 |
| Suhu Akhir ($T_f$) | 0 - 1 |
| Parameter reduksi suhu ($\alpha$) | 0 - 1 |
| Angka replikasi ($Nrep$) | ≥ 1 |

## 2. Pembangkitan Solusi Awal

Selanjutnya perlu dibangkitkan sebuah solusi penjadwalan awal yang fisibel. Pada penelitian ini, solusi dibangkitkan dengan pewarnaan graf yaitu pewarnaan simpul menggunakan metode *Welsh Powell*. Berikut adalah tahapan pembangkitan solusi awal.

a. Merepresentasikan Masalah Dalam Bentuk Graf

Dalam penelitian ini, graf dibangun oleh kegiatan perkuliahan sebagai simpulnya dan sisi untuk menghubungkan antara kegiatan perkuliahan yang diajarkan oleh dosen yang sama atau dipelajari oleh kelas yang sama. Pada penelitian ini, diinginkan jadwal yang tidak bentrok baik untuk dosen pertama dan dosen kedua. Oleh karena itu, dosen kedua akan dilibatkan dalam representasi graf.

Sebelum membuat representasi graf, terlebih dahulu dilakukan pelabelan pada seluruh data yang terdiri dari mata kuliah, kelas, dosen, hari, ruangan, jam. Dalam pelabelan ini, dosen kedua mata kuliah diikutsertakan. Setelah pelabelan, akan diketahui batas bawah dan batas atas untuk setiap elemen perkuliahan.

Selanjutkan akan dibentuk graf dengan terlebih dahulu mendefinisikan simpul. Simpul didefinisikan untuk merepresentasikan mata kuliah. Dua simpul dihubungkan oleh sisi dengan ketentuan jika memenuhi satu atau lebih kondisi berikut.

i. Dosen yang ditugaskan untuk dua mata kuliah tersebut sama minimal satu.
ii. Kelas yang mempelajari dua mata kuliah tersebut sama.

b. Pewarnaan Simpul

Selanjutnya dilakukan pewarnaan dengan menggunakan Algoritma *Welsh Powell*. Langkah pertama adalah mengurutkan simpul berdasarkan derajatnya. Dimulai dari simpul berderajat paling tinggi hingga paling rendah. Langkah kedua adalah pewarnaan simpul, gunakan warna satu untuk mewarnai simpul pertama. Langkah ketiga adalah periksa apakah ada simpul yang tidak bertetangga dengan simpul pertama. Jika ada, urutkan derajat simpul-simpul tersebut dan pilih simpul dengan derajat tertinggi untuk diberi warna satu. Dengan begitu sudah ada dua simpul yang diberi warna satu. Selanjutnya diperiksa simpul yang tidak bertetangga dengan kedua simpul tersebut. Lakukan kembali langkah ketiga hingga tidak ada lagi simpul yang tidak bertetangga. Jika tidak ada lagi simpul lain yang tidak bertetangga dengan simpul tersebut maka pewarnaan simpul dilanjutkan dengan warna lainnya. Ulangi penggunaan warna hingga semua simpul telah diwarnai (Sunarni, Bendi, & Alfian, 2017).

Hari dan ruangan dijadwalkan menggunakan hasil pewarnaan simpul ini sebagai panduan. Pada tahap ini mata kuliah dijadwalkan untuk hari dan ruang tertentu, dengan maksimal tiga aktivitas per hari dan per ruang. Dimulai dari simpul dengan warna satu. Simpul-simpul tersebut secara satu persatu dijadwalkan pada jam dan hari yang sama tetapi di ruangan yang berbeda. Jika seluruh ruangan pada jam dan hari tersebut sudah terpakai, simpul dijadwalkan pada hari selanjutnya. Setelah warna satu sudah terjadwalkan semua, dilanjutkan simpul-simpul warna kedua, dan begitu seterusnya sampai seluruh ruangan pada tiap hari di jam tersebut sudah terpakai. Setelah itu, penjadwalan dimulai dari hari 1 di ruangan 1 tetapi pada jam yang berbeda.

Jadwal yang dihasilkan sampai langkah ini dapat berupa solusi yang layak atau tidak. Jika diperoleh solusi yang tidak layak, maka akan dilakukan pemindahan ruangan, hari, atau waktu untuk kegiatan perkuliahan yang mengalami bentrok. Jika jadwal baru telah layak, maka jadwal tersebut dapat ditetapkan sebagai solusi awal. Selanjutnya dihitung nilai dari fungsi tujuan pada Persamaan (3.1) untuk kemudian dievaluasi.

Tahap selanjutnya akan mulai menggunakan Algoritma *Simulated Annealing*. Misalkan solusi awal adalah $s_0 = s$ dan tetapkan *set counter* $i = 1$.

3. **Pencarian Solusi Tetangga**

Pada tahapan ini akan dicari solusi tetangga ($s'$) secara acak dari solusi awal. Solusi tetangga ($s'$) merupakan solusi yang mirip atau berbeda secara signifikan dengan solusi saat ini. Pada penelitian ini, pembangkitan solusi tetangga dilakukan menggunakan Algoritma *Simple Searching Neighbourhood*. Pencarian solusi tetangga dilakukan sekali pada setiap iterasi. Pada setiap iterasi, dipilih secara acak satu kegiatan perkuliahan yang akan diubah susunannya dan satu kelompok slot waktu kosong pada suatu ruangan. Selanjutnya kelompok atom tersebut dijadwalkan pada slot kosong yang dipilih.

Setelah solusi tetangga berhasil dibangkitkan, dilakukan pemeriksaan terhadap fungsi kendala. Jika ada kendala yang dilanggar, maka solusi tetangga ini tidak akan dianggap dan pencarian akan diulang hingga solusi tetangga layak. Jika tidak ada kendala yang dilanggar, maka selanjutnya menghitung hasil pemenuhan fungsi tujuan dari solusi tetangga tersebut untuk kemudian dievaluasi (Aycan & Ayav, 2009).

**4. Perhitungan Perubahan Sementara Nilai Fungsi Tujuan**

Setelah solusi tetangga diperoleh, selanjutnya dilakukan perhitungan perubahan nilai fungsi tujuan dari solusi tetangga dan solusi awal ($\Delta f$) dengan menggunakan persamaan berikut:

$$\Delta f = f(s') - f(s)$$

**5. Pemeriksaan Nilai Fungsi Tujuan**

a. Jika $\Delta f \geq 0$, maka solusi tetangga diterima sebagai solusi baru. Lanjutkan proses ke Langkah 6.

b. Jika $\Delta f < 0$, maka hitung peluang solusi tersebut akan diterima sebagai solusi yang baru dengan menghitung $P(\Delta f) = \exp\left(-\frac{|\Delta f|}{T_a}\right)$. Selanjutnya, bangkitkan bilangan acak $r$, dengan $0 < r < 1$.

   i. Jika $P(\Delta f) \leq r$, maka solusi tetangga diterima sebagai solusi baru. Lanjutkan proses ke Langkah 6;

   ii. Jika $P(\Delta f) > r$, maka solusi awal dijadikan solusi baru dan proses Langkah 6.

**6. Penambahan *Counter***

Setelah solusi baru diperoleh, pencarian solusi tetangga dan penghitungan nilai fungsi tujuan dilakukan berulang sampai iterasi mencapai nilai *Nrep* yang telah diinisiasi di awal. Selanjutnya, tetapkan $i = i + 1$. Jika $i \leq Nrep$, kembali ke Langkah 3. Selain itu, lanjutkan proses ke Langkah 7.

**7. Jadwal pendinginan**

Jika iterasi sudah mencapai nilai *Nrep* maka dilakukan proses pendinginan dengan cara mereduksi nilai suhu saat ini ($T$) dengan menggunakan parameter reduksi ($\alpha$). Penurunan suhu mengikuti fungsi pendinginan berikut:

$$T = \alpha \cdot T_a$$

Setelah suhu terbaru diperoleh, dilakukan pencarian solusi tetangga. Pencarian ini dimulai kembali dari iterasi pertama dengan menetapkan suhu terbaru menjadi $T$.

8. **Kriteria penghentian algoritma**

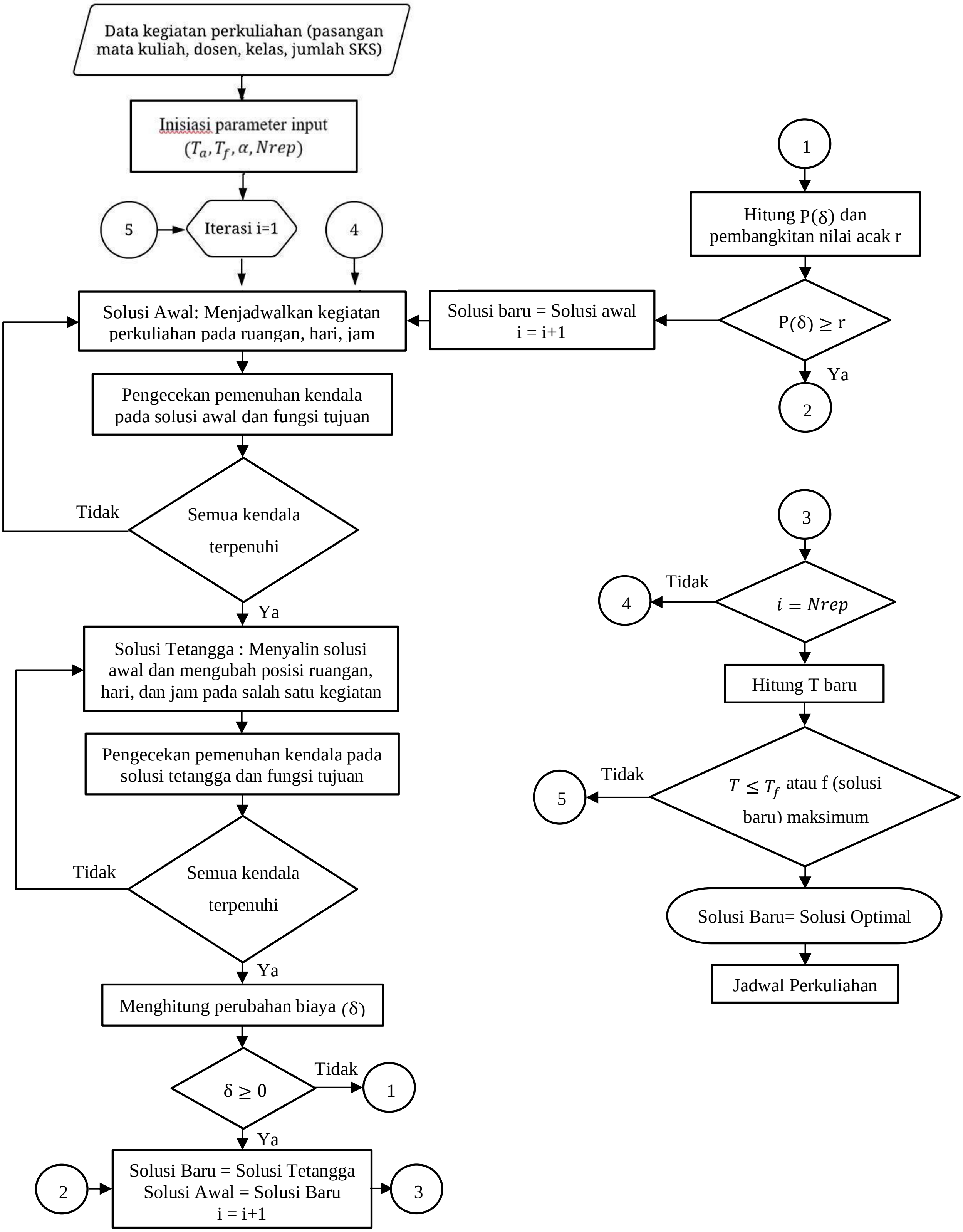
Pada penelitian ini terdapat kriteria penghentian algoritma adalah sebagai berikut:

a. Jika $T < T_f$, maka algoritma berhenti dan tampilkan $s$ sebagai solusi akhir yang optimal dan $f(s)$ sebagai nilai fungsi tujuan yang optimal;

b. Jika $T \geq T_f$ maka reset counter $i = 1$ dan kembali ke Langkah 3.

Kriteria kedua adalah ketika nilai fungsi tujuan mencapai nilai maksimum. Nilai maksimum fungsi tujuan dapat dihitung dari jumlah kegiatan perkuliahan dikali empat ditambah jumlah mata kuliah yang berbobot 4 SKS.

Proses penjadwalan perkuliahan dengan Algoritma *Simulated Annealing* secara garis besar diilustrasikan pada Gambar 4.1.




Gambar 3. 1 Ilustrasi Algoritma SA untuk penjadwalan perkuliahan

Untuk memperjelas langkah kerja Algoritma *Simulated Annealing* pada masalah penjadwalan perkuliahan, maka diberikan contoh masalah penjadwalan berikut. Pada contoh ini sistem belajar mengajar dilakukan selama 5 hari, yaitu Senin-Jumat dan terdapat 10 mata kuliah dengan 4 mata kuliah berbobot 2 SKS, 3 mata kuliah berbobot 3 SKS, dan 3 mata kuliah berbobot 4 SKS. Terdapat pula 3 kelas dan 6 dosen. Adapun ruangan yang disediakan sebanyak 2 ruangan dengan waktu perkuliahan dilaksanakan pada pukul 07.00-12.00 WIB kemudian dilanjutkan pada pukul 13.00-18.00 WIB dan 1 SKS berjalan selama 50 menit. Tabel 3.2 menunjukkan data mata kuliah bersama kelas yang belajar dan dosen yang mengajar dengan SKS setiap mata kuliah tersebut.

Tabel 3. 2 Contoh Data Perkuliahan

| **Kegiatan Perkuliahan** | **Mata Kuliah** | **Kelas** | **Dosen** | **SKS** |
|---|---|---|---|---|
| 1 | $C_1$ | $B_1$ | $A_2$ | 2 |
| 2 | $C_2$ | $B_2$ | $A_6$ | 4 |
| 3 | $C_3$ | $B_1$ | $A_1$ | 2 |
| 4 | $C_4$ | $B_2$ | $A_1$, $A_6$ | 3 |
| 5 | $C_5$ | $B_3$ | $A_4$ | 2 |
| 6 | $C_6$ | $B_3$ | $A_3$ | 4 |
| 7 | $C_7$ | $B_3$ | $A_2$ | 3 |
| 8 | $C_8$ | $B_1$ | $A_3$, $A_5$ | 3 |
| 9 | $C_9$ | $B_2$ | $A_4$, $A_2$ | 2 |
| 10 | $C_{10}$ | $B_1$ | $A_5$ | 4 |

Misalkan ditetapkan temperatur awal ($T_a$) sebesar 50, faktor reduksi suhu ($\alpha$) 0,85, temperatur akhir ($T_f$) 0,005 dan angka replikasi (*Nrep*) sebanyak 5. Langkah pertama adalah representasikan masalah dalam bentuk graf. Untuk keperluan tersebut, setiap data perlu diberi label sebelum pembangkitan solusi awal. Tabel 3.3 adalah pelabelan untuk data hari.

Tabel 3. 3 Contoh pelabelan data hari

| ID Hari | Hari |
|---|---|
| 1 | Senin |
| 2 | Selasa |
| 3 | Rabu |
| 4 | Kamis |
| 5 | Jumat |

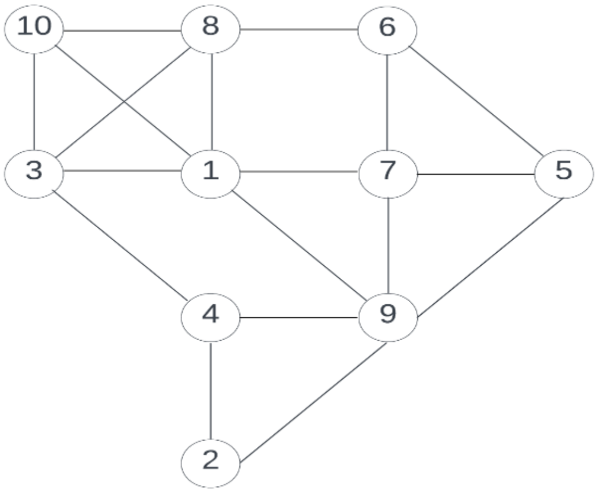
Dalam pelabelan ini, dosen kedua mata kuliah juga diikutsertakan dengan cara memberi label yang berbeda dengan label dari dosen pertama. Setelah pelabelan selesai dilakukan, maka tetapkan batas bawah dan batas atas untuk setiap komponen perkuliahan seperti terlihat pada Tabel 3.4.

Tabel 3. 4 Contoh batas atas dan batas bawah

| Elemen Perkuliahan | Batas Bawah | Batas Atas |
|---|---|---|
| Mata Kuliah | 1 | 10 |
| Dosen | 1 | 6 |
| Kelas | 1 | 3 |
| Ruangan | 1 | 2 |
| Hari | 1 | 5 |
| Jam | 1 | 9 |

Selanjutkan akan dibentuk graf dengan menggunakan nama simpul sesuai label kegiatan perkuliahan, seperti pada Gambar 3.1. Sisi dan simpul direpresentasikan sesuai dengan ketentuan yang sudah dijelaskan sebelumnya. Setelah graf terbentuk, maka dilakukan pewarnaan dengan menggunakan Algoritma *Welsh Powell*. Langkah pertama pewarnaan adalah mengurutkan simpul berdasarkan derajatnya, dimulai dari simpul dengan derajat terbesar. Urutan simpul tersebut dapat dilihat pada Tabel 3.5.

Gambar 3. 2 Contoh graf kegiatan perkuliahan




Tabel 3. 5 Contoh urutan perkuliahan berdasarkan derajat simpul

| Simpul | Mata Kuliah | Kelas | Dosen | SKS | Derajat |
|---|---|---|---|---|---|
| 1 | $C_1$ | $B_1$ | $A_2$ | 2 | 5 |
| 9 | $C_9$ | $B_2$ | $A_4$, $A_2$ | 2 | 5 |
| 3 | $C_3$ | $B_1$ | $A_1$ | 2 | 4 |
| 7 | $C_7$ | $B_3$ | $A_2$ | 3 | 4 |
| 8 | $C_8$ | $B_1$ | $A_3$, $A_5$ | 3 | 4 |
| 4 | $C_4$ | $B_2$ | $A_1$, $A_6$ | 3 | 3 |
| 5 | $C_5$ | $B_3$ | $A_4$ | 2 | 3 |
| 6 | $C_6$ | $B_3$ | $A_3$ | 4 | 3 |
| 10 | $C_{10}$ | $B_1$ | $A_5$ | 4 | 3 |
| 2 | $C_2$ | $B_2$ | $A_6$ | 4 | 2 |

Langkah selanjutnya adalah melakukan pewarnaan simpul menggunakan Algoritma *Welsh Powell* sesuai dengan urutannya, dimulai dengan simpul derajat terbesar, kemudian mewarnai simpul non-tetangga secara berurutan berdasarkan urutan derajat terbesar. Tabel 3.6 menunjukkan matriks ketetanggaan dari graf contoh data penelitian yang berguna untuk memudahkan melihat simpul-simpul yang saling bertetangga dan yang tidak.

Tabel 3. 6 Contoh matriks ajasensi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | **Derajat** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | → | 5 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | → | 2 |
| 3 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | → | 4 |
| 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | → | 3 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | → | 3 |
| 6 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | → | 3 |
| 7 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | → | 4 |
| 8 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | → | 4 |
| 9 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | → | 5 |
| 10 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | → | 3 |

Pertama dipilih simpul 1 yang berderajat 5. Kemudian diperiksa simpul yang tidak bertetangga dengan simpul 1 yaitu simpul 2, simpul 4, simpul 5, dan simpul 6. Derajat tertinggi dari keempat simpul tersebut ialah tiga yang dimiliki simpul 4, simpul 5, dan 6. Karena ketiganya sama, penulis memilih simpul 4. Selanjutnya diperiksa simpul yang tidak bertetangga dengan simpul 1 dan simpul 4. Didapatkan simpul 5 dan simpul 6 dengan derajat 3. Karena derajat simpul tersebut sama, penulis memilih simpul 5. Selanjutnya periksa apakah masih ada simpul yang tidak bertetangga dengan simpul 1, simpul 4, dan simpul 5. Karena tidak ada lagi simpul lain yang tidak bertetangga dengan simpul 1, simpul 4, dan simpul 5 maka pewarnaan simpul dilanjutkan dengan warna kedua. Tabel 3.7. menunjukkan hasil pewarnaan menggunakan Algoritma *Welsh Powell*.

Tabel 3. 7 Contoh hasil pewarnaan graf

| **Simpul** | **Derajat** | **Warna** |
|---|---|---|
| 1 | 5 | 1 |
| 9 | 5 | 2 |
| 3 | 4 | 2 |
| 7 | 4 | 3 |
| 8 | 4 | 3 |
| 4 | 3 | 1 |
| 5 | 3 | 1 |
| 6 | 3 | 2 |
| 10 | 3 | 4 |
| 2 | 2 | 3 |

Hari dan ruangan dijadwalkan menggunakan hasil pewarnaan simpul ini sebagai panduan. Pada tahap ini mata kuliah dijadwalkan untuk hari dan ruang tertentu, dengan maksimal tiga aktivitas per hari dan per ruang. Dimulai dari simpul dengan warna satu, yaitu simpul 1, simpul 4, dan simpul 5. Simpul-simpul tersebut secara satu persatu dijadwalkan pada jam dan hari yang sama tetapi di ruangan yang berbeda. Jika seluruh ruangan pada jam dan hari tersebut sudah terpakai, simpul dijadwalkan pada hari selanjutnya. Setelah warna satu sudah terjadwalkan semua, dilanjutkan simpul-simpul warna 2, dan begitu seterusnya sampai seluruh ruangan pada tiap hari di jam tersebut sudah terpakai. Setelah itu, penjadwalan dimulai dari hari 1 di ruangan 1 tetapi pada jam yang berbeda. Contoh hasil penjadwalan mata kuliah sebelum memperhatikan SKS ditunjukkan pada tabel 3.8.

Tabel 3. 8 Contoh hasil penjadwalan hari dan ruangan

| **Simpul** | **Mata Kuliah** | **Kelas** | **Dosen** | **SKS** | **Hari** | **Ruangan** | **Jam Mulai** |
|---|---|---|---|---|---|---|---|
| 1 | $C_1$ | $B_1$ | $A_2$ | 2 | 1 | 1 | 1 |
| 4 | $C_4$ | $B_2$ | $A_1$, $A_6$ | 3 | 1 | 2 | 1 |
| 5 | $C_5$ | $B_3$ | $A_4$ | 2 | 2 | 1 | 1 |
| 9 | $C_9$ | $B_2$ | $A_4$, $A_2$ | 2 | 2 | 2 | 1 |
| 3 | $C_3$ | $B_1$ | $A_1$ | 2 | 3 | 1 | 1 |
| 6 | $C_6$ | $B_3$ | $A_3$ | 4 | 3 | 2 | 1 |
| 7 | $C_7$ | $B_3$ | $A_2$ | 3 | 4 | 1 | 1 |
| 8 | $C_8$ | $B_1$ | $A_3$, $A_5$ | 3 | 4 | 2 | 1 |
| 2 | $C_2$ | $B_2$ | $A_6$ | 4 | 5 | 1 | 1 |
| 10 | $C_{10}$ | $B_1$ | $A_5$ | 4 | 5 | 2 | 1 |

Jam akhir mata kuliah kemudian ditentukan dengan melihat jumlah SKS untuk setiap mata kuliah. Untuk mata kuliah yang memuat 4 SKS, mata kuliah dibagi menjadi 2 kegiatan perkuliahan yang berbeda, masing-masing berdurasi 2 SKS dan dijadwalkan pada hari yang berbeda dengan minimal terdapat 1 selang hari. Alhasil banyak kegiatan perkuliahan bertambah sebanyak mata kuliah berbobot 4 SKS yang ditandai dengan warna biru. Jadwal yang dihasilkan setelah pemeriksaan SKS dapat dilihat pada Tabel 3.9.

Tabel 3. 9 Contoh jadwal yang dihasilkan setelah pemeriksaan SKS

| **Simpul** | **Mata Kuliah** | **Kelas** | **Dosen** | **SKS** | **Hari** | **Ruangan** | **Jam Mulai** | **Jam Akhir** |
|---|---|---|---|---|---|---|---|---|
| 1 | $C_1$ | $B_1$ | $A_2$ | 2 | 1 | 1 | 1 | 2 |
| 4 | $C_4$ | $B_2$ | $A_1$, $A_6$ | 3 | 1 | 2 | 1 | 3 |
| 5 | $C_5$ | $B_3$ | $A_4$ | 2 | 2 | 1 | 1 | 2 |
| 9 | $C_9$ | $B_2$ | $A_4$, $A_2$ | 2 | 2 | 2 | 1 | 2 |

| Simpul | Mata Kuliah | Kelas | Dosen | SKS | Hari | Ruangan | Jam Mulai | Jam Akhir |
|---|---|---|---|---|---|---|---|---|
| 3 | $C_3$ | $B_1$ | $A_1$ | 2 | 3 | 1 | 1 | 2 |
| 6 | $C_6$ | $B_3$ | $A_3$ | 4 | 3 | 2 | 1 | 2 |
| 7 | $C_7$ | $B_3$ | $A_2$ | 3 | 4 | 1 | 1 | 3 |
| 8 | $C_8$ | $B_1$ | $A_3$, $A_5$ | 3 | 4 | 2 | 1 | 3 |
| 2 | $C_2$ | $B_2$ | $A_6$ | 4 | 5 | 1 | 1 | 2 |
| 10 | $C_{10}$ | $B_1$ | $A_5$ | 4 | 5 | 2 | 1 | 2 |
| 11 | $C_6$ | $B_3$ | $A_3$ | 4 | 1 | 1 | 3 | 4 |
| 12 | $C_2$ | $B_2$ | $A_6$ | 4 | 1 | 2 | 4 | 5 |
| 13 | $C_{10}$ | $B_1$ | $A_5$ | 4 | 2 | 1 | 3 | 4 |

Selanjutnya, dilakukan pengecekan kelayakan jadwal. Jadwal pada tabel 3.9 merupakan jadwal yang sudah layak. Jika jadwal yang diperoleh belum layak, perlu dilakukan perubahan hari, ruangan atau jam dengan memperhatikan kendala. Jika ada dua kegiatan perkuliahan yang bentrok, maka salah satunya dipindahkan jadwal ruangan, hari, dan waktunya ke slot waktu kosong. Selanjutnya dihitung nilai dari fungsi tujuan untuk kemudian dievaluasi. Dengan memberi nilai 1 pada setiap pemenuhan *soft constraint*, maka diperoleh total nilai fungsi tujuan dari jadwal pada Tabel 3.9 adalah 61.

Tahap selanjutnya akan mulai menggunakan Algoritma *Simulated Annealing*. Langkah pertama dari Algoritma *Simulated Annealing* adalah mencari solusi tetangga. Pada setiap iterasi, akan dipilih secara acak satu kegiatan pekuliahan yang akan diubah dan satu slot waktu kosong pada suatu ruangan. Sebagai contoh, pada iterasi pertama dipilih kelompok atom 12 serta slot waktu kosong pada hari ke-2, ruangan ke-2, dan jam ke-3 sampai ke-4. Selanjutnya kelompok atom tersebut dijadwalkan pada slot kosong yang dipilih. Solusi tetangga yang dihasilkan terdapat pada Tabel 3.10.

Tabel 3. 10 Contoh solusi tetangga

| Simpul | Mata Kuliah | Kelas | Dosen | SKS | Hari | Ruangan | Jam Mulai | Jam Akhir |
|---|---|---|---|---|---|---|---|---|
| 1 | $C_1$ | $B_1$ | $A_2$ | 2 | 1 | 1 | 1 | 2 |
| 4 | $C_4$ | $B_2$ | $A_1$, $A_6$ | 3 | 1 | 2 | 1 | 3 |
| 5 | $C_5$ | $B_3$ | $A_4$ | 2 | 2 | 1 | 1 | 2 |
| 9 | $C_9$ | $B_2$ | $A_4$, $A_2$ | 2 | 2 | 2 | 1 | 2 |
| 3 | $C_3$ | $B_1$ | $A_1$ | 2 | 3 | 1 | 1 | 2 |
| 6 | $C_6$ | $B_3$ | $A_3$ | 4 | 3 | 2 | 1 | 2 |
| 7 | $C_7$ | $B_3$ | $A_2$ | 3 | 4 | 1 | 1 | 3 |
| 8 | $C_8$ | $B_1$ | $A_3$, $A_5$ | 3 | 4 | 2 | 1 | 3 |
| 2 | $C_2$ | $B_2$ | $A_6$ | 4 | 5 | 1 | 1 | 2 |
| 10 | $C_{10}$ | $B_1$ | $A_5$ | 4 | 5 | 2 | 1 | 2 |
| 11 | $C_6$ | $B_3$ | $A_3$ | 4 | 1 | 1 | 3 | 4 |
| 12 | $C_2$ | $B_2$ | $A_6$ | 4 | 2 | 2 | 3 | 4 |
| 13 | $C_{10}$ | $B_1$ | $A_5$ | 4 | 2 | 1 | 3 | 4 |

Solusi tetangga yang diperoleh sudah tidak melanggar kendala sehingga setelah iterasi pertama tidak perlu dilakukan pencarian solusi tetangga lagi. Selanjutnya, solusi tetangga yang baru tersebut akan dihitung pemenuhan fungsi tujuannya untuk kemudian dievaluasi.

Langkah ke dua adalah menghitung perubahan sementara nilai fungsi tujuan dari solusi tetangga dan solusi awal dengan menggunakan rumus:

$$\Delta f = f(s') - f(s) = 61 - 61 = 0$$

Karena $\Delta f$ bernilai 0, maka solusi tetangga diterima sebagai solusi baru.

Setelah solusi baru diperoleh, pencarian solusi tetangga dan penghitungan nilai fungsi tujuan dilakukan berulang sampai iterasi mencapai nilai *Nrep* yang telah diinisiasi di awal. Misal iterasi sudah mencapai *Nrep*, yaitu iterasi ke-10. Maka selanjutnya dilakukan proses pendinginan suhu mengikuti persamaan berikut:

$$T = \alpha \cdot T_\alpha = 0{,}85 \cdot 50 = 42{,}5$$

Proses di atas dilakukan berulang sampai suhu saat ini bernilai kurang dari suhu aktir $T_f$. Tabel 3.11 merupakan solusi baru hasil perulangan pertama.

Tabel 3. 11 Contoh solusi baru hasil iterasi pertama

| **Simpul** | **Mata Kuliah** | **Kelas** | **Dosen** | **SKS** | **Hari** | **Ruangan** | **Jam Mulai** | **Jam Akhir** |
|---|---|---|---|---|---|---|---|---|
| 1 | $C_1$ | $B_1$ | $A_2$ | 2 | 1 | 1 | 1 | 2 |
| 2 | $C_2$ | $B_2$ | $A_6$ | 4 | 5 | 1 | 1 | 2 |
| 3 | $C_3$ | $B_1$ | $A_1$ | 2 | 3 | 1 | 1 | 2 |
| 4 | $C_4$ | $B_2$ | $A_1$, $A_6$ | 3 | 1 | 2 | 1 | 3 |
| 5 | $C_5$ | $B_3$ | $A_4$ | 2 | 2 | 1 | 1 | 2 |
| 6 | $C_6$ | $B_3$ | $A_3$ | 4 | 3 | 2 | 1 | 2 |
| 7 | $C_7$ | $B_3$ | $A_2$ | 3 | 4 | 1 | 1 | 3 |
| 8 | $C_8$ | $B_1$ | $A_3$, $A_5$ | 3 | 4 | 2 | 1 | 3 |
| 9 | $C_9$ | $B_2$ | $A_4$, $A_2$ | 2 | 2 | 2 | 1 | 2 |
| 10 | $C_{10}$ | $B_1$ | $A_5$ | 4 | 5 | 2 | 1 | 2 |
| 11 | $C_6$ | $B_3$ | $A_3$ | 4 | 1 | 1 | 3 | 4 |
| 12 | $C_2$ | $B_2$ | $A_6$ | 4 | 2 | 2 | 3 | 4 |
| 13 | $C_{10}$ | $B_1$ | $A_5$ | 4 | 2 | 1 | 3 | 4 |

Dalam kasus ini nilai maksimum dari fungsi tujuan adalah 61. Ketika nilai fungsi tujuan dari solusi sudah mencapai 61, saat itu pula algoritma dihentikan karena telah memenuhi kriteria kedua dari penghentian Algoritma *Simulated Annealing*.