

## BAB 3

### METODOLOGI PENELITIAN

Bab ini membahas metodologi yang digunakan untuk menyelesaikan masalah penjadwalan kereta api dimulai dengan mengidentifikasi masalah kemudian dilanjutkan pembuatan model optimisasi dari masalah penjadwalan kereta api dengan mendefinisikan himpunan, parameter, dan variabel keputusan model sebelumnya. Kemudian model optimisasi diselesaikan dengan teknik *Simulated Annealing*.

#### 3.1 Deskripsi Masalah

Penjadwalan kereta api merupakan suatu kegiatan pembentukan jadwal kedatangan dan keberangkatan kereta api di suatu stasiun dalam suatu perjalanan. Kegiatan ini dilakukan dengan mempertimbangkan banyaknya kereta api, stasiun, petak blok yang dapat digunakan, kecepatan rata-rata minimum kereta, serta waktu *delay* kereta. Penjadwalan kereta api akan dilakukan untuk mengatur jadwal perjalanan selama satu hari. Model akan diimplementasikan untuk penyusunan jadwal perjalanan kereta api dengan rute Bandung – Padalarang. Rute ini memiliki enam stasiun, yaitu: Bandung (BD), Ciroyom (CIR), Andir (AND) merupakan sinyal, Cimindi (CMD), Cimahi (CMI), Gadobangkong (GK), dan Padalarang (PDL).

Stasiun Bandung memiliki enam jalur, dan stasiun Padalarang memiliki empat jalur. Hanya stasiun Bandung yang memiliki depo. Stasiun di antara Bandung dan Padalarang beserta enam petak blok yang menghubungkannya memiliki dua jalur kecuali di Cimahi memiliki tiga jalur. Kereta berhenti di semua stasiun kecuali stasiun Andir (AND) karena stasiun ini merupakan sinyal. Kecepatan kereta diperhitungkan berdasarkan jarak yang harus ditempuh pada setiap petak blok. Terdapat 36 kereta api yang akan dijadwalkan yang terdiri dari 17 kereta api *outbound* dan 19 kereta api *inbound*. Penelitian ini akan menerapkan algoritma *Simulated Annealing* untuk mengonstruksi jadwal perjalanan kereta api.

Penjadwalan keberangkatan dan kedatangan Kereta Api akan diselesaikan dengan pendekatan masalah *job-shop* dengan memisalkan  $n$  buah perjalanan kereta

api yang dapat dinotasikan sebagai  $J_1, J_2, J_3, \dots, J_n$  yang harus dijadwalkan pada  $l$  buah rute. Sebuah perjalanan  $J_i$  melewati suatu rute yang terdiri atas  $q$  buah stasiun

dan  $m$  buah petak blok. Dengan menggunakan pendekatan masalah *job-shop*, perjalanan  $J_i$  dapat dipandang sebagai pekerjaan di mana petak blok yang ada diasumsikan sebagai operasi. Dengan demikian, terdapat  $m$  buah operasi yang dinotasikan sebagai  $o_{i1}, o_{i2}, o_{i3}, \dots, o_{im}$ . Setiap operasi yang dilakukan dalam perjalanan  $J_i$  tersebut menggunakan tepat satu sumber daya berupa satu petak blok pada rute yang dilalui, yaitu operasi  $o_{ik}$ .

### 3.2 Tahapan Penelitian

Tahapan penelitian yang dilakukan sebagai berikut:

1. Studi Pustaka

Pada tahap ini dilakukan studi pustaka dengan cara mempelajari konsep serta teori-teori mengenai masalah penjadwalan kereta api jalur ganda, model *Simulated Annealing* dan algoritmanya, bersumber dari berbagai literatur baik jurnal, buku, dan karya tulis lainnya.

2. Pengumpulan Data

Data yang diperlukan pada penelitian kali ini terdiri dari data kereta api dalam satu perjalanan, kecepatan kereta, banyaknya stasiun, waktu kedatangan dan keberangkatan kereta pada perjalanan *outbound* dan *inbound*, serta banyaknya petak blok. Data tersebut diperoleh dari GAPEKA atau Grafik Perjalanan Kereta Api tahun 2023 yang dapat diakses pada *website*: <https://commuterline.id/informasi-publik/berita/gapeka-2023-kai-commuter-lakukan-penyesuaian-operasional-pelayanan-commuter-line-di-wilayah-2-bandung-mulai-1-juni-2023> sedangkan data yang terdiri atas panjang petak blok serta keterangan jalur ganda atau tidaknya pada petak blok tersebut diperoleh dari *website*: <https://www.arcgis.com/apps/View/index.html?appid=27b7119dc6754d3e9e584a4fa71e5744>

3. Pemodelan

Pada tahap ini dibangun model optimisasi dari masalah penjadwalan Kereta Api dengan mendefinisikan himpunan, parameter dan variabel keputusan dari model optimisasi serta *hard constraint*.

#### 4. Penyelesaian Model

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

#### 5. Validasi

Model dan penyelesaian masalah akan divalidasi dengan menggunakan contoh jadwal fiktif dengan membandingkan solusi optimal yang dihasilkan oleh program yang dibuat menggunakan Python dengan solusi optimal hasil perhitungan manual. 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 kereta api rute Bandung – Padalarang, kemudian akan dianalisis kinerja dari Algoritma *Simulated Annealing* dalam penyusunan jadwal kedatangan dan keberangkatan Kereta Api.

#### 7. Penarikan Kesimpulan

Kesimpulan akan diperoleh dengan mengacu pada hasil yang diperoleh dari implementasi yang telah dilakukan.

### 3.3 Model Optimisasi Penjadwalan Kereta Api

Model optimisasi masalah penjadwalan kereta api dibangun dengan asumsi berikut:

1. Stasiun pemberhentian terbatas pada Bandung, Ciroyom, Cimindi, Cimahi, Gadobangkong, dan Padalarang.
2. Model dibangun untuk kasus kereta api jalur ganda.
3. Satuan waktu terkecil yang digunakan dalam penjadwalan adalah menit.
4. Jadwal perjalanan kereta api yang akan ditentukan adalah jadwal perjalanan untuk satu hari.

5. Kereta api yang digunakan dalam penelitian ini adalah semua kereta yang keluar dari depo Bandung dan berakhir di Padalarang serta semua kereta dari arah sebaliknya yaitu Padalarang ke Bandung.
6. Kereta api yang digunakan dalam penelitian ini adalah kereta api jenis regular atau ekonomi tanpa memperhatikan kereta api lain dengan jalur yang melewati Bandung - Padalarang.
7. Tidak ada urutan prioritas kereta api.

Pada penelitian ini, masalah penjadwalan kereta api dipandang sebagai kasus khusus dari masalah penjadwalan *job-shop*. Misalkan  $n$  buah perjalanan kereta api yang harus dijadwalkan pada  $l$  buah rute. Setiap perjalanan melewati suatu rute yang terdiri atas  $q$  buah stasiun dan  $m$  buah petak blok. Maka pekerjaan dapat merepresentasikan setiap satu perjalanan kereta api yang memiliki  $m$  buah operasi dan setiap operasi yang dilakukan pada perjalanan menggunakan tepat satu sumber daya berupa satu petak blok pada rute yang dilalui.

Tahapan pertama pemodelan adalah mendefinisikan himpunan yang dibutuhkan dalam pemodelan masalah penjadwalan kereta api. Pemodelan masalah penjadwalan kereta api ini merujuk pada model oleh Higgins, dkk. (1996) dalam Dwiyatcita (2012). Adapun himpunan yang digunakan adalah sebagai berikut:

- $S$  = himpunan stasiun,  $S = \{1, 2, 3, \dots, q\}$
- $J$  = himpunan kereta api,  $J = \{1, 2, 3, \dots, n\}$
- $K$  = himpunan petak blok,  $K = \{1, 2, 3, \dots, m\}$

Adapun parameter yang akan digunakan sebagai berikut:

- $J_i$  = perjalanan kereta api  $i$  ( $i = 1, 2, 3, \dots, n$ ) (pekerjaan)
- $o_{ik}$  = operasi di petak blok  $k$  ( $k = 1, 2, 3, \dots, m$ ) (sumber daya) dari kereta api  $i$
- $n$  = banyaknya kereta api
- $m$  = banyaknya petak blok
- $q$  = banyaknya stasiun

- $h = \text{time headway}$
- $d_k = \text{panjang petak blok } k$
- $\underline{v}_{ik} = \text{kecepatan rata - rata minimum kereta api } i \text{ di petak blok ke } - k$
- $\bar{v}_{ik} = \text{kecepatan rata - rata maksimum kereta api } i \text{ di petak blok ke } - k$
- $p_{is} = \text{lama waktu berhenti kereta api } i \text{ di stasiun } s$
- $\tau_{is} = \text{waktu delay kereta api } i \text{ di stasiun } s$
- $M = \text{bilangan bulat positif besar}$

$$A_{ijk} = \begin{cases} 1, & \text{jika kereta api } \textit{outbound} i, \text{ dengan } i \leq r \text{ menggunakan petak} \\ & \text{blok } k \text{ sebelum kereta api } \textit{outbound} j \text{ dengan } j \leq r, \\ 0, & \text{yang lainnya.} \end{cases}$$

$$B_{ijk} = \begin{cases} 1, & \text{jika kereta api } \textit{inbound} i, \text{ dengan } i > r \text{ menggunakan} \\ & \text{petak blok } k \text{ sebelum kereta api } \textit{inbound} j \text{ dengan } j > r \\ 0, & \text{yang lainnya.} \end{cases}$$

Variabel keputusan dari model optimisasi didefinisikan untuk menjadwalkan jam keberangkatan dan kedatangan kereta api. Variabel ini didefinisikan sebagai sebagai berikut:

$X_{ias} = \text{waktu kedatangan kereta api } i \text{ di stasiun } s$

$X_{ids} = \text{waktu keberangkatan kereta api } i \text{ dari stasiun } s$

$C_{maks} = \text{waktu tempuh maksimum}$

Kendala dari model optimisasi masalah penjadwalan kereta api mewakili *hard constraint*, yaitu kendala yang harus dipenuhi pada saat penyusunan jadwal kereta api. Kendala-kendala tersebut adalah sebagai berikut:

### 1. Kendala Urutan Operasi

Urutan operasi dikembangkan dari konsep masalah penjadwalan *job-shop*, yaitu ketika operasi ke-(k+1) pada pekerjaan  $J_i$  hanya bisa dimulai setelah operasi ke-k telah selesai dikerjakan. Waktu operasi dimulai  $o_{i(k+1)}$ , yaitu  $X_{ids}$ , harus lebih dari atau sama dengan waktu dimulainya operasi  $o_{ik}$ . yaitu  $X_{ias}$ ,

ditambah lama waktu penundaan dari suatu perjalanan kereta api ditambah lama waktu pemrosesannya, yaitu  $p_{is}$ . Selanjutnya, terdapat variabel *delay* ( $\tau_{is}$ ) yang berarti lama waktu penundaan dari suatu perjalanan kereta api  $i$  di stasiun  $s$  untuk menghindari konflik. Waktu tiba kereta api di stasiun pertama merupakan waktu tiba kereta api yang keluar dari depo atau waktu kembali dari stasiun tujuan akhir ke stasiun asal. Kendala urutan operasi dituliskan sebagai berikut:

$$X_{ias} + p_{is} + \tau_{is} = X_{ids}, \forall i \leq r, s = 1, 2, 3, \dots, q \quad (3.1)$$

$$X_{ias} + p_{is} + \tau_{is} = X_{ids}, \forall i > r, s = q, q - 1, \dots, 1. \quad (3.2)$$

## 2. Kendala Aturan Penyusunan

Misalkan terdapat kereta api  $i$  dan  $j$  dengan arah yang sama akan menggunakan petak blok ke- $k$  secara bersamaan, sehingga operasi  $o_{ik}$  dan  $o_{jk}$  akan diproses pada waktu yang sama. Terdapat dua langkah yang dapat dilakukan agar tidak dapat terjadi konflik. Kedua langkah tersebut adalah dengan mendahulukan perjalanan kereta api  $J_j$  atau mendahulukan perjalanan kereta api  $J_i$ . Oleh karena itu, kendala di atas dikalikan dengan  $M$ , yaitu bilangan positif besar yang digunakan khusus pada kendala untuk memilih salah satu yaitu pengalihan dengan bilangan  $M$  terdapat pada Kendala (3.3) sampai Kendala (3.10).

Penyusunan untuk kereta api jenis *outbound* didefinisikan pada kendala (3.3) sampai (3.6). Kendala (3.3) dan (3.4) digunakan apabila nilai  $A_{ijk} = 0$ , yaitu perjalanan kereta api ketika  $J_j$  didahulukan, sehingga kereta api  $j$  tiba lebih awal dari kereta api  $i$  di stasiun berikutnya. Nilai  $h$  juga ditambahkan agar terdapat jarak antarmkereta api ketika keluar dan masuk stasiun. Kendala (3.5) dan Kendala (3.6) bekerja sama seperti Kendala (3.3) dan Kendala (3.4) tetapi dengan nilai  $A_{ijk} = 1$ , yang berarti kereta api  $i$  berangkat lebih dulu dari  $j$

$$MA_{ijk} + X_{ia(s+1)} \geq X_{ja(s+1)} + h, \quad (3.3)$$

$$i \neq j; s = 1, 2, 3, \dots, q - 1; k = 1, 2, 3, \dots, m.$$

$$MA_{ijk} + X_{ids} \geq X_{jds} + h, \quad (3.4)$$

$$i \neq j; s = 1, 2, 3, \dots, q - 1; k = 1, 2, 3, \dots, m.$$

$$M(1 - A_{ijk}) + X_{ja(s+1)} \geq X_{ia(s+1)} + h, \quad (3.5)$$

$$i \neq j; s = 1, 2, 3, \dots, q - 1; k = 1, 2, 3, \dots, m.$$

$$M(1 - A_{ijk}) + X_{jds} \geq X_{ids} + h, \quad (3.6)$$

$$i \neq j; s = 1, 2, 3, \dots, q - 1; k = 1, 2, 3, \dots, m.$$

Penyusunan pada kereta api *inbound* dapat dijelaskan dengan cara yang sama seperti pada kereta api *outbound* kendala aturan penyusunan pada kereta api *inbound* diberikan pada Kendala (3.7) sampai Kendala (3.10).

$$MB_{ijk} + X_{ias} \geq X_{jas} + h, \quad (3.7)$$

$$i \neq j; s = q - 1, q - 2, \dots, 1; k = m, m - 1, m - 2, \dots, 1.$$

$$MB_{ijk} + X_{ia(s+1)} \geq X_{ja(s+1)} + h, \quad (3.8)$$

$$i \neq j; s = 1, 2, 3, \dots, q - 1; k = 1, 2, 3, \dots, m.$$

$$M(1 - B_{ijk}) + X_{jas} \geq X_{ias} + h, \quad (3.9)$$

$$i \neq j; s = q - 1, q - 2, \dots, 1; k = m, m - 1, m - 2, \dots, 1.$$

$$M(1 - B_{ijk}) + X_{jd(s+1)} \geq X_{id(s+1)} + h, \quad (3.10)$$

$$i \neq j; s = q - 1, q - 2, \dots, 1; k = m, m - 1, m - 2, \dots, 1.$$

### 3. Kendala Aturan Lama Waktu Beroperasi

Waktu penggunaan sumber daya pada masalah penjadwalan *job-shop* secara umum dijadikan suatu input. Waktu tersebut pada masalah penjadwalan kereta api sama dengan jarak tempuh dibagi dengan kecepatan rata-ratanya. Waktu rata-rata minimum dan maksimum penggunaan suatu petak blok diberikan pada Kendala (3.10) untuk kereta api *outbound* dan Kendala (3.11) untuk kereta api *inbound*.

$$\frac{d_k}{v_{ik}} \leq X_{ia(s+1)} - X_{ids} \leq \frac{d_k}{v_{ik}}, \quad (3.11)$$

$$i = 1, 2, 3, \dots, r; k = 1, 2, 3, \dots, m; s = 1, 2, 3, \dots, q - 1$$



$$\frac{d_k}{\bar{v}_{ik}} \leq X_{ias} - X_{id(s+1)} \leq \frac{d_k}{\underline{v}_{ik}}, \quad (3.12)$$

$$i = 1,2,3, \dots, r; k = 1,2,3, \dots, m; s = 1,2,3, \dots, q - 1$$

#### 4. Kendala Stasiun Pemberhentian

Jika terdapat kereta api yang berhenti di stasiun-stasiun tertentu, maka terdapat kendala yang ditambahkan khusus untuk kereta api tersebut. Jika kereta api tidak berhenti di stasiun ke- $s$ , maka waktu kedatangan dan keberangkatan kereta api tersebut di stasiun ke- $s$  adalah sama. Selain itu, sebagai input, waktu tunggu di stasiun tersebut bernilai nol. Kendala ini dituliskan sebagai berikut:

$$X_{ias} = X_{ids}, i \in J \text{ dan } s \in S \quad (3.13)$$

Adapun batasan variabel dari model adalah sebagai berikut:

$$h, p_{is}, X_{ias}, X_{ids} \geq 0 \quad (3.14)$$

$$A_{ijk}, B_{ijk} \in \{0,1\}. \quad (3.15)$$

Fungsi tujuan dari model optimisasi adalah untuk meminimumkan waktu tempuh maksimum dari kereta api. Waktu tempuh dihitung berdasarkan selisih antara waktu kedatangan di stasiun pertama dan waktu keberangkatan dari stasiun akhir, lalu kembali ke stasiun awal atau masuk ke dalam depo di mana kereta api mendapatkan perawatan dan perbaikan mesin. Kereta api dari 1 sampai  $r$  (*outbound*) berakhir di stasiun  $q$ . Sedangkan Kereta Api  $r + 1$  sampai  $n$  (*inbound*) berakhir di Stasiun 1. Fungsi tujuan diekspresikan sebagai berikut:

**Meminimumkan:**

$$C_{maks} = \sum_{i=1}^r (X_{idq} - X_{ia1}) + \sum_{i=r+1}^n (X_{id1} - X_{iaq})$$

Selengkapnya, model optimisasi masalah penjadwalan kereta api jalur ganda dengan rute Bandung – Padalarang adalah sebagai berikut:

**Meminimumkan:**

$$C_{maks} = \sum_{i=1}^r (X_{idq} - X_{ia1}) + \sum_{i=r+1}^n (X_{id1} - X_{iaq})$$

terhadap:

$$X_{ias} + p_{is} + \tau_{is} = X_{ids}, i \leq r, s = 1, 2, 3, \dots, q.$$

$$X_{ias} + p_{is} + \tau_{is} = X_{ids}, i > r, s = q, q - 1, \dots, 1.$$

$$MA_{ijk} + X_{ia(s+1)} \geq X_{ja(s+1)} + h,$$

$$i \neq j; s = 1, 2, 3, \dots, q - 1; k = 1, 2, 3, \dots, m.$$

$$MA_{ijk} + X_{ids} \geq X_{jds} + h,$$

$$i \neq j; s = 1, 2, 3, \dots, q - 1; k = 1, 2, 3, \dots, m.$$

$$M(1 - A_{ijk}) + X_{ja(s+1)} \geq X_{ia(s+1)} + h,$$

$$i \neq j; s = 1, 2, 3, \dots, q - 1; k = 1, 2, 3, \dots, m.$$

$$M(1 - A_{ijk}) + X_{jds} \geq X_{ids} + h,$$

$$i \neq j; s = 1, 2, 3, \dots, q - 1; k = 1, 2, 3, \dots, m.$$

$$MB_{ijk} + X_{ias} \geq X_{jas} + h,$$

$$i \neq j; s = q - 1, q - 2, \dots, 1; k = m, m - 1, m - 2, \dots, 1.$$

$$MB_{ijk} + X_{ia(s+1)} \geq X_{ja(s+1)} + h,$$

$$i \neq j; s = q - 1, q - 2, \dots, 1; k = m, m - 1, m - 2, \dots, 1.$$

$$M(1 - B_{ijk}) + X_{jas} \geq X_{ias} + h,$$

$$i \neq j; s = q - 1, q - 2, \dots, 1; k = m, m - 1, m - 2, \dots, 1.$$

$$M(1 - B_{ijk}) + X_{jd(s+1)} \geq X_{id(s+1)} + h,$$

$$i \neq j; s = q - 1, q - 2, \dots, 1; k = m, m - 1, m - 2, \dots, 1.$$

$$\frac{d_k}{v_{ik}} \leq X_{ia(s+1)} - X_{ids} \leq \frac{d_k}{\underline{v}_{ik}},$$

$$i = 1, 2, 3, \dots, r; k = 1, 2, 3, \dots, m; s = 1, 2, 3, \dots, q - 1$$

$$\frac{d_k}{v_{ik}} \leq X_{ias} - X_{id(s+1)} \leq \frac{d_k}{\underline{v}_{ik}},$$

$$i = 1, 2, 3, \dots, r; k = 1, 2, 3, \dots, m; s = 1, 2, 3, \dots, q - 1$$

$$X_{ias} = X_{ids}, i \in J \text{ dan } s \in S$$

$$h, p_{is}, X_{ias}, X_{ids} \geq 0$$

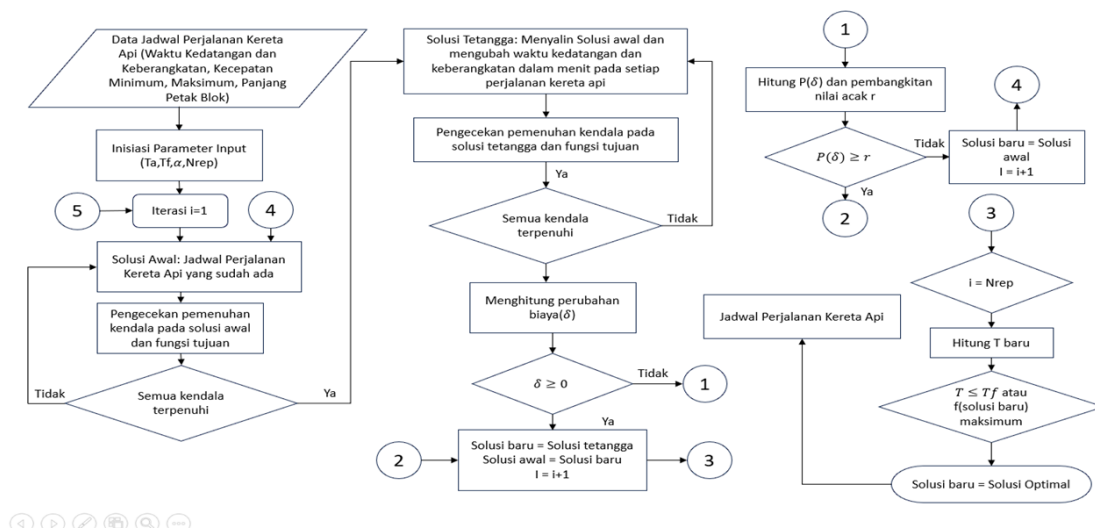
$$A_{ijk}, B_{ijk} \in \{0, 1\}, \forall i, j \in J, k \in K$$

Model optimisasi diatas termasuk ke dalam kategori model *mixed integer programming*. Pada subbab selanjutnya akan dibahas teknik penyelesaian model dengan menggunakan Simulated Annealing.

### 3.4 Algoritma Simulated Annealing untuk Penjadwalan Kereta Api

Dalam penelitian ini akan digunakan Algoritma *Simulated Annealing* untuk menyelesaikan masalah penjadwalan kereta api rute Bandung – Padalarang. Berdasarkan pada simulasi yang dilakukan Metropolis (Dowsland &Thompson, 2012). Tujuan dari penjadwalan pada penelitian ini adalah meminimumkan waktu tempuh maksimum atau  $C_{maks}$ . Oleh karena itu, tujuan dari penggunaan Simulated Annealing adalah untuk meminimumkan  $C_{maks}$  dengan memenuhi kendala-kendala yang telah dibahas sebelumnya sehingga diperoleh jadwal baru untuk kereta api untuk satu hari dengan rincian waktu kedatangan dan keberangkatan setiap kereta api pada setiap stasiun.

Selanjutnya akan dijelaskan tahapan Algoritma *Simulated Annealing* pada masalah penjadwalan kereta api menggunakan ilustrasi pada diagram alir seperti pada Gambar 3.1.



Gambar 3.1 Diagram Alir Proses Penjadwalan Kereta Api.

Untuk menjelaskan bagaimana Algoritma *Simulated Annealing* diterapkan pada masalah penjadwalan kereta api akan digunakan contoh kasus kecil masalah penjadwalan kereta api. Misalkan terdapat lima kereta api, tiga stasiun

pemberhentian, dua petak blok, data panjang petak blok, kecepatan kereta api yang dihitung berdasarkan jarak yang harus ditempuh pada setiap petak blok, serta waktu kedatangan dan waktu keberangkatan yang dibangkitkan secara hipotetik yang nantinya akan dioptimalkan menggunakan Algoritma Simulated Annealing. Berikut merupakan tahapan Penerapan Algoritma *Simulated Annealing* pada masalah penjadwalan kereta api.

### 3.4.1. Inisiasi Parameter Input

Dalam langkah awal dari proses Simulated Annealing, diperlukan penetapan nilai parameter input. Beberapa parameter input yang diterapkan dalam Algoritma Simulated Annealing melibatkan elemen-elemen berikut (Silitonga & Apdillah, 2017, dalam Wahyuni, 2022):

- $T_a$  atau suhu awal, yang berfungsi sebagai penanda awal iterasi.
- $T_f$  atau suhu akhir, yang menandakan batas akhir iterasi untuk menghentikan proses iterasi.
- $\alpha$  atau parameter reduksi suhu, merupakan angka untuk menurunkan suhu secara perlahan.
- Nrep atau angka replikasi, yang menjadi batas pengulangan loop yang harus dilakukan sebelum suhu diturunkan.

Hakim dan Hasibuan (2021) menyarankan bahwa suhu awal harus diatur pada tingkat yang cukup tinggi agar memungkinkan pergerakan atom yang bebas tetapi tidak boleh terlalu tinggi karena akan memakan waktu yang cukup lama. Kemudian suhu akhir harus diatur dengan tepat mendekati nol agar algoritma tidak berjalan sangat lama. Tabel 3.1 menunjukkan rentang nilai yang dapat digunakan sebagai parameter input.

Tabel 3.1 Rentang nilai parameter input

| Parameter Input                     | Rentang Nilai |
|-------------------------------------|---------------|
| Suhu awal ( $T_a$ )                 | $> 0$         |
| Suhu akhir ( $T_f$ )                | (0,1)         |
| Parameter reduksi suhu ( $\alpha$ ) | (0,1)         |
| Angka replikasi (Nrep)              | $\geq 1$      |

### 3.4.2. Inisiasi Solusi Awal

Secara teori, pemilihan solusi awal dianggap tidak memiliki pengaruh terhadap kualitas solusi akhir. Solusi cenderung konvergen secara menyeluruh dan bersifat independen terhadap solusi awal. Namun, penggunaan metode heuristik yang optimal dalam pembangkitan solusi awal dapat secara signifikan mempercepat waktu komputasi untuk mencapai solusi akhir. Dalam penelitian ini, solusi awal menggunakan jadwal yang telah ditetapkan sebelumnya. Misalkan diberikan jadwal kereta api seperti tertera pada Tabel 3.2 dan Tabel 3.3. Kedua jadwal tersebut selanjutnya akan digunakan sebagai solusi awal.

Tabel 3.2 Contoh jadwal kereta api *outbound* yang akan dioptimalkan

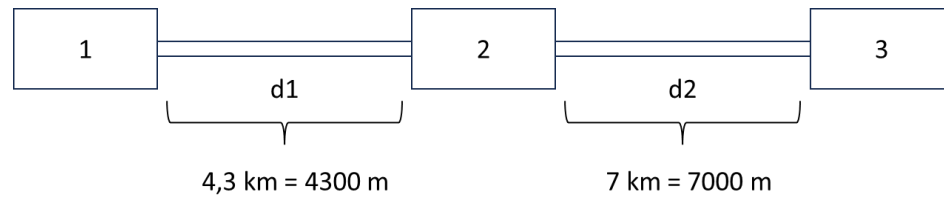
| Outbound |        |           |        |           |        |           |
|----------|--------|-----------|--------|-----------|--------|-----------|
| Kereta/  | A      |           | B      |           | C      |           |
| Stasiun  | Datang | Berangkat | Datang | Berangkat | Datang | Berangkat |
| 1        | 05.25  | 05.30     | 09.20  | 09.26     | 11.42  | 11.45     |
| 2        | 05.36  | 05.37     | 09.31  | 09.32     | 11.50  | 11.52     |
| 3        | 05.42  | 05.46     | 09.37  | 09.40     | 12.00  | 12.01     |

Tabel 3.3 Contoh jadwal kereta api *inbound* yang akan dioptimalkan

| Inbound |        |           |        |           |        |           |
|---------|--------|-----------|--------|-----------|--------|-----------|
| Kereta/ | D      |           | E      |           | F      |           |
| Stasiun | Datang | Berangkat | Datang | Berangkat | Datang | Berangkat |
| 3       | 08.07  | 08.09     | 12.15  | 12.18     | 13.57  | 13.59     |
| 2       | 08.25  | 08.32     | 12.24  | 12.27     | 14.05  | 14.07     |
| 1       | 08.37  | 08.39     | 12.33  | 12.34     | 14.12  | 14.14     |

Data jarak petak blok dapat dilihat pada Gambar 3.2. Misalkan terdapat tiga stasiun yang memiliki dua petak blok yang menghubungkannya dua jalur, dan

kereta api berhenti di setiap stasiun. Berdasarkan data yang diperoleh dari Tabel 3.2, Tabel 3.3, dan Gambar 3.2, didapat data untuk menghitung kendala pada Tabel 3.4.



**Gambar 3.2** Data simulasi panjang petak blok antar stasiun

**Tabel 3.4** Data simulasi dari perjalanan kereta api stasiun 1 - stasiun 3

| Stasiun | Indeks Petak | Jarak antar stasiun | Kecepatan Min | Kecepatan Maks | Waktu tempuh min (menit) | Waktu tempuh maks | Waktu tunggu |
|---------|--------------|---------------------|---------------|----------------|--------------------------|-------------------|--------------|
|         | blok         | (km)                |               |                |                          |                   |              |
| 1       | -            | -                   | -             | -              | -                        | -                 | 4            |
| 2       | d1           | 4,3                 | 43,00         | 51,6           | 5                        | 6                 | 3            |
| 3       | d2           | 7                   | 26,25         | 84             | 5                        | 16                | 2            |

Kecepatan minimum diperoleh berdasarkan jarak yang harus ditempuh setiap petak blok antar stasiun yaitu jarak dibagi waktu tempuh maksimum. Kecepatan maksimum diperoleh dari jarak antar stasiun dibagi waktu tempuh minimum. Waktu tempuh minimum didapat dari minimum waktu yang diperlukan setiap kereta untuk sampai pada stasiun selanjutnya. Sebagai contoh, misalkan pada kereta A dari Stasiun 1 menuju Stasiun 2 memiliki waktu tempuh sebanyak 6 menit lalu kereta B dengan asal dan tujuan yang sama memiliki waktu tempuh sebanyak 5 menit, kemudian hal ini dilakukan pada setiap kereta dengan asal dan tujuan yang sama, sehingga waktu tempuh minimal yang dibutuhkan kereta dari Stasiun 1 ke stasiun 2 adalah 5 menit. Waktu tempuh maksimum didapat dari maksimum waktu yang diperlukan setiap kereta untuk sampai pada stasiun selanjutnya hasilnya diperoleh dengan cara yang sama seperti waktu tempuh minimum tetapi yang diambil nilainya adalah nilai maksimum yang dibutuhkan kereta api dari stasiun asal sampai stasiun tujuan. Berdasarkan data waktu perjalanan pada Tabel 3.2 dan Tabel 3.3 maka diperoleh jadwal kedatangan dan keberangkatan pada Tabel 3.5 dan Tabel 3.6.

Tabel 3.5 Simulasi jadwal kedatangan (*Arrive*) dan keberangkatan (*Departure*) kereta api dari stasiun 1 ke stasiun 3 (menit ke-) untuk kereta *outbound*

| <b>Outbound</b> |             |             |             |             |             |             |
|-----------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <b>Stasiun/</b> | <b>1</b>    |             | <b>2</b>    |             | <b>3</b>    |             |
| <b>Kereta</b>   | <b>1(a)</b> | <b>1(d)</b> | <b>2(a)</b> | <b>2(d)</b> | <b>3(a)</b> | <b>3(d)</b> |
| A               | 0           | 5           | 11          | 12          | 17          | 21          |
| B               | 235         | 241         | 246         | 247         | 252         | 255         |
| C               | 377         | 386         | 396         | 399         | 412         | 416         |

Tabel 3.5 merupakan jadwal kedatangan dan keberangkatan kereta *outbound* dalam menit, sebagai contoh kereta A pada stasiun 1 berangkat pada menit ke 5 menuju stasiun 2 dan menempuh perjalanan sebanyak 6 menit sehingga kereta A pada stasiun 2 datang pada saat menit ke 11 dengan waktu tunggu sebanyak 1 menit di stasiun 2 maka kereta A berangkat pada menit ke 12 lalu menempuh perjalanan dengan waktu sebanyak 5 menit maka kereta datang pada Stasiun 3 pada menit ke 17 kemudian memiliki waktu tunggu sebanyak 4 menit dan berangkat pada menit ke 21 menuju depo.

Tabel 3.6 Simulasi jadwal kedatangan (*Arrive*) dan keberangkatan (*Departure*) kereta api dari stasiun 1 ke stasiun 3 (menit ke-) untuk kereta *inbound*

| <b>Inbound</b>  |             |             |             |             |             |             |
|-----------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <b>Stasiun/</b> | <b>3</b>    |             | <b>2</b>    |             | <b>1</b>    |             |
| <b>Kereta</b>   | <b>3(a)</b> | <b>3(d)</b> | <b>2(a)</b> | <b>2(d)</b> | <b>1(a)</b> | <b>1(d)</b> |
| D               | 7           | 9           | 18          | 25          | 30          | 32          |
| E               | 248         | 251         | 257         | 260         | 266         | 267         |
| F               | 350         | 355         | 367         | 372         | 383         | 386         |

Tabel di atas merupakan jadwal kedatangan dan keberangkatan kereta *inbound* dalam menit, sebagai contoh kereta D pada stasiun 3 berangkat pada menit ke 9 menuju stasiun 2 dan menempuh perjalanan sebanyak 9 menit sehingga kereta D pada stasiun 2 datang pada saat menit ke 18 dengan waktu tunggu sebanyak 7 menit di stasiun 2 maka kereta D berangkat pada menit ke 25 lalu menempuh perjalanan dengan waktu sebanyak 5 menit maka kereta datang pada stasiun ke 1 pada menit ke 30 kemudian memiliki waktu tunggu sebanyak 2 menit dan berangkat pada menit ke 32.

ke 32 menuju depo. Hal ini berlaku pada setiap perjalanan kereta-kereta yang ada pada penelitian ini. Setelah diperoleh data waktu kedatangan dan waktu keberangkatan pada setiap kereta outbound dan inbound dapat diperoleh  $X_{idq} = 416 + 255 + 21 = 692$  ,  $X_{ia1} = 0 + 235 + 377 = 612$  ,  $X_{id1} = 32 + 267 + 386 = 685$  ,  $X_{iaq} = 7 + 248 + 350 = 605$  lalu dihitung solusi awalnya dengan substitusi ke Persamaan (14) sehingga diperoleh  $C_{maks}$  sebesar 155.

### 3.4.3. Pembangkitan Solusi Tetangga

Langkah selanjutnya yang perlu dilakukan adalah membangkitkan solusi tetangga. Solusi tetangga yang dimaksud adalah solusi yang mengalami perubahan keadaan atau posisi atom di dalamnya atau dalam hal ini pengurangan atau penambahan waktu tunggu kereta di stasiun atau pengurangan waktu kedatangan kereta pada stasiun. Dalam penelitian ini pembangkitan solusi tetangga akan menggunakan Algoritma Simple Searching Neighbourhood yang terinspirasi dari Aycan dan Ayav (2009) dalam Wahyuni (2022).

Solusi tetangga diperoleh dengan skema sebagai berikut: Pada setiap iterasi, dipilih secara acak dua kereta pada setiap perjalanan *outbound* dan *inbound*, pilih kereta kedua dan ketiga pada perjalanan *outbound* yaitu kereta B dan C lalu kurangi waktu kedatangan dan keberangkatan kereta B sebesar tiga menit sehingga kereta B memiliki waktu perjalanan lebih cepat tiga menit, pada kereta C kurangi waktu kedatangan dan keberangkatan kereta C sebesar dua menit, lalu untuk perjalanan *inbound* pilih kereta kedua dan ketiga yaitu kereta E dan F lalu kurangi waktu kedatangan dan keberangkatan kereta E sebesar tiga menit, pada kereta F kurangi waktu kedatangan dan keberangkatan kereta F sebesar dua menit.

Tabel 3.7 Contoh Solusi tetangga

| Outbound |      |      |      |      |      |      |
|----------|------|------|------|------|------|------|
| Stasiun/ | 1    |      | 2    |      | 3    |      |
| Kereta   | 1(a) | 1(d) | 2(a) | 2(d) | 3(a) | 3(d) |
| A        | 0    | 5    | 11   | 12   | 17   | 21   |
| B        | 232  | 238  | 243  | 244  | 249  | 252  |



|   |     |     |     |     |     |     |
|---|-----|-----|-----|-----|-----|-----|
| C | 375 | 384 | 394 | 397 | 410 | 414 |
|---|-----|-----|-----|-----|-----|-----|

| Inbound  |      |      |      |      |      |      |
|----------|------|------|------|------|------|------|
| Stasiun/ | 3    |      | 2    |      | 1    |      |
| Kereta   | 3(a) | 3(d) | 2(a) | 2(d) | 1(a) | 1(d) |
| D        | 7    | 9    | 18   | 25   | 30   | 32   |
| E        | 245  | 248  | 254  | 257  | 263  | 264  |
| F        | 348  | 353  | 365  | 370  | 381  | 384  |

Setelah berhasil menemukan solusi tetangga, dilakukan verifikasi terhadap pemenuhan kendala. Jika terdapat pelanggaran kendala, solusi tetangga sebelumnya diabaikan dan pencarian solusi tetangga diulang hingga ditemukan solusi tetangga yang memenuhi syarat. Dalam hal ini, karena solusi tetangga pada Tabel 3.5 tidak melanggar kendala, iterasi pertama tidak memerlukan pengulangan pencarian solusi tetangga. Selanjutnya, solusi tetangga tersebut akan dihitung hasil pemenuhan fungsi tujuannya untuk evaluasi lebih lanjut.

#### 3.4.4. Penghitungan Perubahan Biaya

Dalam kasus optimisasi, biaya merupakan nilai pemenuhan fungsi tujuan dari solusi yang terbentuk pada tahap ini akan dihitung perubahan biaya solusi tetangga dan biaya solusi awal. Berdasarkan solusi awal dan solusi tetangga yang dipaparkan sebelumnya, maka perubahan biaya dapat dihitung:

$$\delta = f(s_0) - f(s') = 155 - 160 = -5 \quad (3.1)$$

Apabila terjadi perubahan biaya ( $\delta$ ) lebih besar dari nol atau sama dengan nol maka solusi tetangga diterima sebagai solusi baru. Namun jika  $\delta$  kurang dari nol, perlu dilakukan perhitungan probabilitasnya ( $P(\delta)$ ). Dalam hal ini karena hasil perhitungan  $\delta$  pada persamaan (3.1) bernilai -5 maka perlu menghitung  $P(\delta)$ . Nilai  $P(\delta)$  sangat bergantung pada suhu saat ini, sehingga pada iterasi pertama diperlukan nilai suhu awal. Misalkan suhu awal diinisiasi bernilai 75. Maka  $P(\delta)$  dapat dihitung sebagai berikut:

$$P(\delta) = \exp\left(-\frac{|\delta|}{T}\right) = \exp\left(-\frac{5}{75}\right) = 0,93 \quad (3.2)$$

Setelah didapatkan nilai  $P(\delta)$  harus dibangkitkan nilai acak real ( $r$ ) dari interval 0 dan 1. Jika  $P(\delta)$  kurang dari  $r$  yang dibangkitkan, maka solusi tetangga diterima sebagai Solusi baru. Sedangkan jika  $P(\delta)$  tidak kurang dari  $r$ , maka solusi awal atau solusi sebelumnya yang dijadikan solusi baru

### 3.4.5. Proses Pendinginan

Jika iterasi sudah mencapai  $N_{rep}$ , selanjutnya dilakukan proses pendinginan yaitu mereduksi nilai suhu saat ini ( $T$ ) dengan menggunakan parameter reduksi ( $\alpha$ ). Pada penelitian ini, akan digunakan rumus penurunan suhu geometric sebagai fungsi pendinginan seperti pada Persamaan 3.2.

$$T = \alpha \cdot T \quad (3.3)$$

Sebagai contoh, misal iterasi sudah mencapai  $N_{rep}$  yang diinisiasi, yaitu iterasi ke-10. Maka selanjutnya dilakukan proses pendinginan seperti berikut.

$$T = \alpha \cdot T = 0,87 \cdot 75 = 65,25 \quad (3.4)$$

Setelah diperoleh nilai suhu terbaru, dilakukan tahapan pencarian Solusi tetangga dimulai dari iterasi kesatu kembali dengan menetapkan suhu terbaru menjadi  $T$ . Proses itu kemudian dilakukan berulang sampai suhu saat ini bernilai kurang dari suhu akhir ( $T_f$ ). Tabel 3.8 merupakan solusi baru hasil perulangan pertama.

Tabel 3.8 Contoh Solusi Baru pada Akhir Loop Pertama

| Outbound |      |        |        |      |      |          |          |      |      |          |          |
|----------|------|--------|--------|------|------|----------|----------|------|------|----------|----------|
| 1(a)     | 1(d) | Tempuh | tunggu | 2(a) | 2(d) | Tempuh.1 | tunggu.1 | 3(a) | 3(d) | Tempuh.2 | tunggu.2 |
| 0        | 7    | 0      | 7      | 11   | 12   | 0        | 6        | 17   | 21   | 0        | 3        |
| 234      | 238  | 6      | 4      | 244  | 250  | 5        | 6        | 255  | 256  | 5        | 1        |
| 379      | 386  | 6      | 7      | 392  | 395  | 5        | 3        | 400  | 407  | 5        | 7        |
| Inbound  |      |        |        |      |      |          |          |      |      |          |          |
| 3(a)     | 3(d) | Tempuh | tunggu | 2(a) | 2(d) | Tempuh.1 | tunggu.1 | 1(a) | 1(d) | Tempuh.2 | tunggu.2 |
| 1        | 2    | 0      | 1      | 2    | 4    | 0        | 2        | 4    | 5    | 0        | 1        |
| 245      | 248  | 16     | 3      | 264  | 271  | 6        | 7        | 277  | 279  | 6        | 2        |
| 374      | 377  | 5      | 3      | 382  | 389  | 6        | 7        | 395  | 398  | 5        | 3        |

### 3.4.6. Penghentian Algoritma

Penelitian ini menggunakan kriteria penghentian algoritma ketika suhu telah mencapai suhu akhir. Sebagai contoh dalam kasus ini nilai minimum dari fungsi

tujuan adalah 135, maka disaat nilai dari solusi baru mencapai suhu akhir yang dipilih pada kasus ini yaitu 135 saat itu pula algoritma dihentikan. Nilai minimum fungsi tujuan dapat dihitung dari jumlah menit waktu keberangkatan kereta saat mencapai stasiun akhir ketika kereta akan menuju depo dikurangi jumlah menit waktu kedatangan kereta pada stasiun pertama. Perhitungan nilai ini berlaku untuk perjalanan kereta *outbound* maupun *inbound*.