

BAB III

METODOLOGI PENELITIAN

Bab ini membahas representasi jaringan dari sebuah gedung, model optimisasi penentuan jalur evakuasi, dan metode penyelesaiannya menggunakan metode NSGA-II.

3.1 Deskripsi Masalah

Penelitian ini membahas masalah penentuan jalur evakuasi dari sebuah gedung. Posisi awal pengungsi adalah di dalam ruangan atau koridor. Jalur evakuasi adalah rute yang harus dilalui pengungsi dari posisi awal, yang disebut titik evakuasi, ke titik kumpul yang berada di luar gedung. Lebih jauh, penelitian ini akan meneliti masalah penentuan jalur evakuasi di Gedung Sekolah Pascasarjana UPI. Penelitian jalur evakuasi sangat penting untuk memastikan efektivitas jalur evakuasi yang diterapkan di Gedung Sekolah Pascasarjana UPI. Selain itu, penelitian ini menjadi lebih penting karena hingga saat ini belum ada penelitian serupa yang dilakukan. Dikarenakan perencanaan jalur evakuasi melibatkan banyak faktor, maka akan digunakan model multiobjektif. Penelitian ini akan menggunakan metode NSGA-II untuk mencari jalur evakuasi yang optimal untuk menyelesaikan masalah multiobjektif.

3.2 Data Penelitian

Data yang digunakan dalam penelitian ini adalah denah Gedung Sekolah Pascasarjana UPI yang terdiri dari enam lantai, yang masing-masing lantai dihubungkan oleh empat tangga yang berbeda. Data tersebut terdiri dari titik evakuasi dan titik kumpul. Titik evakuasi merujuk kepada ruang kelas, ruang dosen, mushola, dan *restroom*. Sedangkan, titik kumpul pada Gedung Sekolah Pascasarjana UPI berada di luar gedung.

Dalam penelitian ini, dapat dihitung jarak yang dapat ditempuh oleh pengungsi dari tiap-tiap simpul sumber menuju simpul tujuan, luas area untuk mengetahui kapasitas jalan, serta mengestimasi jumlah pengungsi yang dapat melewati jalur evakuasi dalam satu waktu tertentu untuk mengetahui kecepatan pengungsi pada setiap jalur. Dari data yang diperoleh, akan dihitung waktu total minimum yang dibutuhkan pengungsi untuk melakukan evakuasi serta tingkat

kemacetan selama proses evakuasi dengan menentukan jalur evakuasi yang optimal.

Masalah penentuan jalur evakuasi yang optimal akan dilakukan dengan pendekatan multiobjektif yang diselesaikan dengan metode *Non-dominated Sorting Genetic Algorithm II*. Metode penyelesaian akan dibahas secara lengkap pada sub bab berikutnya.

3.3 Representasi Jaringan

Langkah pertama dalam menyelesaikan masalah perencanaan jalur evakuasi adalah dengan merepresentasikan jalur evakuasi ke dalam jaringan yang terdiri dari simpul sumber, simpul perantara, simpul tujuan, dan busur. Simpul digunakan untuk menggambarkan lokasi dari area seperti ruangan dan koridor. Sedangkan, busur digambarkan sebagai penghubung antara dua simpul jalur evakuasi seperti tangga atau lorong yang dapat dilalui oleh pengungsi.

Simpul sumber mewakili titik di dalam gedung tempat para pengungsi terkonsentrasi, seperti ruang kelas, ruang dosen, mushola dan *restroom*. Simpul tujuan mewakili titik kumpul di luar bangunan. Sedangkan, simpul perantara merupakan simpul-simpul yang menghubungkan simpul sumber dengan simpul tujuan.

Setiap busur memiliki bobot berupa panjang antar simpul dan kapasitas busur. Dalam penelitian ini, panjang antar simpul dapat diperoleh dengan menghitung jarak antara dua simpul. Kapasitas busur merupakan kapasitas maksimum pengungsi yang dapat melewati busur (jalan). Dalam penelitian ini, kapasitas busur dapat diperoleh dengan menghitung luas area seperti lorong atau koridor.

Dalam penelitian ini, volume aliran merepresentasikan jumlah pengungsi yang melewati jalan (busur) dalam satu waktu tertentu. Dalam hal ini, volume aliran memberikan informasi tentang seberapa cepat pengungsi melakukan evakuasi dalam kaitannya dengan kapasitas busur. Rasio volume aliran terhadap kapasitas busur menunjukkan seberapa dekat kapasitas penggunaan jalur evakuasi tersebut dengan kapasitas maksimum yang dapat ditampung oleh luas area atau lebar jalan. Semakin besar rasio ini, maka semakin dekat kapasitas busur dengan batas maksimumnya, yang mengindikasikan kepadatan atau kemacetan pada jalur evakuasi tersebut.

3.4 Menyusun Asumsi

Tujuan dari penelitian ini adalah untuk menentukan jalur evakuasi yang optimal yang dapat meminimalkan total waktu evakuasi dan meminimalkan kemacetan selama proses evakuasi. Adapun asumsi-asumsi yang digunakan adalah sebagai berikut.

1. Tujuan model adalah meminimalkan waktu evakuasi dan meminimalkan kemacetan selama proses evakuasi.
2. Arah busur diasumsikan selalu menuju simpul tujuan yang terdekat.
3. Kecepatan maksimum setiap pengungsi diasumsikan sama, dengan kata lain atribut pengungsi seperti usia dan jenis kelamin diabaikan.
4. Setiap pengungsi mengetahui dengan baik setiap komponen dan tata letak gedung.

3.5 Pembangunan Model Optimasi

Masalah yang dibahas dalam penelitian ini adalah penentuan jalur evakuasi yang optimal yang dapat meminimalkan total waktu dan meminimalkan kemacetan selama proses evakuasi. Masalah ini dikategorikan sebagai masalah multiobjektif. Selanjutnya, akan dibentuk model optimisasi penentuan jalur evakuasi berbasis NSGA-II yang merujuk pada Li, Fang, dkk. (2010).

Terlebih dahulu didefinisikan himpunan, parameter, dan variabel keputusan yang digunakan untuk menurunkan model optimisasi penentuan jalur evakuasi sebagai berikut:

1. Himpunan
 - N : Himpunan simpul pada jaringan evakuasi
 - S : Himpunan simpul sumber (titik evakuasi)
 - D : Himpunan simpul tujuan akhir (lokasi titik berkumpul)
2. Parameter
 - C_{ij} : Kapasitas busur untuk jalur evakuasi antara simpul i dan j
 - l_{ij} : Jarak antara simpul i dan j
 - V_{ij} : Volume aliran pengungsi pada jalur evakuasi antara simpul i dan j
 - v_{max} : Kecepatan maksimum pengungsi ketika tidak ada hambatan

3. Variabel Keputusan

- x_{ij} : Variabel keputusan biner yang menentukan apakah jalur evakuasi dari simpul i ke j dipilih ($x_{ij} = 1$) atau tidak ($x_{ij} = 0$)

Pada penelitian ini akan diberikan bobot untuk masing-masing busur yaitu C_{ij} dan l_{ij} . Misalkan v_{max} adalah kecepatan pengungsi ketika tidak ada hambatan dari simpul i menuju simpul j . Kecepatan pengungsi disetiap busur (i, j) nilainya tidak selalu konstan tetapi bergantung pada volume aliran pengungsi yang melewati busur (i, j) dan kapasitas dari busur (i, j) , sehingga kecepatan pengungsi di setiap busur (i, j) akan mengalami penurunan kecepatan secara eksponensial apabila terindikasi kepadatan atau kemacetan pada jalur evakuasi tersebut. Merujuk pada Li, Fang, dkk. (2010), kecepatan pengungsi di setiap busur $v(i, j)$ dapat dihitung dengan :

$$v(i, j) = \begin{cases} v_{max}, & \text{jika } V_{ij}/C_{ij} < 0,5 \\ v_{max} * e^{-0,5*(V_{ij}/C_{ij})}, & \text{lainnya} \end{cases} \quad (3.1)$$

Karena kecepatan pengungsi tidak konstan tetapi berubah pada selang waktu Δt , maka waktu yang diperlukan pengungsi di setiap busur (i, j) dapat dihitung dengan:

$$t(i, j) = \frac{l_{ij}}{v(i, j)} \quad (3.2)$$

Misalkan $f(i, j)$ merepresentasikan tingkat kemacetan antara simpul i dan j pada selang waktu Δt . Merujuk pada Li, Fang, dkk. (2010), fungsi kemacetan dapat dinyatakan sebagai

$$f(i, j) = \begin{cases} 0, & \text{jika } (V_{ij}/C_{ij}) < 0,5 \\ e^{0,5*(V_{ij}/C_{ij}-0,5)} - 1, & \text{lainnya} \end{cases} \quad (3.3)$$

Adapun kendala-kendala dari model optimisasi penentuan jalur evakuasi adalah sebagai berikut:

1. Setiap simpul pada jaringan evakuasi setidaknya terhubung dengan minimal satu jalur evakuasi sehingga tidak ada simpul yang terisolasi tanpa jalur evakuasi yang dapat digunakan. Kendala ini direpresentasikan oleh persamaan:

$$\sum_j x_{ij} \geq 1, \quad i \neq j, \quad j \in N$$

2. Jumlah volume aliran pengungsi yang memasuki simpul i sama dengan volume aliran pengungsi yang keluar dari simpul j . Kendala ini direpresentasikan oleh persamaan:

$$\sum_{i \in N} V_{ij} - \sum_{j \in N} V_{ij} = 0, \quad \forall i, j \in (N - S \cup D)$$

3. Volume aliran pengungsi pada setiap busur tidak boleh melebihi kapasitas maksimum busur. Kendala ini direpresentasikan oleh persamaan:

$$V_{ij} \leq C_{ij}, \quad i, j \in N$$

4. Setiap jalur evakuasi berawal dari simpul sumber. Kendala ini direpresentasikan oleh persamaan:

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in S$$

5. Setiap jalur evakuasi berakhir pada simpul tujuan. Kendala ini direpresentasikan oleh persamaan:

$$\sum_{i \in N} x_{ij} = 1, \quad \forall j \in D$$

6. Batasan variabel yang bersifat biner (0 atau 1) untuk memastikan bahwa setiap jalur evakuasi dipilih atau tidak dipilih untuk digunakan selama evakuasi. Batasan ini ditulis sebagai berikut:

$$x_{ij} = \{0,1\}; \quad i, j \in N$$

Dengan demikian, model optimisasi penentuan jalur evakuasi multiobjektif berbasis NSGA-II diformulasikan sebagai berikut:

Meminimumkan

$$T = \sum_i \sum_j t(i,j)x_{ij} \quad (3.4)$$

Meminimumkan

$$F = \sum_i \sum_j f(i,j)x_{ij} \quad (3.5)$$

Terhadap:

$$\sum_j x_{ij} \geq 1, \quad i \neq j, \quad j \in N \quad (3.6)$$

$$\sum_{i \in N} V_{ij} - \sum_{j \in N} V_{ij} = 0, \quad \forall i, j \in (N - S \cup D) \quad (3.7)$$

$$V_{ij} \leq C_{ij}, \quad i, j \in N \quad (3.8)$$

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in S \quad (3.9)$$

$$\sum_{i \in N} x_{ij} = 1, \quad \forall j \in D \quad (3.10)$$

$$x_{ij} = \{0,1\}; \quad i, j \in N \quad (3.11)$$

3.6 Penyelesaian Model Optimasi

Dalam penelitian ini, penyelesaian dari model multiobjektif dilakukan dengan menggunakan teknik yang dikembangkan oleh Kalyanmoy Deb (2001). Tujuan penyelesaian model adalah menentukan jalur evakuasi yang efektif dan volume aliran pengungsi pada setiap jalur yang dilewati sedemikian sehingga dapat meminimalkan waktu evakuasi dan meminimalkan kemacetan selama proses evakuasi berlangsung. Terdapat beberapa metode yang dapat digunakan untuk menyelesaikan masalah multiobjektif, salah satunya adalah dengan menggunakan algoritma *Non-dominated Sorting Genetic Algorithm* (NSGA-II) yang merupakan pengembangan dari algoritma genetika berdasarkan penelitian Li, Fang, dkk. (2010). Adapun langkah-langkah penyelesaian perencanaan jalur evakuasi menggunakan NSGA-II adalah sebagai berikut.

1. Pengungsi diatur dan ditempatkan ke masing-masing ruangan pada setiap lantai gedung.
2. Inisialisasi populasi awal dilakukan secara acak, di mana *encoding chromosome* digunakan untuk merepresentasikan solusi, dalam hal ini digunakan untuk mengkodekan jalur evakuasi dari simpul awal menuju simpul tujuan. Dalam penelitian ini kromosom dibentuk dari bilangan biner yang merepresentasikan jalur evakuasi dipilih atau tidak dipilih untuk digunakan selama evakuasi. Panjang kromosom ditentukan oleh banyaknya jalur dalam jaringan evakuasi.
3. Setelah populasi awal dihasilkan, GA akan menerjemahkan setiap gen ke dalam variabel yang sesuai, yaitu jalur evakuasi dan menghitung fungsi tujuan yaitu total waktu evakuasi dari seluruh jaringan dan total kemacetan selama proses

evakuasi berlangsung. Selanjutnya, tentukan *parent* dan *offspring* awal. Gabungkan kedua solusi *parent* dan *offspring* $R_t = P_t \cup Q_t$ dan lakukan *Non-dominated Sort* sehingga diperoleh *Pareto-optimal front*. Langkah-langkah *non-dominated sort* adalah sebagai berikut:

- a. Tetapkan $i = 1$ dan buat himpunan kosong $P' = \emptyset$.
 - b. Untuk sebuah solusi $j \in P$ dengan $j \neq i$, periksa apakah solusi j mendominasi solusi i pada setiap fungsi tujuan (f_1 dan f_2). Jika ya, lanjut ke langkah d.
 - c. Jika masih ada solusi yang tersisa di P , tambahkan $j + 1$ dan lakukan langkah b. Jika tidak, atur $P' = P' \cup \{i\}$.
 - d. Tambahkan $i + 1$. Jika $i \leq N$, kembali ke langkah b. Jika tidak, berhenti dan nyatakan P' sebagai himpunan *non-dominated*.
4. Bentuk *parent* baru P_{t+1} dari hasil *Pareto-optimal front*. Langkah-langkah untuk membentuk *parent* baru adalah sebagai berikut:
- a. Atur $P_{t+1} = \emptyset$ dan $i = 1$.
 - b. Lakukan $P_{t+1} = P_{t+1} \cup F_i$ dan $i = i + 1$, hingga $|P_{t+1}| + |F_i| < N$. Lakukan *crowding sort* pada hasil *Pareto-optimal front* dengan menggunakan perhitung *crowding distance* untuk setiap individu di setiap *front* untuk menentukan solusi mana yang akan dipilih untuk menjadi bagian dari P_{t+1} .

Crowding distance adalah metode yang digunakan dalam algoritma NSGA-II untuk mengukur kerapatan solusi di dalam *front*. Hal ini membantu dalam mempertahankan keanekaragaman solusi dalam populasi. *Crowding distance* dapat diperoleh dengan menghitung jarak antara dua buah solusi. Dalam permasalahan multiobjektif dengan menggunakan NSGA-II, solusi yang memiliki *crowding distance* lebih tinggi akan lebih diprioritaskan untuk dipertahankan dalam populasi.

5. Bentuk Q_{t+1} dari P_{t+1} dengan menggunakan *tournament selection*. Adapun langkah-langkah *tournament selection* adalah sebagai berikut:
 - a. Tentukan ukuran *tournament*, yaitu jumlah individu yang akan dipertandingkan dalam setiap *tournament*. Biasanya, ukuran *tournament* ini merupakan parameter yang ditentukan sebelumnya oleh pengguna.

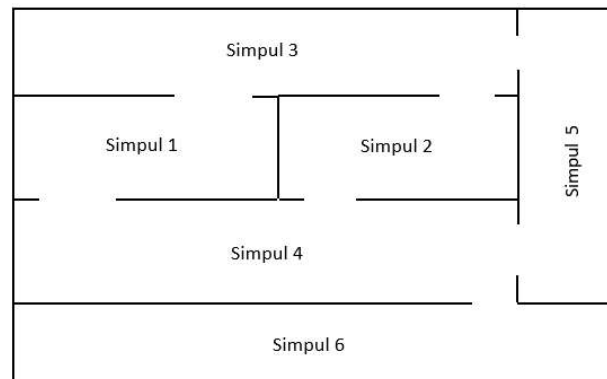
- b. Lakukan *tournament* sebanyak yang diperlukan untuk memilih semua individu yang diperlukan untuk membentuk populasi berikutnya. Dalam setiap *tournament*, pilih secara acak sejumlah individu sesuai dengan ukuran *tournament*.
 - c. Evaluasi semua individu yang berpartisipasi dalam *tournament* dilakukan berdasarkan *rank* atau *crowding distance*.
 - d. Pilih individu dengan *rank* atau *crowding distance* tertinggi sebagai pemenang *tournament*. Individu ini akan menjadi salah satu *parent* untuk menghasilkan generasi berikutnya.
 - e. Ulangi langkah a hingga d sebanyak yang diperlukan. Jumlah *tournament* yang dilakukan biasanya sama dengan jumlah individu dalam populasi yang ingin dihasilkan untuk generasi berikutnya.
6. Lakukan operator algoritma genetika yaitu *crossover* dan mutasi pada Q_{t+1} . Proses *crossover* dilakukan pada kromosom yang telah lolos seleksi *tournament*. Pada penelitian ini, akan digunakan *multi-point crossover*. Pada *crossover* terdapat parameter probabilitas *crossover* (P_c), di mana nilai P_c antara 0 sampai 1. Kemudian pilih bilangan acak antara 0 dan 1. Jika bilangan acak yang dipilih kurang dari P_c maka dilakukan *crossover*. Jika bilangan acak yang dipilih lebih dari P_c maka tidak terjadi proses *crossover*. Langkah-langkah *multi-point crossover* adalah sebagai berikut:
- a. Pilih dua kromosom dari populasi secara acak.
 - b. Tentukan titik pemotongan secara acak dengan lebih dari satu titik pemotongan di antara gen-gen kromosom yang akan dilakukan *crossover*.
 - c. Buat *offspring* baru dengan mengambil bagian pertama dari salah satu *parent* hingga titik pemotongan pertama, lalu beralih ke *parent* lainnya, dan seterusnya sesuai dengan titik pemotongan yang telah ditentukan.
 - d. Setelah proses *crossover*, hasilnya terdapat dua *offspring* baru yang kemudian akan menjadi bagian dari populasi berikutnya.

Setelah proses *crossover*, kromosom-kromosom tersebut akan dilakukan proses mutasi. Dalam algoritma genetika, mutasi adalah proses di mana gen dalam kromosom diubah secara acak untuk menciptakan variasi baru. Mutasi penting dalam menjaga keberagaman genetik dalam populasi dan mencegah solusi

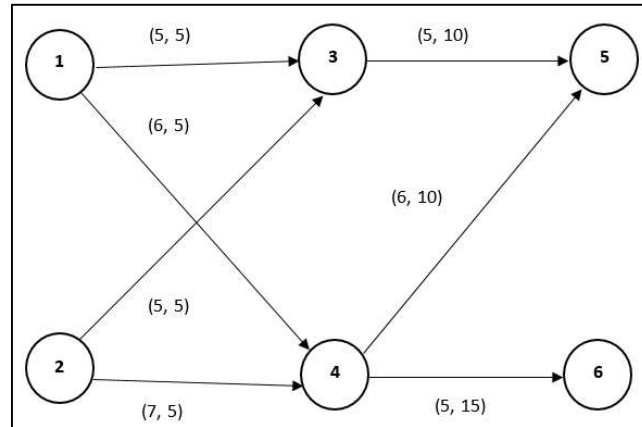
terjebak dalam solusi optimum lokal. Pada proses mutasi terdapat probabilitas mutasi (P_m). Probabilitas ini akan menentukan seberapa sering mutasi terjadi pada kromosom. Langkah-langkah mutasi adalah sebagai berikut:

- a. Pilih satu kromosom yang akan dilakukan mutasi.
 - b. Untuk setiap bit dalam kromosom akan dilakukan pemilihan bilangan secara acak antara 0 dan 1. Jika bilangan acak yang dihasilkan kurang dari nilai P_m maka akan terjadi mutasi. Jika bilangan acak yang dihasilkan lebih dari nilai P_m maka tidak akan terjadi mutasi.
 - c. Jika gen mengalami mutasi, nilai bit tersebut akan diubah, misalnya dari 0 menjadi 1 atau dari 1 menjadi 0.
 - d. Kromosom yang telah dimutasi akan dimasukkan kembali ke dalam populasi.
 - e. Ulangi langkah a hingga d untuk kromosom lain dalam populasi.
7. Ulangi langkah 3, 4, dan 5 hingga mencapai nilai objektif yang paling minimum atau hingga kriteria pemberhentian terpenuhi.

Untuk memperjelas langkah penyelesaian tersebut, diberikan contoh kasus sebagai berikut.



Gambar 3.1 Contoh denah bangunan



Gambar 3.2 Representasi jaringan dari denah gedung pada Gambar 3.1

Gambar 3.2 adalah salah satu contoh bagaimana merepresentasikan gedung ke dalam bentuk jaringan dengan bobot pada busur masing-masing yang merepresentasikan kapasitas jalur evakuasi (C_{ij}) dan jarak antara dua simpul (l_{ij}). Simpul 1 dan 2 merupakan simpul sumber atau titik evakuasi. Simpul 3 dan 4 adalah simpul perantara. Sedangkan, simpul 5 dan 6 adalah simpul tujuan atau titik kumpul. Pada contoh kasus ini, bobot untuk setiap busur (v_i, v_j) terdapat pada Tabel 3.1.

Tabel 3.1 Kapasitas jalur evakuasi, volume aliran pengungsi pada jalur evakuasi, dan jarak antara dua simpul

| (v_i, v_j) | (C_{ij}) | (l_{ij}) |
|--------------|------------|------------|
| (1,3) | 5 | 5 |
| (1,4) | 6 | 5 |
| (2,3) | 5 | 5 |
| (2,4) | 7 | 5 |
| (3,5) | 5 | 10 |
| (4,5) | 6 | 10 |
| (4,6) | 5 | 15 |

Langkah pertama adalah menempatkan pengungsi di masing-masing simpul sumber secara acak. Misalkan terdapat 5 orang pengungsi, di mana 3 orang ditempatkan di simpul 1 dan 2 orang ditempatkan di simpul 2. Pada contoh kasus ini, misalkan pengungsi pada simpul 1 dan 2 selalu berkelompok dan akan bergerak menuju jalur evakuasi yang sama.

Langkah kedua adalah inisialisasi populasi di mana *encoding chromosome* merepresentasikan solusi yang dalam hal ini adalah jalur evakuasi dari simpul awal menuju simpul tujuan. Setiap bit dalam *chromosome* merepresentasikan busur dalam urutan $\{(v_1, v_3), (v_1, v_4), (v_3, v_5), (v_4, v_5), (v_4, v_6), (v_2, v_3), (v_2, v_4)\}$. Nilai 0 digunakan untuk merepresentasikan ketidakberadaan busur (atau dengan kata lain tidak dilalui sebagai jalur evakuasi). Sedangkan, nilai 1 digunakan untuk merepresentasikan keberadaan busur (atau dengan kata lain dilalui sebagai jalur evakuasi). Misalkan dibangkitkan 6 individu sebagai populasi awal yang dapat dilihat pada Tabel 3.2.

Tabel 3.2 Contoh kasus populasi awal

| | | | | | | | |
|------------|---|---|---|---|---|---|---|
| Kromosom 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| Kromosom 2 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| Kromosom 3 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| Kromosom 4 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Kromosom 5 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| Kromosom 6 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |

Sebagai contoh, pada kromosom 1, bit pertama (1) menunjukkan bahwa busur (v_1, v_3) ada dalam solusi. Sedangkan, bit kedua (0) menunjukkan bahwa busur (v_1, v_4) tidak ada dalam solusi. Begitupun selanjutnya. Sehingga jalur evakuasi yang terbentuk untuk kromosom 1 adalah $\{(v_1, v_3), (v_2, v_3), (v_3, v_5)\}$. Selanjutnya, kromosom 1 sampai kromosom 6 akan membentuk suatu populasi.

Langkah ketiga adalah mengevaluasi setiap kromosom dalam populasi terhadap fungsi tujuan, yaitu meminimumkan total waktu evakuasi dan meminimumkan kemacetan, serta menentukan *parent* dan *offspring* awal. Misalkan kromosom 1, 2, dan 3 menjadi *parent* awal, sedangkan kromosom 4, 5, dan 6 menjadi *offspring* awal. Substitusi kapasitas dan volume aliran pengungsi pada jalur evakuasi ke Persamaan (3.1). Pada penelitian ini, diasumsikan kecepatan pengungsi ketika tidak ada hambatan v_{max} adalah $1,5 \text{ m/s}$. Sebagai contoh, pada kromosom 1 dapat diperoleh kecepatan pengungsi di setiap busur yang dilewati adalah sebagai berikut. Jumlah pengungsi yang melewati jalur (1, 3) adalah 3 orang pengungsi

sehingga rasio dari volume dan kapasitas adalah $\frac{V_{(1,3)}}{C_{(1,3)}} = \frac{3}{5} > 0,5$, maka kecepatan pengungsi yang melewati jalur (1, 3) adalah

$$v(1,3) = 1,5 \times e^{-0,5(3/5)} = 1,11 \text{ m/s}$$

Jumlah pengungsi yang melewati jalur (2, 3) adalah 2 orang pengungsi sehingga rasio dari volume dan kapasitas adalah $\frac{V_{(2,3)}}{C_{(2,3)}} = \frac{2}{5} < 0,5$, maka kecepatan pengungsi yang melewati jalur (2, 3) adalah kecepatan pengungsi ketika tidak ada hambatan v_{max} , yaitu

$$v(2,3) = 1,5 \text{ m/s}$$

Jumlah pengungsi yang melewati jalur (3, 5) adalah 5 orang pengungsi sehingga rasio dari volume dan kapasitas adalah $\frac{V_{(3,5)}}{C_{(3,5)}} = \frac{5}{5} > 0,5$, maka kecepatan pengungsi yang melewati jalur (3, 5) adalah

$$v(3,5) = 1,5 \times e^{-0,5(5/5)} = 0,907 \text{ m/s}$$

Selanjutnya, substitusikan nilai kecepatan tersebut untuk menghitung waktu dan tingkat kemacetan pada masing-masing jalur menggunakan Persamaan (3.2) dan Persamaan (3.3). Sebagai contoh, nilai waktu dan kemacetan pada kromosom 1 adalah sebagai berikut.

Untuk jalur (1, 3) diperoleh hasil

$$t(1,3) = \frac{l_{13}}{v(1,3)} = \frac{5}{1,11} = 4,5$$

$$f(1,3) = e^{0,5 * (V_{13}/C_{13} - 0,5)} - 1 = e^{0,5 * (3/5 - 0,5)} = 0,0513$$

Untuk jalur (2, 3) diperoleh hasil

$$t(2,3) = \frac{l_{23}}{v(2,3)} = \frac{5}{1,5} = 3,33$$

karena rasio dari volume dan kapasitas untuk jalur (2, 3) adalah $\frac{V_{(2,3)}}{C_{(2,3)}} = \frac{2}{5} < 0,5$ maka tingkat kemacetan pada jalur (2, 3) adalah $f(2,3) = 0$. Untuk jalur (3, 5) diperoleh hasil

$$t(3,5) = \frac{l_{35}}{v(3,5)} = \frac{10}{0,907} = 11,12$$

$$f(3,5) = e^{0,5*(V^{35}/c_{35}^{-0,5})} - 1 = e^{0,5*(5/5^{-0,5})} = 0,2840$$

Selanjutnya, substitusikan nilai waktu dan nilai kemacetan tersebut ke Persamaan (3.4) dan Persamaan (3.5) sehingga diperoleh total waktu evakuasi pengungsi dan total kemacetan pada masing-masing kromosom. Sebagai contoh, total waktu evakuasi pengungsi dan total kemacetan pada kromosom 1 adalah sebagai berikut.

$$t_1 = 4,5 + 3,33 + 11,12 = 18,83$$

$$f_1 = 0,0513 + 0 + 0,2840 = 0,33$$

Selanjutnya, lakukan hal tersebut untuk kromosom lainnya. Hasil perhitungan total waktu evakuasi dan tingkat kemacetan untuk kromosom yang lain disajikan dalam Tabel 3.3.

Tabel 3.3 Evaluasi kromosom terhadap fungsi tujuan

| Parent P_t | | |
|--------------|-------|-------|
| Kromosom | T | C |
| 1 | 18,83 | 0,33 |
| 2 | 23,5 | 0,102 |
| 3 | 26,26 | 0 |

| Offspring Q_t | | |
|-----------------|--------|-------|
| Kromosom | T | C |
| 4 | 26,839 | 0,102 |
| 5 | 22,93 | 0 |
| 6 | 27,78 | 0,051 |

Selanjutnya, gabungkan kedua solusi *parent* dan *offspring* $R_t = P_t \cup Q_t$ untuk dilakukan *non-dominated sort*. Pada contoh kasus ini, misalkan $i = 1$ adalah kromosom 1 dan $F_1 = \emptyset$. Untuk $j = 2$ dan $j \neq i$, kromosom 2 memiliki solusi nilai yang tidak mendominasi kromosom 1, baik pada fungsi tujuan pertama T maupun fungsi tujuan kedua C . Karena masih ada solusi lain yang tersisa, maka tambahkan $j + 1 = 3$. Kromosom 3 memiliki solusi nilai yang tidak mendominasi kromosom 1. Ulangi langkah tersebut hingga tidak ada solusi lain yang tersisa. Selanjutnya, $F_1 = \emptyset \cup \{1\} = \{1\}$ dan tambahkan $i + 1 = 2$. Lakukan langkah-langkah *non-*

dominated sort pada solusi lain yang tersisa sehingga diperoleh hasil *non-dominated sort* pada masing-masing *front* adalah sebagai berikut:

$$\mathbf{Front. } F_1 = \{1, 5\}$$

$$F_2 = \{2, 3\}$$

$$F_3 = \{4, 6\}$$

Langkah keempat adalah membentuk *parent* baru P_{t+1} dari hasil *Pareto-optimal front*. Pada contoh kasus ini, misalkan $P_{t+1} = \emptyset$ dan $i = 1$. $|P_{t+1}| + |F_i| < N$ adalah $0 + 2 \leq 3$, maka P_{t+1} akan diperbaharui menjadi $P_{t+1} = \{1, 5\}$. Untuk $i = 2$, $|P_{t+1}| + |F_i| < N$ adalah $2 + 2 \not\leq 3$, maka akan dilakukan *crowding sort* pada *front* F_2 dengan menggunakan perhitungan *crowding distance* untuk menentukan solusi mana yang akan dipilih untuk menjadi bagian dari P_{t+1} .

Pada contoh kasus ini, *front* F_2 hanya memiliki dua solusi yaitu kromosom 2 dan kromosom 3 sehingga *crowding distance* antara dua solusi ini adalah sama. Sehingga dapat digunakan aturan seleksi tambahan, seperti memilih solusi yang memiliki nilai *fitness* lebih baik untuk salah satu fungsi tujuan. Dalam hal ini, kromosom 3 memiliki nilai pada fungsi tujuan kedua C yang lebih baik dibandingkan kromosom 2. Oleh karena itu, akan dipilih kromosom 3 untuk menjadi bagian dari $P_{t+1} = \{1, 5, 3\}$. Karena $|P_{t+1}| = 3 \leq 3 = N$ maka $|P_{t+1}|$ sudah terpenuhi (maksimal).

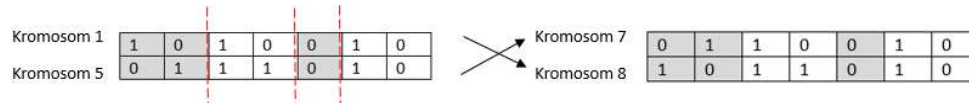
Langkah kelima adalah membentuk *offspring* Q_{t+1} dari P_{t+1} dengan menggunakan *tournament selection*. Misalkan akan dilakukan *tournament selection* dari solusi berikut:

$$(1,5), (1,3), (5,3)$$

sehingga masing-masing solusi mengikuti *tournament* sebanyak dua kali. Seleksi *tournament* dilakukan berdasarkan *rank* atau *crowding distance*. *Front* F_1 memiliki *rank* 1, *front* F_2 memiliki *rank* 2, dan seterusnya. Jika dua solusi berada pada *front* yang berbeda maka seleksi *tournament* dilakukan dengan menggunakan *rank*. Jika dua solusi berada pada *front* yang sama maka seleksi *tournament* dilakukan dengan menggunakan *crowding distance*. Pada contoh kasus ini, hasil dari seleksi *tournament* adalah $\{1,1,3\}$.

Langkah keenam adalah melakukan operator pada algoritma genetika yaitu *crossover* dan mutasi. Proses *crossover* dilakukan pada individu yang telah lolos

seleksi *tournament*. Pada penelitian ini, akan digunakan *multi-point crossover*. Pada contoh kasus ini, misalkan nilai P_c yang digunakan adalah 0,7 dan misalkan bilangan acak yang dipilih untuk kromosom 1 dan kromosom 3 adalah 0,4. Akibatnya kromosom 1 dan kromosom 3 akan dilakukan proses *crossover*. Misalkan dipilih titik pemotongan pada posisi kedua, keempat, dan kelima.



Setelah proses *crossover*, kromosom-kromosom tersebut akan dilakukan proses mutasi. Pada contoh kasus ini, mutasi akan dilakukan dengan probabilitas mutasi sebesar 0,6. Misalkan kromosom 7 terpilih untuk dimutasi. Misalkan bit pertama pada kromosom 7 menghasilkan bilangan acak 0,7 maka bit pertama tidak akan mengalami mutasi. Selanjutnya, pada bit kedua pada kromosom 7 menghasilkan bilangan acak 0,8 maka bit kedua tidak akan mengalami mutasi. Selanjutnya, pada bit ketiga pada kromosom 7 menghasilkan bilangan acak 0,3 maka bit ketiga akan mengalami mutasi, dari 1 menjadi 0. Begitupun seterusnya hingga bit terakhir pada kromosom 7. Hasil mutasi pada contoh kasus ini adalah sebagai berikut.

Sebelum mutasi: kromosom 7

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|

Setelah mutasi: kromosom 9

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|

Dari proses seleksi *tournament*, *crossover*, dan mutasi, diperoleh hasil $Q_{t+1} = \{1, 8, 9\}$. Nilai objektif untuk masing-masing solusi tersebut adalah sebagai berikut.

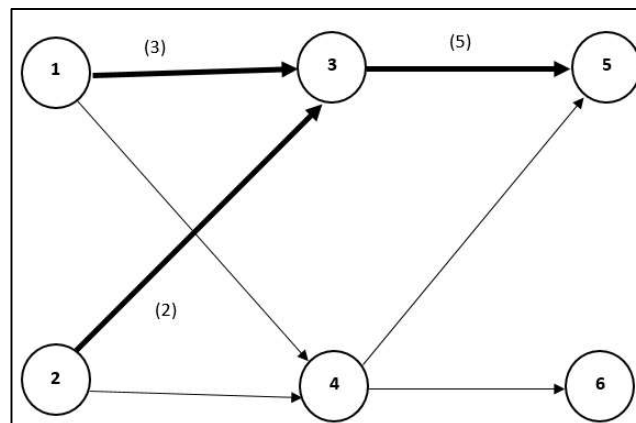
| Kromosom | T | C |
|----------|-------|------|
| 1 | 18,83 | 0,33 |
| 8 | 18,83 | 0,33 |
| 9 | 24,1 | 0,28 |

Untuk contoh kasus yang diberikan, diperoleh nilai objektif untuk P_{t+1} dan Q_{t+1} adalah sebagai berikut.

| Parent P_{t+1} | | |
|------------------|-------|------|
| Kromosom | T | C |
| 1 | 18,83 | 0,33 |
| 5 | 22,93 | 0 |
| 3 | 26,26 | 0 |

| Offspring Q_{t+1} | | |
|---------------------|-------|------|
| Kromosom | T | C |
| 1 | 18,83 | 0,33 |
| 8 | 18,83 | 0,33 |
| 9 | 24,1 | 0,28 |

Jadi, terdapat beberapa jalur evakuasi optimal yang dapat dilalui oleh pengungsi yaitu pada *individu 1* = $\{(v_1, v_3), (v_2, v_3), (v_3, v_5)\}$ dengan total waktu evakuasi pengungsi adalah 18,83 dan kemacetan selama proses evakuasi adalah 0,33 di mana volume aliran pengungsi pada masing-masing busur adalah 3, 2, dan 5.



Gambar 3.3 Contoh Kasus Jalur Evakuasi yang Optimal

Pada contoh kasus ini, belum tentu hasil tersebut optimal karena baru dilakukan satu kali iterasi. Untuk mencapai hasil yang optimal perlu dilakukan lebih dari satu kali iterasi.