

BAB III

METODOLOGI PENELITIAN

Bab ini membahas deskripsi masalah, tahapan penelitian, model optimisasi, dan menyajikan teknik penyelesaian model.

3.1 Deskripsi Masalah

Penelitian ini membahas masalah penentuan lintasan terpendek *fuzzy*, yaitu lintasan di mana panjang lintasannya berupa bilangan *fuzzy*. Misalkan diberikan n lokasi di mana waktu tempuh antara lokasi-lokasi tidak pasti. Karena ketidakpastiannya, maka waktu tempuh tersebut memenuhi keanggotaan *fuzzy*. Lintasan terpendek adalah lintasan dari sumber ke tujuan dengan total waktu tempuh semimumimum mungkin. Lintasan terpendek yang dibahas dalam penelitian ini adalah lintasan terpendek dari satu sumber ke satu tujuan. Penelitian ini akan menyelesaikan masalah lintasan terpendek *fuzzy* dengan menggunakan algoritma genetika.

Tahap pertama yang dilakukan pada penelitian ini adalah merepresentasikan jalanan sebagai graf berbobot. Selanjutnya algoritma genetika akan digunakan untuk mencari lintasan terpendek. Pada penelitian ini data yang dibutuhkan adalah waktu tempuh antar lokasi yang merupakan data sekunder yang didapatkan dari aplikasi *Google maps* dan *Google Earth*. Jarak adalah panjang lintasan yang ditempuh suatu benda dari titik satu ke titik lainnya dalam satuan km. Waktu tempuh diperoleh dari estimasi waktu yang ada di *google maps* untuk sampai ke tujuan dalam satuan menit. Sedangkan *Google Earth* digunakan untuk memilih jalan utama.

3.2 Tahapan Penelitian

Tahapan yang dilakukan penelitian ini adalah sebagai berikut:

1. Studi pustaka

Pada tahapan ini dilakukan studi pustaka terkait penelitian, termasuk penelitian terdahulu tentang graf, algoritma genetika, *fuzzy*, dan penelitian

serupa tentang pencarian lintasan optimisasi melalui referensi buku dan artikel jurnal.

2. Pengumpulan Data

Tahapan awal penelitian pengumpulan data yang diperlukan untuk analisis. Data yang dikumpulkan mencakup jarak dari sumber ke tujuan serta waktu yang dibutuhkan untuk pergi dari titik sumber ke titik tujuan.

3. Representasi Graf

Pada tahap ini, jalanan di Bandung Raya akan direpresentasikan ke dalam bentuk graf yang terdiri dari simpul, sisi, dan bobotnya. Dimana bobot dari graf merupakan nilai indeks yang diperoleh dengan menggunakan *fuzzy* berdasarkan tingkat kemacetan.

4. Pembangunan Model Optimisasi

Pada tahap ini, akan dibangun model optimisasi berupa model *shortest path problem* dengan mendefinisikan asumsi - asumsi model.

5. Penyelesaian Model Optimisasi

Pada tahap ini, model optimisasi akan diselesaikan dengan menggunakan *ranking fuzzy* dan algoritma genetika.

6. Validasi

Model dan teknik penyelesaian akan divalidasi dengan cara membandingkan hasil dari perhitungan manual dengan komputasi pada contoh kasus. Apabila hasilnya belum valid, maka tahapan akan kembali ke proses pemodelan. Apabila hasilnya valid, maka proses selanjutnya akan dilanjutkan ke tahapan implementasi.

7. Implementasi

Model optimisasi dan teknik penyelesaian selanjutnya akan diimplementasikan pada masalah penentuan lintasan wisata di Bandung Raya.

8. Penarikan kesimpulan

Pada tahap ini dilakukan penarikan kesimpulan berdasarkan analisis hasil implementasi.

3.3 Asumsi dan Model Optimisasi

Model optimisasi masalah lintasan terpendek *fuzzy* diformalisan merujuk pada model Awaliyah (2024). Asumsi-asumsi yang digunakan dalam pemodelan adalah sebagai berikut:

1. hanya terdapat satu sumber,
2. jalan yang digunakan adalah jalan utama, dan
3. semua ruas jalan yang dipilih dianggap bisa dilalui.

Tahapan pertama pemodelan adalah mendefinisikan parameter dan variabel keputusan yang digunakan dalam model. Pertama-tama, jaringan jalan direpresentasikan sebagai graf *fuzzy*. Graf *fuzzy* adalah graf dengan bobot berupa bilangan *fuzzy*. Graf *fuzzy* direpresentasikan sebagai graf dengan simpul sumber merepresentasikan titik keberangkatan dan simpul akhir merepresentasikan simpul tujuan. Sisi pada graf yang menghubungkan dua simpul merepresentasikan jalan. Bobot pada sisi menyatakan waktu tempuh. Penelitian ini menggunakan bobot *fuzzy* karena waktu tempuh pada suatu perjalanan dapat berbeda-beda atau tidak dapat dipastikan dengan jelas karena dipengaruhi oleh padatnya kendaraan yang berbeda-beda setiap waktu. Ketidakjelasan waktu tempuh ini diatasi dengan konsep *fuzzy*. Masalah lintasan terpendek dimodelkan sebagai graf berarah $G = (V, E)$. Graf berarah digunakan dalam penelitian karena ruas jalan yang ada memiliki arah, yaitu satu arah dan dua arah. Pada penelitian ini bilangan *fuzzy* yang digunakan adalah bilangan *fuzzy* trapesium.

Diberikan graf berarah *fuzzy* $G = (V, E)$. Misalkan:

V : Himpunan semua simpul

S : Himpunan simpul sumber, dengan $S \subseteq V$.

T : Himpunan simpul tujuan, dengan $T \subseteq V$.

$\tilde{c}_{ij}$: bobot *fuzzy* dari di antara simpul i dan j

Variabel keputusan model didefinisikan untuk menentukan lintasan dari simpul i ke simpul j dipilih. Variabel keputusan ini didefinisikan sebagai berikut:

$$x_{ij} = \begin{cases} 1, & \text{Jika lintasan dari simpul } i \text{ ke simpul } j \text{ dipilih} \\ 0, & \text{Yang lainnya.} \end{cases}$$

Fungsi tujuan dari model didefinisikan untuk meminimumkan total biaya *fuzzy* dari setiap simpul sumber ke simpul tujuan. Fungsi tujuan ini diekspresikan sebagai berikut:

Meminimumkan:

$$\sum_{i \in V} \sum_{j \in V} \tilde{c}_{ij} x_{ij}$$

Adapun kendala-kendala dari model optimisasi adalah sebagai berikut:

1. Setiap simpul i hanya dapat menjadi simpul asal paling banyak satu kali.

Kendala tersebut diekspresikan sebagai berikut:

$$\sum_{j \in V} x_{ij} \leq 1, \quad i \in V$$

2. Setiap simpul j hanya dapat dikunjungi paling banyak satu kali. Kendala tersebut diekspresikan sebagai berikut:

$$\sum_{i \in V} x_{ij} \leq 1, \quad j \in V$$

3. Setiap lintasan berawal dari simpul sumber. Kendala tersebut diekspresikan sebagai berikut:

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

4. Setiap lintasan berakhir pada simpul tujuan. Kendala tersebut diekspresikan sebagai berikut:

$$\sum_{i \in V} x_{ij} = 1, \quad j \in T$$

5. Untuk simpul lainnya, jumlah aliran yang masuk = jumlah aliran keluar.

Kendala tersebut diekspresikan sebagai berikut:

$$\sum_{i \in V} x_{il} - \sum_{j \in V} x_{ij} = 0, \quad l \in (V - S \cup T)$$

Batasan variabel menentukan bahwa keputusan x_{ij} bernilai biner. Batasan ini dituliskan sebagai berikut:

$$x_{ij} \in \{0,1\}$$

Selengkapnya, model optimisasi pencarian lintasan terpendek adalah sebagai berikut:

Meminimumkan:

$$z = \sum_{i=1} \sum_{j=1} \tilde{c}_{ij} x_{ij}$$

Dengan kendala:

$$\begin{aligned} \sum_{j \in V} x_{ij} &\leq 1 & i \in V \\ \sum_{i \in V} x_{ij} &\leq 1, & j \in V \\ \sum_{j \in V} x_{ij} &= 1, & i \in S \\ \sum_{i \in V} x_{ij} &= 1, & j \in T \\ \sum_{i \in V} x_{il} - \sum_{j \in V} x_{ij} &= 0, & l \in (V - S \cup T) \\ x_{ij} &\in \{0,1\} \end{aligned}$$

3.4 Teknik Penyelesaian

Pada subbab ini akan dibahas teknik penyelesaian masalah lintasan terpendek *fuzzy* dengan menggunakan algoritma genetika. Algoritma genetika (GA) dipilih dalam penelitian ini karena telah terbukti efektif dalam menyelesaikan berbagai permasalahan optimisasi lintasan terpendek sebagaimana dibuktikan oleh penelitian sebelumnya (Lin dkk., 2020). Solusi dari permasalahan direpresentasikan sebagai kromosom. Setiap kromosom tersusun dari rangkaian gen. Proses algoritma genetika dimulai dengan inisialisasi parameter, pengkodean kromosom, pembangkitan populasi awal. Kromosom – kromosom ini mewakili individu-individu dalam populasi yang merupakan himpunan solusi dari permasalahan. Setelah itu, himpunan solusi dievaluasi menggunakan nilai *fitness*. Semakin tinggi nilai *fitness*-nya, semakin besar peluang himpunan solusi tersebut mendekati solusi optimal. Individu yang baik dapat bereproduksi dengan individu lain dalam populasinya. Individu baru yang dihasilkan disebut keturunan yang

membawa beberapa sifat dari induknya. Individu yang tidak bisa bereproduksi akan dieliminasi. Dengan cara ini, populasi akan berkembang menjadi generasi baru dengan karakteristik semakin baik dan diharapkan menghasilkan kromosom baru dengan nilai *fitness* lebih rendah dari sebelumnya. Teknik penyelesaian ini merujuk pada hasil penelitian Lin (2020). Pada penelitian tersebut membahas masalah lintasan terpendek *fuzzy* dalam jaringan *fuzzy* dengan menggunakan algoritma genetika.

Bobot graf diasumsikan merupakan bilangan *fuzzy* trapesium (a_1, a_2, a_3, a_4) . Selanjutnya bilangan *fuzzy* tersebut akan dikonversi menjadi bilangan *crisp* menggunakan fungsi *ranking* trapesium dengan Persamaan (2.14).

Algoritma genetika akan digunakan untuk menentukan lintasan terpendek *fuzzy* dari sebuah simpul sumber ke sebuah simpul tujuan dari graf *fuzzy*. Berikut adalah tahapan algoritma genetika dalam menentukan lintasan terpendek:

1. Inisialisasi parameter

Sebelum menjalankan algoritma genetika perlu ditetapkan terlebih dahulu nilai parameter - parameter yang akan digunakan. Parameter – parameter tersebut adalah:

a. Ukuran populasi (*pop size*)

Parameter ini mengatur banyaknya kromosom yang akan diproses dalam setiap iterasi.

b. Banyaknya generasi (iterasi)

Parameter ini menentukan banyaknya siklus atau iterasi yang akan diproses dalam setiap iterasi.

c. Probabilitas Crossover (P_c)

P_c dinyatakan sebagai bilangan real antara 0 dan 1. Parameter ini menentukan kemungkinan terjadinya *crossover* pada suatu kromosom.

d. Probabilitas mutasi (P_m)

P_m dinyatakan sebagai bilangan real antara 0 dan 1. Parameter ini mengatur kemungkinan terjadinya mutasi pada suatu kromosom.

2. Pengkodean kromosom

Dalam algoritma genetika, calon solusi diwakili oleh kromosom. Sejumlah kromosom disebut populasi. Dalam penelitian ini, kromosom direpresentasikan

dalam bentuk *permutation encoding*. Kromosom dari *permutation encoding* berupa kumpulan dari nilai integer yang mewakili suatu populasi dalam sebuah urutan. Panjang kromosom menyatakan banyaknya lokasi. Elemen setiap gen menyatakan sebuah lokasi. Gambar 3.1 adalah contoh representasi kromosom dengan lima gen.

5	3	1	2	4
---	---	---	---	---

Gambar 3.1 Representasi Kromosom

3. Pembangkitan populasi awal

Populasi awal dibentuk dengan cara membangkitkan kromosom secara acak sebanyak *pop size*.

4. Evaluasi Nilai *fitness*

Nilai *fitness* digunakan untuk menentukan baik atau tidaknya individu yang direpresentasikan oleh kromosom. Nilai *fitness* ini dijadikan acuan dalam mencapai nilai optimal. Pada penelitian ini, fungsi *fitness* dari kromosom k_i adalah:

$$f(k_i) = \sum_{i=1}^{n-1} P(\tilde{A}) \quad (3.1)$$

$$fitness = \frac{1}{f(k_i)} \quad (3.2)$$

Dengan:

$P(\tilde{A}_i)$: *Ranking* untuk himpunan bilangan *fuzzy trapesium*

$\tilde{A}_i$: Himpunan Bilangan *fuzzy trapezium*/ bobot *fuzzy* sisi (b_i, b_{i+1}) .

5. Seleksi

Dalam algoritma genetika, seleksi digunakan untuk memilih lebih banyak kromosom yang memiliki nilai *fitness* lebih tinggi dan memilih lebih sedikit kromosom memiliki nilai *fitness* lebih kecil. Dalam penelitian ini menggunakan seleksi dengan metode *ranking*. Seleksi *ranking* adalah suatu metode seleksi yang dilakukan dengan memberi urutan pada populasi sesuai dengan nilai *fitnessnya*. Nilai *fitness* terbesar diberi nilai 1, yang kedua terbesar diberi nilai 2 dan seterusnya sampai yang terakhir diberi nilai K (jumlah kromosom dalam populasi).

6. Kawin silang (*crossover*)

Metode *crossover* digunakan untuk menggabungkan kembali dua kromosom induk untuk menghasilkan dua kromosom anak baru. Dalam metode *crossover*, beberapa sifat dari dua kromosom induk diwariskan ke dua kromosom anaknya yang menjelaskan solusi yang berbeda dari kromosom induknya. Pada *crossover* terdapat parameter probabilitas *crossover* (P_c) di mana nilai P_c antara 0 sampai 1. Sebuah bilangan acak dipilih antara 0 sampai 1. Jika bilangan acak yang dipilih kurang dari P_c maka dilakukan *crossover* sedangkan jika bilangan acak dipilih lebih dari P_c maka tidak terjadi *crossover*.

Penelitian ini menggunakan metode *partially mapped crossover* (PMX). PMX merupakan salah satu metode *crossover* yang digunakan pada algoritma genetika terutama dalam permasalahan optimisasi yang melibatkan permutasi seperti masalah lintasan terpendek. PMX digunakan untuk menggabungkan dua individu dalam suatu populasi untuk menghasilkan dua keturunan yang diwarisi dari induknya, namun tetap menjaga keutuhan dan sifat dari solusi sebelumnya. Ilustrasi metode *partially mapped crossover* dapat dilihat pada Gambar 3.2

1. Pilih posisi untuk menentukan substring secara acak

Kromosom 1	1	2	4	3	5
------------	---	---	---	---	---

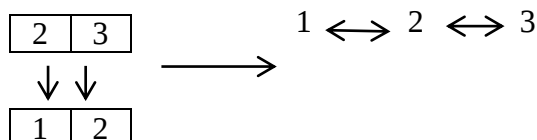
Kromosom 4	2	3	5	1	4
------------	---	---	---	---	---

2. Tukar substring di antara induk

Proto-Offspring 1	2	3			
-------------------	---	---	--	--	--

Proto Offspring 2	1	2			
-------------------	---	---	--	--	--

3. Tentukan hubungan mapping



4. Tentukan kromosom *offspring* mengacu pada hubungan mapping

Offspring 1	2	3	4	1	5
-------------	---	---	---	---	---

Offspring 2	1	2	5	3	4
-------------	---	---	---	---	---

Gambar 3.2 Ilustrasi Metode Partially Mapped Crossover

Langkah-langkah metode *partially mapped crossover* yaitu:

- Pilih dua posisi kromosom dari populasi secara acak. Substring yang berada dalam dua posisi ini dinamakan daerah pemetaan.
- Tukar dua substring antar induk untuk menghasilkan *proto-offspring*.
- Tentukan hubungan pemetaan diantara dua daerah pemetaan.
- Tentukan kromosom keturunan mengacu pada hubungan pemetaan.

7. Mutasi

Setelah proses *crossover*, langkah selanjutnya adalah proses mutasi berdasarkan parameter probabilitas mutasi (P_m). Probabilitas mutasi ini menentukan seberapa sering mutasi akan terjadi pada kromosom. Jika probabilitas mutasi terlalu kecil, maka semakin kecil kemungkinan individu baru dengan karakteristik berbeda muncul yang mengakibatkan banyak individu dengan gen baru yang mungkin merupakan solusi terbaik tetapi tidak masuk evaluasi. Jika probabilitas mutasi terlalu besar, maka akan terlalu banyak terjadinya mutasi yang menyebabkan keturunan yang termutasi kehilangan kemiripan dengan induknya.

Pada penelitian ini, metode mutasi yang digunakan adalah *swap mutation*. Pada metode ini, proses mutasi dilakukan dengan cara menukar gen yang dipilih secara acak dengan gen sesudahnya. Jika gen tersebut berada di akhir kromosom maka ditukar dengan gen pertama (Khoirussoleh, 2014). Ilustrasi *swap mutation* dapat dilihat pada Gambar 3.3

Sebelum mutasi:

Kromosom 4	2	3	5	1	4
------------	---	---	---	---	---

Setelah mutasi:

Kromosom 4	2	3	1	5	4
------------	---	---	---	---	---

Gambar 3.3 Ilustrasi *Swap Mutation*

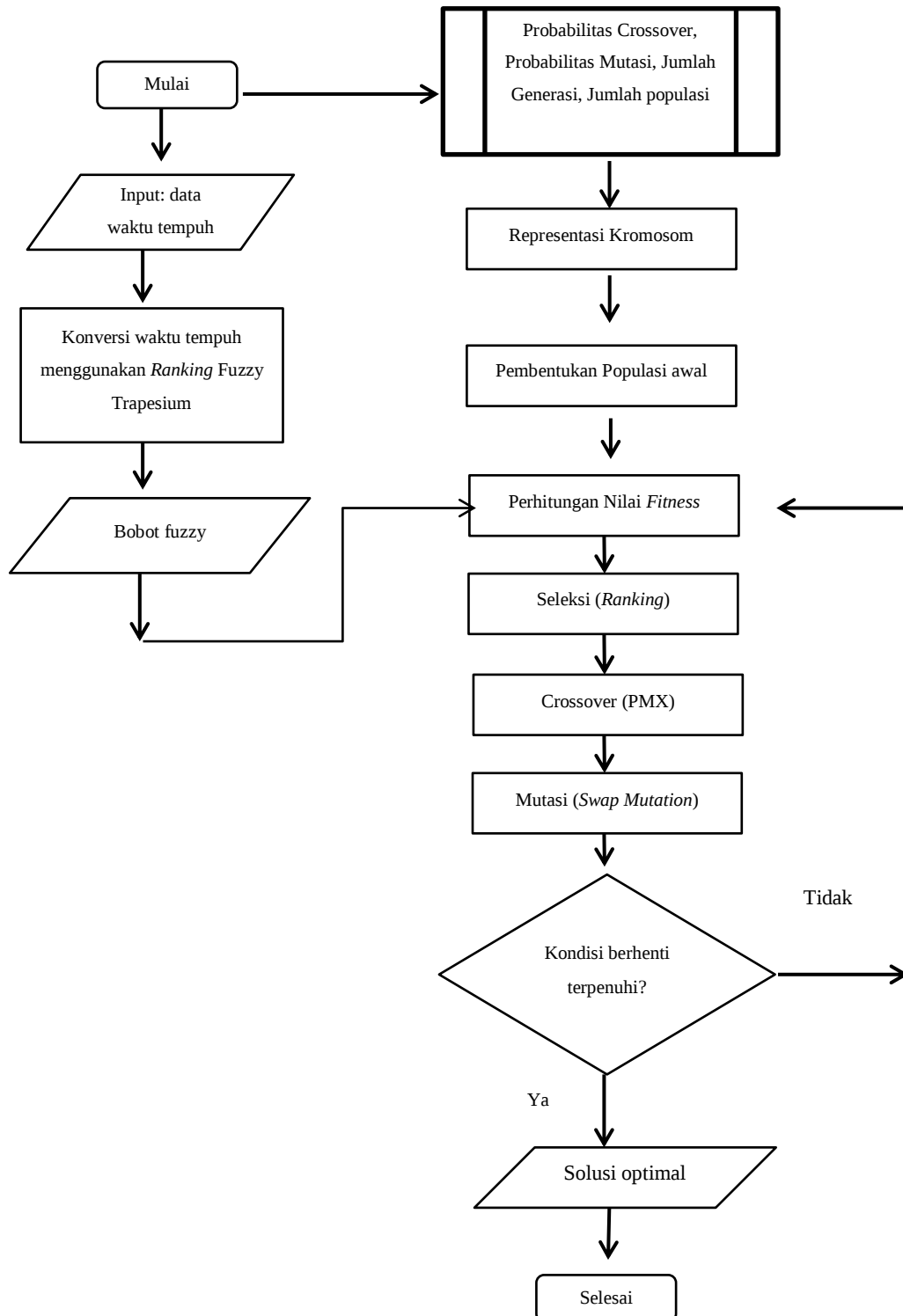
Langkah-langkah *swap mutation* adalah sebagai berikut:

- Memilih kromosom yang akan dimutasi secara acak
- Memilih dua nilai dari gen pada kromosom
- Selanjutnya menukar posisi kedua gen yang telah dipilih

8. Kriteria pemberhentian

Lakukan langkah awal hingga akhir sebanyak parameter jumlah generasi yang dilakukan. Pencarian dengan algoritma genetika akan berhenti hingga iterasinya telah mencapai jumlah maksimum generasi yang ditentukan. Selanjutnya analisis hasil dari iterasi per generasinya dan ambil nilai *fitness* terbesar.

Tahapan-tahapan GA dalam menyelesaikan lintasan terpendek *fuzzy* digambarkan pada flowchart pada Gambar 3.4.



Gambar 3.4 Flowchart Cara Kerja Algoritma genetika