

BAB III

METODOLOGI PENELITIAN

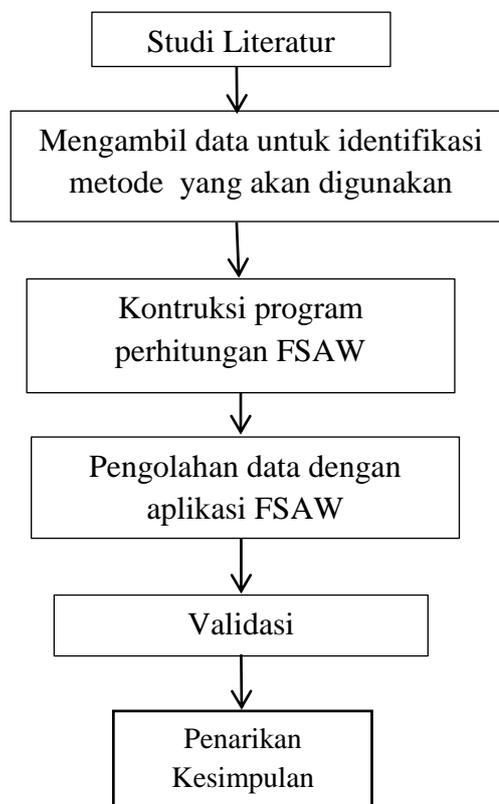
Pada bab ini akan dijelaskan mengenai metodologi penelitian, *Fuzzy Simple Additive Weighing* (FSAW), dan kontruksi program yang akan dibuat dengan aplikasi Java.

3.1 Metodologi Penelitian

Pada penelitian ini metode yang digunakan untuk pengambilan keputusan adalah *Fuzzy Simple Additive Weighing* (FSAW). Metode ini dipilih karena data yang digunakan berupa data *fuzzy*, di mana data seperti ini memiliki kesamaran dalam penafsirannya. Berikut ini langkah-langkah penelitian yang dilakukan :

1. Studi literatur mengenai Logika *Fuzzy*, *Simple Additive Weighing*, dan *Fuzzy Simple Additive Weighing*.
2. Mengambil data dari angket yang disebar peneliti menggunakan aplikasi *Google Form*.
3. Pengolahan data dilakukan dengan menggunakan bantuan bahasa pemrograman *Javafx* yaitu dengan membuat program Aplikasi Pengambilan Keputusan dengan Metode FSAW.
4. Setelah program aplikasi selesai kemudian dilakukan proses validasi, yaitu perhitungan secara manual sebagai pembanding.

Langkah-langkah dari metodologi penelitian di atas disajikan dalam bentuk *flowchart* berikut ini :



Gambar 3.1 *Flowchart* Metodologi Penelitian

3.2 Fuzzy Multi Attribute Decision Making (FMADM)

Pada FMADM alternatif-alternatif sudah diketahui dan ditentukan sebelumnya. Pengambil keputusan dapat menyeleksi beberapa kemungkinan alternatif dengan sumber, tujuan, dan koefisien yang bernilai fuzzy. FMADM mengharuskan pengambil keputusan menentukan prioritas atau ranking berdasarkan kriteria yang diberikan. Secara umum, FMADM memiliki suatu tujuan tertentu, yang dapat diklasifikasikan dalam 2 tipe, yaitu (Simoès-Marques, 2000), yaitu menyeleksi alternatif dengan atribut (kriteria) dengan ciri-ciri terbaik; dan mengklasifikasi alternatif berdasarkan peran tertentu.

Untuk menyelesaikan masalah FMADM, dibutuhkan 2 tahap, yaitu :

1. Membuat *rating* pada setiap alternatif berdasarkan agregasi derajat kecocokan pada semua kriteria;
2. Meranking semua alternatif untuk mendapatkan alternatif terbaik. Ada 2 cara yang dapat digunakan dalam proses perankingan, yaitu melalui *defuzzy* atau

melalui relasi preferensi *fuzzy*. Metode *defuzzy* dilakukan dengan pertama-tama membuat bentuk *crisp* dari bilangan *fuzzy*, proses perankingan didasarkan atas bilangan *crisp* tersebut; model ini memang mudah untuk diimplementasikan, namun kita sangat dimungkinkan untuk kehilangan beberapa informasi terutama yang menyangkut ketidakpastian. Penggunaan relasi preferensi *fuzzy* lebih menjamin ketidakpastian yang melekat pada bilangan *fuzzy* hingga proses perankingan (Hwang, 1992).

3.3 Fuzzy Simple Additive Weighting (FSAW)

Fuzzy SAW pertama kali dikembangkan oleh Baas dan Kwarknaak (Hwang, 1992). Pada fuzzy MADM masalah, nilai kriteria, dan bobot yang berhubungan berupa data fuzzy yang berbentuk variabel linguistik. Metode ini adalah gabungan antara Metode *Fuzzy* dan Metode *Simple Additive Weighting*. Dimana pada penyelesaiannya akan ada konversi ke bentuk bilangan fuzzy, baru kemudian dikonversi ke bilangan *crisp* (Kusumadewi & dkk, 2006). Setelah dikonversi maka dihitung dengan Metode *Simple Additive Weighting*. Dibawah ini akan diuraikan langkah-langkah dari Metode *Fuzzy Simple Additive Weighting* :

1. Pilih alternatif (X_i , $i = 1, 2, \dots, n$) dan atribut yang akan digunakan.
2. Berikan penilaian alternatifnya (A) berdasarkan atribut dan tentukan nilai bobot (W) dari atribut sesuai bilangan fuzzynya.
3. Tentukan atribut yang menguntungkan dan atribut biayanya.
4. Konversikan bilangan fuzzy pada setiap penilaian ke bilangan *crisp* yang berkisar 0 sampai 1.
5. Pengkontruksian matriks keputusan (D) yang sudah dikonversikan.

$$D_{ij} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} \\ \vdots & \vdots & \ddots & \vdots \\ d_{i1} & d_{i2} & \cdots & d_{ij} \end{bmatrix}$$

dimana d_{ij} merupakan penilaian terhadap alternatif a_i berdasarkan kriteria c_j , $i = 1, 2, 3, \dots, m$ dan $j = 1, 2, 3, \dots, n$.

6. Pengkontruksian matriks keputusan yang ternormalisasi (R) .

$$R_{ij} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1j} \\ \vdots & \vdots & \ddots & \vdots \\ r_{i1} & r_{i2} & \cdots & r_{ij} \end{bmatrix}$$

dimana

$$r_{ij} \begin{cases} \frac{X_{ij}}{\max_i X_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{i}{\min_i X_{ij}} & \text{Jika } j \text{ adalah atribut biaya} \end{cases}$$

r_{ij} menyatakan rating kinerja ternormalisasi dari alternatif X_i pada atribut C_j ; $i=1,2,3,\dots,m$ dan $j=1,2,3,\dots,n$.

7. Perhitungan skor dari setiap alternatif.

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

8. Pilih nilai V_i yang lebih besar dimana mengindikasikan bahwa alternatif X_i lebih terpilih.

3.4 Konstruksi Program

Seperti telah dikemukakan sebelumnya, bahwa untuk membantu mempermudah proses pengambilan keputusan dengan multi alternatif dan multi kriteria, penulis merancang program aplikasi dengan bahasa pemrograman Java. Rancangan desain program Aplikasi *Fuzzy Simple Additive Weighting* bertujuan untuk memberikan kemudahan dalam proses perhitungan pengambilan keputusan dengan menggunakan Metode *Fuzzy SAW*.

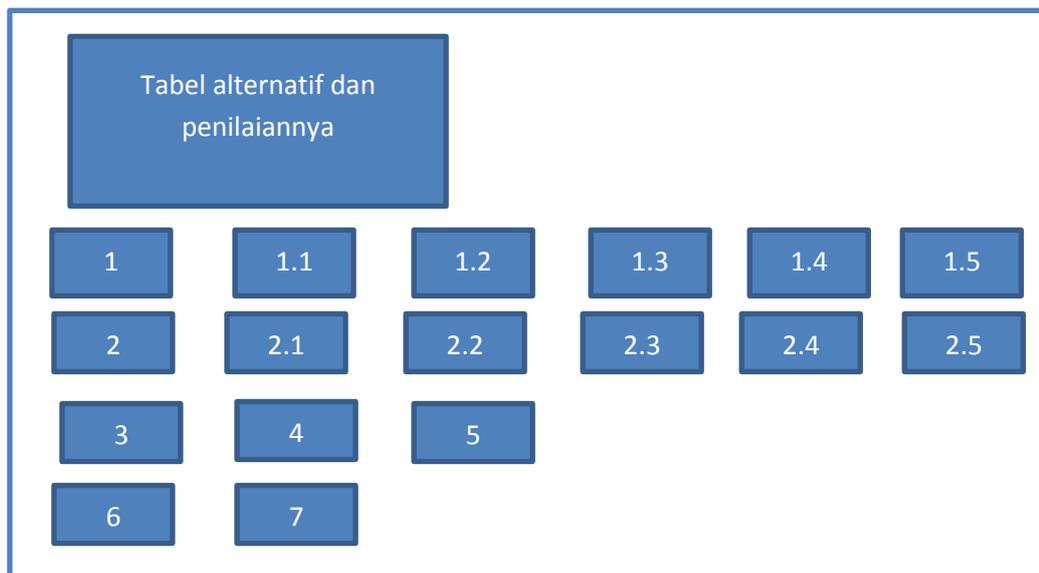
Rancangan desain program Aplikasi *Fuzzy SAW* berfungsi untuk memperoleh alternatif terbaik berdasarkan kriteria-kriteria yang sesuai dengan keinginan dan kebutuhan konsumen. Pada rancangan desain program Aplikasi *Fuzzy SAW* ini terdapat input yang diperlukan untuk diproses sehingga menghasilkan output yang diinginkan dalam hal ini adalah alternatif terbaik.

- Input/masukan
 - Alternatif
 - Nilai alternatif
 - Nilai bobot
- Output/keluaran
 - Matriks keputusan
 - Matriks keputusan yang ternormalisasi

- Alternatif yang terpilih
- Skor masing-masing alternatif

3.5 Desain Aplikasi

Subbab ini akan menjelaskan mengenai bagaimana rancangan desain program Aplikasi *Fuzzy SAW*. Pembuatan desain program Aplikasi *Fuzzy SAW* ini berguna untuk mempermudah penulis dalam pembuatan program Aplikasi *Fuzzy SAW* tersebut. Desain input untuk program Aplikasi *Fuzzy SAW* adalah sebagai berikut.



Gambar 3. 1 Desain Program Aplikasi Fuzzy SAW

Keterangan untuk desain program Aplikasi *Fuzzy SAW* adalah sebagai berikut:

- Kotak 1 : Input alternatif yang merupakan tempat wisata
- Kotak 1.1 – 1.5 : Input nilai atribut setiap alternatif, yaitu harga, fasilitas, image destinasi, atraksi, dan akses.
- Kotak 2 – 2.5 : Input bobot atribut
- Kotak 3 : Tombol matriks keputusan untuk memunculkan tabel matriks keputusan
- Kotak 4 : Tombol matriks normalisasi untuk menampilkan tabel matriks normalisasi
- Kotak 5 : Tombol reset untuk menghapus semua data
- Kotak 6 : Tombol cari untuk menampilkan hasil alternatif yang terbaik
- Kotak 7 : Menampilkan alternatif terbaik

- **Desain tampilan matriks keputusan**

Desain tampilan output berupa tabel matriks keputusan pada program Aplikasi *Fuzzy SAW* adalah sebagai berikut :

Gambar 3. 2 Desain Tampilan Output Matriks Keputusan

- **Desain tampilan matriks normalisasi**

Sedangkan desain tampilan output berupa tabel matriks normalisasi pada program Aplikasi *Fuzzy SAW* adalah sebagai berikut:

Gambar 3. 3 Desain Tampilan Output Matriks Normalisasi

3.6 Prosedur Program

Setelah membuat desain program Aplikasi *Fuzzy SAW*, hal selanjutnya penulis membuat algoritma program Aplikasi *Fuzzy SAW*. Secara garis besar algoritma program Aplikasi *Fuzzy SAW* adalah sebagai berikut:

1. Input alternatif (x)
2. Input nilai atribut alternatif (z)
3. Input nilai bobot (y)
4. Baca nilai alternatif (z)
5. Konversi (z) ke bilangan *crisp*
6. Buat matriks keputusan (R), berdasarkan hasil konversi
7. Baca nilai bobot (y)
8. Konversi (y) ke bilangan *crisp*
9. Hitung matriks normalisasi R dengan perumusan sebagai berikut:

Dengan, i menyatakan jumlah alternatif

Pengulangan sebanyak i

$$\text{Jika } (j = 0), \text{ maka } R(0, i) = \frac{\min(x(0,i))}{x(0,i)}$$

$$\text{Jika } (j! = 0), \text{ maka } R(0, i) = \frac{x(:,i)}{\max(x(0,i))}$$

10. Menampilkan matriks normalisasi
11. Hitung score dari setiap alternatif dengan perumusan sebagai berikut:

$$V(i) = \text{sum}(y(i) \cdot R(j, :))$$

12. Cari alternatif terbaik

$$\text{hasil} = \max(V)$$

13. Cari indeks alternatif terbaik
14. Tampilkan alternatif terbaik