

## BAB 2 LANDASAN TEORI

### 2.1 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) adalah sebuah sistem yang memiliki peran untuk membantu seseorang atau suatu organisasi dalam mengambil keputusan terhadap masalah yang sedang mereka alami [16]. Basis dari sistem tersebut adalah komputerisasi [17].

Menurut Moore dan Chang, Sistem Pendukung Keputusan adalah suatu sistem yang memiliki kegunaan dalam orientasi perencanaan waktu, orientasi keputusan, dapat mendukung analisa data Ad Hoc, dan suatu model untuk keputusan [18]. Adanya Sistem Pendukung Keputusan ini dapat membantu individual maupun perusahaan dalam menyelesaikan masalah yang dialami baik terstruktur maupun tidak terstruktur [17].

### 2.2 Simple Additive Weighting

*Simple Additive Weighting* (SAW), yang juga sering dikenal dengan nama metode penjumlahan terbobot, adalah salah satu metode algoritma yang dapat digunakan untuk menyelesaikan masalah sistem pendukung keputusan [19]. *Simple Additive Weighting* mendapatkan julukan tersebut karena metode perhitungan algoritma ini menjumlahkan setiap bobot terhadap nilai-nilai alternatif kriteria [20]. Tujuan dari perhitungan tersebut adalah untuk dapat membandingkan nilai rekomendasi akhir setiap alternatif kriteria sehingga mendapatkan hasil perhitungan rekomendasi yang lebih baik dan seimbang [21].

Adapun langkah penyelesaian dalam menggunakannya adalah sebagai berikut [22],

1. Menentukan  $C_j$ , yaitu kriteria-kriteria dari benda yang akan direkomendasikan.
2. Apabila kriteria memiliki nilai yang tidak kuantitatif maka cari nilai alternatif untuk kriteria tersebut, yaitu  $A_j$ .
3. Menentukan bobot ( $W$ ) dari setiap kriteria yang menjadi acuan untuk rekomendasi.

4. Membuat matriks keputusan( $X$ ) dengan nilai masing-masing kriteria( $C_j$ ) yang digunakan. Apabila ada kriteria yang memiliki nilai tidak kuantitatif maka akan digunakan nilai alternatif( $A_i$ ) dari kriteria tersebut. Hal ini dilakukan agar matriks keputusan dapat digunakan untuk perhitungan matriks. Contoh matriks keputusan dapat dilihat pada 2.1.

$$X = \begin{vmatrix} r_{11} & r_{12} & \dots & r_{1j} \\ \vdots & & & \vdots \\ r_{i1} & r_{i2} & \dots & r_{ij} \end{vmatrix} \quad (2.1)$$

5. Langkah selanjutnya adalah menormalisasikan matriks keputusan( $X$ ) yang telah didapat. Normalisasi setiap nilai matriks( $R_{ij}$ ) didapatkan dengan cara membagi setiap nilai kategori( $C_j$ ) atau alternatif( $A_i$ ) dengan nilai maksimal yang ada di kategori atau alternatif tersebut apabila kategori adalah tipe keuntungan. Apabila kategori adalah tipe biaya maka nilai minimum kategori akan dibagi dengan nilai kategori( $C_j$ ) atau alternatif( $A_i$ ) satu-persatu. Untuk lebih jelasnya dapat dilihat dirumus sebagai berikut,

- (a) Apabila  $j$  adalah nilai tipe keuntungan(*Benefit*) akan menggunakan rumus 2.2.

$$R_{ij} = \frac{X_{ij}}{\text{Max}X_{ij}} \quad (2.2)$$

- (b) Apabila  $j$  adalah nilai tipe biaya(*Cost*) akan menggunakan rumus 2.3.

$$R_{ij} = \frac{\text{Min}X_{ij}}{X_{ij}} \quad (2.3)$$

Keterangan:

- (a) Kriteria termasuk tipe keuntungan(*benefit*) apabila semakin besar nilai kategori tersebut maka akan semakin menguntungkan kepada pengambil keputusan, dan sebaliknya apabila nilai yang lebih rendah semakin baik bagi pengambil keputusan maka kriteria tersebut termasuk tipe biaya(*cost*).
- (b) Jika kriteria adalah tipe keuntungan(*benefit*) maka masing-masing nilai akan dibagi dengan nilai maksimal yang ada di kategori tersebut, dan

sebaliknya apabila kriteria adalah tipe biaya(*cost*) maka nilai minimum yang ada di kriteria akan dibagi dengan masing-masing nilai kriteria tersebut.

6. Hasil perhitungan normalisasi masing-masing nilai dengan rumus yang terdapat di rumus 2.2 dan rumus 2.3 di setiap nilai kriteria akan menghasilkan matriks normalisasi(*R*) yang dapat dilihat pada 2.4.

$$R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1j} \\ \vdots & & & \vdots \\ r_{i1} & r_{i2} & \dots & r_{ij} \end{pmatrix} \quad (2.4)$$

7. Setelah matriks normalisasi(*R*) didapat maka selanjutnya adalah perkalian matriks antara bobot kriteria(*W*) dengan matriks yang telah dinormalisasikan dimana nilai setiap kolom yang ada di matriks dikali dengan setiap bobot. Kemudian seluruh nilai kolom kriteria setiap baris matriks akan dijumlahkan untuk mendapatkan nilai preferensi(*V*). Rumus untuk perhitungan tersebut dapat dilihat pada 2.5.

$$V_i = \sum_{j=i}^n W_j R_{ij} \quad (2.5)$$

Langkah terakhir adalah mengurutkan setiap nilai preferensi(*V<sub>i</sub>*) yang didapat untuk mendapatkan ranking rekomendasi[11].

### 2.3 System Development Life Cycle

*System Develoment Life Cycle*(SDLC) adalah metodologi siklus perkembangan perangkat lunak yang biasa digunakan untuk memastikan perangkat yang dikembangkan dapat dibuat secara efektif dengan cara membagi proses pengembangan menjadi pekerjaan bertahap[23]. SDLC dapat berfungsi sebagai sarana komunikasi antara pengembang dengan klien mengenai kebutuhan perangkat lunak dan juga dapat mengurangi biaya yang dibutuhkan[24].

Terdapat 6 tahap kerja yang ada di dalam metodologi SDLC. 6 tahapan tersebut dibuat agar tim pengembang dapat fokus dalam pengembangan perangkat lunak dan memastikan kebutuhan dan biaya yang diperlukan untuk perangkat lunak yang sedang dikembangkan sesuai dengan kebutuhan klien[25]. 6 tahap itu adalah

sebagai berikut[26],

1. Perencanaan

Pada tahap ini tim pengembang akan melakukan komunikasi dengan klien untuk menentukan kebutuhan yang dimiliki perangkat lunak. Perhitungan biaya juga akan dilakukan di tahap ini untuk memastikan biaya yang diperlukan tidak melewati ekspektasi klien.

2. Analisis

Tahap ini berguna untuk memahami permasalahan dalam pengembangan perangkat lunak dan mendefinisikan kebutuhan klien yang diperlukan untuk perangkat lunak yang sedang dikembangkan.

3. Rancangan

Pada tahap ini tim pengembang akan merancang perangkat lunak yang akan dikembangkan. Perancangan tersebut dapat berupa *flowchart* maupun sketsa antarmuka perangkat lunak.

4. Implementasi

Pada tahap ini tim pengembang akan menerapkan rancangan yang telah dibuat di tahap rancangan.

5. Pengujian & Integrasi Setelah perangkat lunak berhasil dikembangkan maka akan dilakukan pengujian untuk memastikan perangkat lunak sesuai dengan kebutuhan klien dan tidak memiliki *bug* maupun *error*. Setelah pengujian selesai maka tim pengembang akan mengintegrasikan perangkat lunak ke sistem yang dimiliki oleh klien.

6. Pemeliharaan Tahap terakhir adalah pemeliharaan perangkat lunak. Pemeliharaan ini biasanya dilakukan secara berkala dalam kurung waktu tertentu untuk memastikan perangkat lunak dapat berjalan dengan baik.

#### 2.4 Skala Likert

Skala Likert adalah skala penelitian dalam ilmu statistik yang digunakan untuk mengukur pendapat seseorang baik itu positif maupun negatif[27]. Nama skala likert itu sendiri diambil dari ahli psikologi sosial amerika serikat yang bernama Rensis Likert[28]. Skala yang digunakan dalam skala likert biasanya memiliki 5 tingkat tanggapan baik dalam bentuk positif maupun negatif[27].

Apabila tingkat tanggapan adalah positif maka tingkat 1 adalah sangat tidak setuju dan tingkat 5 adalah sangat setuju[29]. Sebaliknya, apabila tingkat tanggapan adalah negatif maka tingkat 1 adalah sangat setuju dan tingkat 5 adalah sangat tidak setuju[29].

Skala likert memiliki beberapa keunggulan dibandingkan dengan skala lain seperti skala thurstone[29]. Dalam jumlah *item* yang sama, kereliabilitan skala likert lebih tinggi dari skala thurstone[29]. Selain itu skala likert juga dinyatakan dapat memberikan pendapat responden yang akurat berdasarkan pertanyaan yang diberikan[27].

