

## **BAB VII**

### ***WEIGHTED PRODUCT (WP)***

#### **I. TUJUAN**

- Memahami metode WP dalam menentukan keputusan
- Mampu mengaplikasikan metode WP dalam Sistem Pendukung Keputusan menggunakan Python

#### **II. PENDAHULUAN**

##### ***Metode Weighted Product***

*Weighted Product* (WP) adalah metode untuk menyelesaikan permasalahan pengambilan keputusan yang memiliki banyak atribut dengan cara mengalikan nilai atribut tiap alternatif yang sudah dipangkatkan dengan bobot atributnya (Bozorg-Haddad et al., 2021). Metode ini mengevaluasi beberapa alternatif terhadap sekumpulan atribut atau kriteria yang tidak saling bergantung satu sama lain.

##### ***Langkah-Langkah Penyelesaian Metode Weighted Product***

Menurut Bozorg-Haddad et al (2021), sistem pendukung keputusan dengan menggunakan metode WP dapat dilakukan dengan langkah-langkah berikut:

##### **1. Mendefinisikan permasalahan pengambilan keputusan**

Tahapan ini meliputi penentuan kriteria ( $C_i$ ) yang akan dijadikan acuan dalam pengambilan keputusan, bobot ( $W_i$ ) masing-masing kriteria, kriteria mana yang termasuk *benefit* dan *cost*, alternatif ( $A_i$ ) apa saja yang akan dipertimbangkan, dan juga pembuatan matriks keputusan berdasarkan nilai atribut (*rating* kecocokan) dari tiap alternatif pada tiap kriteria.

##### **2. Normalisasi bobot**

Karena metode ini menggunakan pemangkatan nilai atribut dengan bobotnya, nilai akan menjadi sangat besar jika bobot bernilai besar. Oleh karena itu, bobot dari tiap kriteria harus dinormalisasi dengan rumus berikut:

$$\text{Bobot Ternormalisasi} = \frac{\text{Bobot kriteria}}{\text{Total seluruh bobot kriteria}}$$

Tiap bobot yang sudah dinormalisasi akan bernilai lebih dari 0 dan kurang dari 1, dan jika seluruh bobot dijumlahkan, hasilnya akan sama dengan 1.

### 3. Menggabungkan preferensi alternatif

Tahapan ini meliputi perhitungan nilai tiap alternatif ( $S_i$ ) dengan cara mengalikan semua nilai atribut yang telah dipangkatkan dengan bobot ternormalisasi. Rumus perhitungannya sebagai berikut:

$$S_i = \prod_{j=1}^n (r_{i,j})^{w_j}$$

Keterangan:

$S$  : Preferensi alternatif

$r$  : Nilai kriteria/*rating* kecocokan

$w$  : Bobot kriteria ternormalisasi

$i$  : Indeks alternatif

$j$  : Indeks kriteria

$n$  : Jumlah kriteria

Kemudian, hasil dari vektor  $S$  akan dinormalisasi kembali dengan cara membagi nilai  $S_i$  dengan total dari semua nilai  $S$ . Sebenarnya, penentuan peringkat dapat dilakukan tanpa melakukan normalisasi vektor  $S$ , tetapi kita tidak bisa mengukur seberapa baik alternatif tersebut dibandingkan dengan alternatif lain. Normalisasi kembali dapat mengatasi permasalahan tersebut. Berikut adalah rumus untuk normalisasi vektor  $S$ :

$$V_i = \frac{S_i}{\sum_{i=1}^m S_i}$$

Keterangan:

$V$  : Preferensi alternatif ternormalisasi

$S$  : Preferensi alternatif

$i$  : Indeks alternatif

$m$  : Jumlah alternatif

Alternatif dengan nilai  $V$  terbesar adalah alternatif terbaik untuk penyelesaian masalah pemilihan keputusan berdasarkan metode WP.

### III. LANGKAH PRAKTIKUM

Contoh studi kasus dikutip dari buku Sri et al (2006), suatu perusahaan di Daerah Istimewa Yogyakarta (DIY) ingin membangun sebuah gudang yang akan digunakan sebagai tempat menyimpan sementara hasil produksinya. Ada 3 lokasi yang akan menjadi alternatif, yaitu:

$A_1$  : Ngemplak

$A_2$  : Kalasan

$A_3$  : Kotagede

Pada kasus ini, terdapat 5 kriteria yang dijadikan sebagai acuan dalam pengambilan keputusan, yaitu:

$C_1$  : Jarak dengan pasar terdekat (km)

$C_2$  : Kepadatan penduduk di sekitar lokasi (orang/km<sup>2</sup>)

$C_3$  : Jarak dari pabrik (km)

$C_4$  : Jarak dari gudang yang sudah ada (km)

$C_5$  : Harga tanah (1.000.000 Rp/m<sup>2</sup>)

Selain itu, tingkat kepentingan (bobot) dari tiap kriteria menggunakan nilai dalam skala 1-5 dengan keterangan sebagai berikut:

1 : Sangat rendah

2 : Rendah

3 : Cukup

4 : Tinggi

5 : Sangat tinggi

Nilai atribut dari tiap alternatif telah dirangkum dalam tabel berikut:

Alternatif	Kriteria				
	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>
A <sub>1</sub>	0,75	2000	18	50	500
A <sub>2</sub>	0,50	1500	20	40	450
A <sub>3</sub>	0,90	2050	35	35	800

Berdasarkan kasus tersebut, langkah pertama yang harus kita lakukan adalah menentukan kriteria mana yang termasuk *benefit* dan *cost* dan juga alasannya.

Kriteria	Benefit/Cost	Alasan
C <sub>1</sub>	Cost	Jarak dari pasar diutamakan sedekat mungkin untuk menekan biaya distribusi
C <sub>2</sub>	Cost	Gudang diutamakan jauh dari perumahan penduduk agar proses distribusi barang tidak mengganggu warga
C <sub>3</sub>	Cost	Jarak dari pabrik diutamakan sedekat mungkin untuk menekan biaya distribusi
C <sub>4</sub>	Benefit	Jarak dari gudang lain diutamakan sejauh mungkin untuk mengurangi resiko kerugian serentak akibat bencana seperti banjir, kebakaran, gempa bumi, dll.
C <sub>5</sub>	Cost	Harga tanah diutamakan yang murah untuk menekan biaya yang harus dikeluarkan perusahaan.

Dari informasi yang sudah disimpulkan sebelumnya, buat *list* dalam Python untuk menyimpan nama-nama alternatif, nilai matriks keputusan, nilai *cost/benefit* tiap kriteria, dan juga bobot dari tiap kriteria.

```
# alternatif
a = ["Ngemplak","Kalasan","Kotagede"]
# matriks keputusan
x = [
    [0.75,2000,18,50,500],
    [0.50,1500,20,40,450],
    [0.90,2050,35,35,800]
]
# penentuan benefit/cost tiap kriteria
# cost = -1, benefit = 1
k = [-1,-1,-1,1,-1]
# bobot dari tiap kriteria
w = [5,3,4,4,2]
```

Selanjutnya, lakukan normalisasi terhadap bobot dari tiap kriteria. Ambil dimensi dari matriks keputusan  $x$  untuk dijadikan parameter *looping* untuk perhitungan-perhitungan berikutnya.

```
# ambil dimensi dari matriks keputusan
# m = jumlah alternatif, n = jumlah kriteria
m = len(x)
n = len(x[0])

# normalisasi bobot
w_norm = [x/sum(w) for x in w]
```

Kode di atas akan menghasilkan bobot baru dengan perhitungan manual sebagai berikut:

$$W_1 = \frac{5}{5 + 3 + 4 + 4 + 2} = 0,28$$

$$W_2 = \frac{3}{5 + 3 + 4 + 4 + 2} = 0,17$$

$$W_3 = \frac{4}{5 + 3 + 4 + 4 + 2} = 0,22$$

$$W_4 = \frac{4}{5 + 3 + 4 + 4 + 2} = 0,22$$

$$W_5 = \frac{2}{5 + 3 + 4 + 4 + 2} = 0,11$$

Jumlah total dari bobot yang telah dinormalisasi adalah 1, terbukti dengan perhitungan di bawah ini:

$$\begin{aligned}\sum_{i=1}^5 W_i &= W_1 + W_2 + W_3 + W_4 + W_5 \\ \sum_{i=1}^5 W_i &= 0,28 + 0,17 + 0,22 + 0,22 + 0,11 \\ \sum_{i=1}^5 W_i &= 1\end{aligned}$$

Setelah normalisasi bobot, lakukan perhitungan untuk vektor  $S$ . Vektor  $S$  adalah hasil perkalian dari nilai atribut yang dipangkatkan dengan bobot kriteria yang sudah dinormalisasi.

```
# menghitung vektor S
s = [1]*m
for i in range(m):
    for j in range(n):
        s[i] = s[i]*x[i][j]**(k[j]*w_norm[j])
```

Kode di atas akan menghasilkan vektor S yang perhitungannya dilakukan sebagai berikut:

$$S_1 = 0,75^{-0,28} \times 2000^{-0,17} \times 18^{-0,22} \times 50^{0,22} \times 500^{-0,11} = 0,1920$$

$$S_2 = 0,50^{-0,28} \times 1500^{-0,17} \times 20^{-0,22} \times 40^{0,22} \times 450^{-0,11} = 0,2120$$

$$S_3 = 0,90^{-0,28} \times 2050^{-0,17} \times 35^{-0,22} \times 35^{0,22} \times 800^{-0,11} = 0,1375$$

Selanjutnya, normalisasi vektor S untuk mendapatkan vektor V. Tampilkan hasil dari vektor V dan juga alternatif terbaik berdasarkan nilai terbesar di vektor V.

```
# menghitung vektor V
v = [x/sum(s) for x in s]

# ambil indeks nilai terbesar dari vektor V
max_index = v.index(max(v))

# tampilkan hasil vektor v dan alternatif terbaik
print("Hasil matriks V adalah {}".format(v))
print("Lokasi gudang terbaik berdasarkan metode WP adalah {}".format(a[max_index]))
```

Kode tersebut menghitung normalisasi vektor S yang disimpan pada vektor V, dengan perhitungan sebagai berikut:

$$V_1 = \frac{0,1920}{0,1920 + 0,2120 + 0,1375} = 0,3546$$

$$V_2 = \frac{0,2120}{0,1920 + 0,2120 + 0,1375} = 0,3916$$

$$V_3 = \frac{0,1375}{0,1920 + 0,2120 + 0,1375} = 0,2539$$

Nilai terbesar dari vektor V terdapat pada  $V_2$ , yaitu sebesar 0,3916. Oleh karena itu, alternatif terbaik berdasarkan metode WP adalah alternatif kedua, yaitu Kalasan.

#### **IV. TUGAS PRAKTIKUM**

1. Buatlah program latihan praktikum di atas dengan menggunakan inputan serta buatlah studi kasus dengan data lebih dari 8 dan selesaikan dengan program di atas.
2. Buatlah program no.1 dengan menggunakan GUI dari *library* Streamlit.