

BAB VIII

ANALYTICAL HIERARCHY PROCESS (AHP)

I. TUJUAN

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

II. PENDAHULUAN

Metode Analytical Hierarchy Process

Analytical Hierarchy Process (AHP) dikembangkan oleh Dr. Thomas L. Saaty, seorang ahli matematika dari Universitas Pittsburgh, pada awal 1970-an. AHP dirancang untuk menangkap persepsi manusia terkait permasalahan tertentu melalui proses sistematis yang menghasilkan skala preferensi dari berbagai alternatif. Metode ini cocok untuk menyusun model permasalahan yang tidak terstruktur, baik untuk masalah kuantitatif, berbasis penilaian, maupun situasi kompleks dengan data statistik yang minim atau kualitatif. AHP sering digunakan *dalam decision support systems* untuk membantu pengguna membuat keputusan yang lebih tepat berdasarkan persepsi, pengalaman, atau intuisi (Sasongko et al., 2017).

Metode AHP memiliki beberapa kelebihan dibandingkan metode lain, diantaranya yaitu:

- A. Kesatuan (*Unity*), AHP dapat menjadikan sebuah permasalahan yang luas dan tidak terstruktur menjadi sebuah model yang fleksibel dan tergolong mudah dipahami.
- B. Kompleksitas (*Complexity*), AHP dapat memecahkan suatu permasalahan yang tergolong kompleks melalui sebuah pendekatan sistem dan pengintegrasian secara deduktif.
- C. Saling ketergantungan (*Inter Dependence*), AHP dapat diimplementasikan pada elemen-elemen sistem yang tidak saling berhubungan dan tidak memerlukan hubungan linier.

- D. Struktur Hirarki (*Hierarchy Structuring*), AHP dapat mewakili pemikiran alamiah yang cenderung mengelompokkan elemen sistem ke dalam level-level yang berbeda dimana masing-masing level berisikan elemen yang serupa.
- E. Pengukuran (*Measurement*), AHP menyediakan sebuah skala pengukuran dan metode untuk mendapatkan nilai prioritas masing-masing elemen kriteria.
- F. Konsistensi (*Consistency*), AHP mempertimbangkan suatu nilai konsistensi yang logis dalam penilaian yang digunakan untuk menentukan suatu prioritas.
- G. Sintesis (*Synthesis*), AHP mengarah pada perkiraan keseluruhan dalam hirarki untuk mengetahui seberapa diinginkannya masing-masing alternatif yang ada.
- H. *Trade Off*, AHP mempertimbangkan prioritas relatif masing-masing faktor yang terdapat pada sistem sehingga orang mampu memilih alternatif terbaik berdasarkan tujuan sesuai dengan yang diharapkan.
- I. Penilaian dan Konsensus (*Judgement and Consensus*), AHP tidak mengharuskan adanya suatu konsensus, tapi menggabungkan hasil dari sebuah penilaian yang berbeda.
- J. Pengulangan Proses (*Process Repetition*), AHP mampu membuat orang menyaring definisi dari suatu permasalahan dan mengembangkan penilaian serta pengertian mereka melalui proses pengulangan.

Selain kelebihan di atas, metode AHP juga memiliki beberapa kekurangan, diantaranya yaitu:

- A. Metode AHP memiliki ketergantungan pada input utamanya. Input utama yang dimaksud adalah berupa persepsi atau penafsiran seorang ahli sehingga dalam hal ini melibatkan subyektifitas sang ahli selain itu juga model menjadi tidak berarti jika ahli tersebut memberikan penilaian yang salah.
- B. Metode AHP ini hanya metode matematis. Tanpa ada pengujian secara statistik berdasarkan data historis permasalahan yang telah terjadi sebelumnya, sehingga tidak ada batas kepercayaan dan informasi pendukung yang kuat dari kebenaran model yang terbentuk.

Langkah-Langkah Penyelesaian Metode *Analytical Hierarchy Process*

Metode AHP memiliki beberapa tahapan sebelum mencapai keputusan terbaik. Tahapan-tahapan tersebut dapat dirangkum sebagai berikut:

1. Membangun hirarki dari masalah

Tahapan ini meliputi penentuan tujuan dari penyelesaian masalah, kriteria (C_i) apa yang menjadi acuan pengambilan keputusan, bobot (W_i) dari tiap kriteria, dan juga alternatif (A_i) apa saja yang akan akan dipertimbangkan.

2. Menghitung bobot dari alternatif maupun kriteria

Tahapan ini dimulai dengan membuat perbandingan berpasangan (*pairwise comparison*) dari kriteria dan juga alternatif. Hasil akhirnya berupa matriks 2 dimensi yang tiap elemennya berisi nilai perbandingan dari variabel di baris dan kolom tersebut. Jika variabel A lebih baik atau lebih penting dari B, hasil perbandingannya akan lebih besar dari 1, sedangkan jika variabel A lebih buruk atau lebih tidak penting dari variabel B, maka hasil perbandingannya akan berada dalam *range* $0 < \text{perbandingan} < 1$. Jika kedua variabel sama bagus, sama penting atau dibandingkan dengan dirinya sendiri, maka hasil perbandingannya adalah 1.

Selanjutnya, matriks tersebut dinormalisasi dengan cara membagi tiap elemen dengan total dari elemen-elemen dalam kolom yang sama. Berikut adalah rumus normalisasi matriks perbandingan:

$$r_{i,j} = \frac{x_{i,j}}{\sum_{i=1}^m x_{i,j}}$$

Keterangan:

- r : Elemen matriks perbandingan ternormalisasi
- x : Elemen matriks perbandingan sebelum normalisasi
- i : Indeks baris
- j : Indeks kolom
- m : Jumlah baris

Jumlah total dalam tiap kolom matriks yang telah dinormalisasi harus bernilai 1, atau memenuhi persamaan di bawah ini:

$$\sum_{i=1}^m r_{i,j} = 1$$

Keterangan:

r : Elemen matriks perbandingan ternormalisasi

i : Indeks baris

j : Indeks kolom

m : Jumlah baris

Selanjutnya, bobot dari kriteria dan alternatif akan dihitung berdasarkan rata-rata dari tiap baris matriks perbandingan yang sudah dinormalisasi sebelumnya. Rumus perhitungannya sebagai berikut:

$$W_i = \frac{\sum_{j=1}^n r_{i,j}}{n}$$

Keterangan:

W : Bobot

r : Elemen matriks perbandingan ternormalisasi

i : Indeks baris

j : Indeks kolom

n : Jumlah kolom

3. Uji validitas bobot

Bobot yang sudah dihitung harus diuji validitasnya melalui perhitungan *Consistency Index* (CI) dan *Consistency Ratio* (CR). Tahap pertama untuk mencari CI dan CR adalah dengan menghitung *Consistency Value* (CV) yang didapatkan dari penjumlahan hasil kali seluruh bobot dengan elemen dalam satu baris, kemudian dibagi dengan bobot baris tersebut. Rumusnya adalah sebagai berikut:

$$CV_i = \frac{\sum_{j=1}^n W_j \times r_{i,j}}{W_i}$$

Keterangan:

CV : *Consistency Value*

W : Bobot

i : Indeks baris
 j : Indeks kolom
 n : Jumlah kolom

Kemudian, cari nilai *Eigen* (λ) yang diperoleh dari rata-rata nilai CV. Berikut adalah rumus perhitungannya:

$$\lambda = \frac{CV_i}{\sum_{i=1}^m CV_i}$$

Keterangan:

λ : Nilai *Eigen*
 CV : *Consistency Value*
 i : Indeks baris
 m : Jumlah baris

Selanjutnya, hitung nilai CI dengan menggunakan nilai *Eigen* yang didapatkan dari tahap sebelumnya. Rumus perhitungannya adalah sebagai berikut:

$$CI = \frac{\lambda - n}{n - 1}$$

Keterangan:

CI : *Consistency Index*
 λ : Nilai *Eigen*
 n : Jumlah kriteria/alternatif (tergantung sedang menghitung CI dari matriks perbandingan kriteria atau alternatif)

Kemudian, ambil nilai *Random Consistency Index* (RI) yang berupa nilai baku berdasarkan nilai n pada perhitungan sebelumnya. Berikut adalah tabel acuan nilai RI:

n	2	3	4	5	6	7	8	9	10	11	12	13	14
RI	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,51	1,53	1,54	1,56	1,57

Terakhir, penentu validitas bobot adalah nilai CR yang didapatkan dari rumus berikut:

$$CR = \frac{CI}{RI}$$

Keterangan:

CR : Consistency Ratio

CI : Consistency Index

RI : Random Consistency Index

Bobot dianggap konsisten jika nilai $CR \leq 0,1$. Jika nilai CR melebihi 0,1, pembobotan harus diulang hingga konsisten.

4. Penghitungan *ranking* secara keseluruhan

Ranking dihitung dengan cara menjumlahkan hasil kali bobot alternatif pada tiap kriteria dengan bobot kriterianya. Alternatif yang memiliki total terbesar adalah alternatif terbaik berdasarkan metode AHP.

$$\omega_i = \sum_{i=1}^m WC_i \times WA_i$$

Keterangan:

ω : Bobot akhir

WC : Bobot kriteria

WA : Bobot kriteria dari alternatif

i : Indeks kriteria

m : Jumlah kriteria

III. LANGKAH PRAKTIKUM

Contoh studi kasus, Dinda akan merayakan ulang tahunnya yang ke-17. Orang tuanya memperbolehkan Dinda untuk meminta sebuah hadiah pada mereka. Dinda bilang bahwa dia menginginkan sebuah motor dengan alternatif sebagai berikut:

A₁ : Yamaha

A₂ : Honda

A₃ : Suzuki

A₄ : Kawasaki

Kemudian, Dinda ingin memilih motor terbaik dengan kriteria-kriteria berikut sebagai acuan dalam pengambilan keputusan:

C₁ : Gaya

C₂ : Keandalan

C₃ : Keekonomisan bahan bakar

Untuk membuat keputusan, Dinda menerapkan metode AHP pada permasalahan tersebut. Dinda menggunakan skala preferensi 1 sampai 4 untuk mendefinisikan tingkat kepentingannya.

Intensitas	Keterangan
1	X dan Y sama penting/baik
2	X sedikit lebih penting/baik dari Y
3	X lebih penting/baik dari Y
4	X jauh lebih penting/baik dari Y

Dinda sudah menyimpulkan beberapa hal berdasarkan preferensinya. Berikut adalah preferensi Dinda terhadap kriteria yang digunakan:

1. Keandalan 2 kali lebih penting dari gaya
2. Gaya 3 kali lebih penting dari keekonomisan bahan bakar
3. Keandalan 4 kali lebih penting dari keekonomisan bahan bakar

Dinda juga sudah mengamati alternatif-alternatif yang ada, sehingga ia dapat menyimpulkan hal berikut:

1. Preferensi gaya dari yang terbaik

Kawasaki > Honda > Yamaha > Suzuki

2. Preferensi keandalan dari yang terbaik

Honda > Yamaha > Kawasaki > Suzuki

3. Nilai keekonomisan bahan bakar berdasarkan data

Yamaha : 60

Honda : 80

Suzuki : 60

Kawasaki : 80

Berdasarkan informasi di atas, kita dapat membuat matriks perbandingan berpasangan dari kriteria penilaian terlebih dahulu.

$$MPB_k = \begin{bmatrix} 1 & 1/2 & 3 \\ 2 & 1 & 4 \\ 1/3 & 1/4 & 1 \end{bmatrix}$$

10/3 7/4 8

Kemudian, normalisasi matriks tersebut dengan membagi nilai tiap elemen dengan total dari kolom tersebut (yang dicetak tebal).

$$MPB_{k_{norm}} = \begin{bmatrix} \frac{3}{10} & \frac{2}{7} & \frac{3}{8} \\ \frac{6}{10} & \frac{4}{7} & \frac{4}{8} \\ \frac{1}{10} & \frac{1}{7} & \frac{1}{8} \end{bmatrix}$$

1 1 1

Untuk menghitung normalisasi matriks perbandingan berpasangan, buat fungsi dengan bahasa Python sebagai berikut:

```
import numpy as np

def normalize_comparation(M):
    """Fungsi untuk normalisasi matriks perbandingan berpasangan"""
    # menjumlahkan elemen-elemen dalam kolom yang sama
```



```

sumCol = M.sum(axis = 0) # axis 0 = kolom, 1 = baris
norm = M / sumCol
return norm

```

Setelah itu, hitung bobot kriteria dengan cara merata-ratakan tiap baris pada matriks perbandingan yang sudah dinormalisasi.

$$Wk = \begin{bmatrix} 0,3202 \\ 0,5572 \\ 0,1226 \end{bmatrix}$$

Perhitungan bobot dapat dilakukan menggunakan fungsi yang tertera pada kode berikut:

```

def weight(M):
    """Fungsi untuk menghitung bobot"""
    # menghitung rata-rata baris
    return np.mean(M, axis=1)

```

Bobot selanjutnya akan dicek validitasnya dengan mencari nilai CV, *Eigen*, CI, RI, dan CR. Nilai CV didapatkan dengan menjumlahkan perkalian bobot dengan tiap baris dari matriks perbandingan yang sudah dinormalisasi, lalu dibagi dengan jumlah kriterianya (n). Nilai *Eigen* (λ) didapatkan dari rata-rata nilai CV.

$$CV = \begin{bmatrix} 3,0186 \\ 3,0299 \\ 3,0065 \end{bmatrix}$$

$$\lambda = \overline{CV}$$

$$\lambda = 3,0183$$

Nilai CI didapatkan dengan menggunakan nilai *Eigen* dan juga n. Kemudian, nilai RI untuk nilai n sebesar 3 didapatkan dari tabel nilai baku RI yang sudah dilampirkan sebelumnya.

$$CI = \frac{\lambda - n}{n - 1}$$

$$CI = \frac{3,0183 - 3}{3 - 1}$$

$$CI = 0,0092$$

$$RI = 0,58$$

Terakhir, nilai CR didapatkan dari pembagian nilai CI dengan RI. Jika nilai $CR \leq 0,1$, bobot dianggap konsisten. Sebaliknya, jika nilai CR melebihi 0,1, ulangi pembobotan sampai bobot konsisten.

$$CR = \frac{CI}{RI}$$

$$CR = \frac{0,0092}{0,58}$$

$$CR = 0,0158$$

Karena nilai $CR \leq 0,1$, maka bobot dari kriteria dianggap konsisten. Pengecekan validitas bobot juga dapat dilakukan dengan fungsi yang tertera pada kode berikut:

```
def validity_check(M,W):
    """Fungsi untuk mengecek validitas bobot"""
    n = len(M)
    RI = {
        2: 0.00, 3: 0.58, 4: 0.90, 5: 1.12, 6: 1.24, 7: 1.32, 8: 1.41, 9: 1.45,
        10: 1.51, 11: 1.53, 12: 1.54, 13: 1.56, 14: 1.57
    }
```

```
# menghitung CV
CV = M @ W / W # operator @ mengalikan matriks sesuai dengan prinsipnya
print(f"CV: {CV}\n")

# menghitung nilai eigen
eigen = np.mean(CV)
print(f"Eigen: {eigen}\n")

# mencari nilai RI
print(f"RI: {RI[n]}\n")
```

```

# menghitung CI
CI = (eigen - n) / (n - 1)
print(f"CI: {CI}\n")

# menghitung CR
CR = CI / RI[n]
print(f"CR: {CR}\n")

if CR <= 0.1:
    print("Bobot valid!\n")
else:
    print("Bobot tidak valid!\n")

```

Pengaplikasian fungsi-fungsi tersebut pada matriks kriteria dapat dilihat pada kode di bawah ini:

```

# alternatif
A = ["Yamaha", "Honda", "Suzuki", "Kawasaki"]

print("=== Kriteria ===")

# matriks perbandingan berpasangan kriteria
# urutan: gaya, keandalan, keekonomisan
MPBk = np.array([
    [1, 1/2, 3],
    [2, 1, 4],
    [1/3, 1/4, 1]
])
print(f"Matriks perbandingan: {MPBk}\n")

# normalisasi
MPBk_norm = normalize_comparation(MPBk)
print(f"Matriks perbandingan setelah normalisasi: {MPBk_norm}\n")

# perhitungan bobot
Wk = weight(MPBk_norm)
print(f"Bobot kriteria: {Wk}\n")

# uji validitas bobot
validity_check(MPBk, Wk)

```

Selanjutnya, lakukan hal yang sama untuk tiap kriteria pada alternatif. Kriteria pertama adalah gaya, yang hasilnya adalah sebagai berikut:

$$MPBg = \begin{bmatrix} 1 & 1/2 & 2 & 1/3 \\ 2 & 1 & 3 & 1/2 \\ 1/2 & 1/3 & 1 & 1/4 \\ 3 & 2 & 4 & 1 \end{bmatrix} \quad MPBg_{norm} = \begin{bmatrix} 2/13 & 3/23 & 2/10 & 4/25 \\ 4/13 & 6/23 & 3/10 & 6/25 \\ 1/13 & 2/23 & 1/10 & 3/25 \\ 6/13 & 12/23 & 4/10 & 12/25 \end{bmatrix}$$

$$Wg = \begin{bmatrix} 0,1611 \\ 0,2771 \\ 0,0960 \\ 0,4658 \end{bmatrix} \quad CV = \begin{bmatrix} 4,0160 \\ 4,0416 \\ 4,0152 \\ 4,0513 \end{bmatrix}$$

$$\lambda = 4,0310$$

$$RI = 0,9$$

$$CI = 0,0103$$

$$CR = 0,0115$$

```
print("=== Gaya ===")

# matriks perbandingan berpasangan kriteria gaya dari tiap alternatif
# urutan: yamaha, honda, suzuki, kawasaki
MPBg = np.array([
    [1, 1/2, 2, 1/3],
    [2, 1, 3, 1/2],
    [1/2, 1/3, 1, 1/4],
    [3, 2, 4, 1]
])
print(f"Matriks perbandingan: {MPBg}\n")

# normalisasi
MPBg_norm = normalize_comparation(MPBg)
print(f"Matriks perbandingan setelah normalisasi: {MPBg_norm}\n")

# perhitungan bobot
Wg = weight(MPBg_norm)
```

```
print(f"Bobot: {Wg}\n")

# uji validitas bobot
validity_check(MPBg,Wg)
```

Kriteria kedua adalah keandalan. Berdasarkan pengamatan yang telah dilakukan oleh Dinda, hasil perhitungannya adalah sebagai berikut:

$$MPBa = \begin{bmatrix} 1 & 1/2 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 1/3 & 1/4 & 1 & 1/2 \\ 1/2 & 1/3 & 2 & 1 \end{bmatrix} \quad MPBa_{norm} = \begin{bmatrix} 6/23 & 6/25 & 3/10 & 4/13 \\ 12/23 & 12/25 & 4/10 & 6/13 \\ 2/23 & 3/25 & 1/10 & 1/13 \\ 3/23 & 4/25 & 2/10 & 2/13 \end{bmatrix}$$

$$Wa = \begin{bmatrix} 0,2771 \\ 0,4658 \\ 0,0960 \\ 0,1610 \end{bmatrix} \quad CV = \begin{bmatrix} 4,0416 \\ 4,0513 \\ 4,0152 \\ 4,0160 \end{bmatrix}$$

$$\lambda = 4,0310$$

$$RI = 0,9$$

$$CI = 0,0103$$

$$CR = 0,0115$$

```
print("=== Keandalan ===")

# matriks perbandingan berpasangan kriteria keandalan dari tiap alternatif
# urutan: yamaha, honda, suzuki, kawasaki
MPBa = np.array([
    [1, 1/2, 3, 2],
    [2, 1, 4, 3],
    [1/3, 1/4, 1, 1/2],
    [1/2, 1/3, 2, 1]
])
print(f"Matriks perbandingan: {MPBa}\n")
```

```

# normalisasi
MPBa_norm = normalize_comparation(MPBa)
print(f"Matriks perbandingan setelah normalisasi: {MPBa_norm}\n")

# perhitungan bobot
Wa = weight(MPBa_norm)
print(f"Bobot: {Wa}\n")

# uji validitas bobot
validity_check(MPBa,Wa)

```

Kriteria terakhir adalah keekonomisan bahan bakar. Berdasarkan data yang dikumpulkan oleh Dinda, hasil perhitungannya adalah sebagai berikut:

$$MPBe = \begin{bmatrix} 1 & \frac{60}{80} & 1 & \frac{60}{80} \\ \frac{80}{60} & 1 & \frac{80}{60} & 1 \\ 1 & \frac{60}{80} & 1 & \frac{60}{80} \\ \frac{80}{60} & 1 & \frac{80}{60} & 1 \end{bmatrix} \quad MPBe_{norm} = \begin{bmatrix} 14/63 & 7/35 & 14/63 & 7/35 \\ 18/63 & 10/35 & 18/63 & 10/35 \\ 14/63 & 7/35 & 14/63 & 7/35 \\ 18/63 & 10/35 & 18/63 & 10/35 \end{bmatrix}$$

$$We = \begin{bmatrix} 0,2143 \\ 0,2857 \\ 0,2143 \\ 0,2857 \end{bmatrix} \quad CV = \begin{bmatrix} 4 \\ 4 \\ 4 \\ 4 \end{bmatrix}$$

$$\lambda = 4$$

$$RI = 0,9$$

$$CI = 0$$

$$CR = 0$$

```

print("=== Keekonomisan ===")

# matriks perbandingan berpasangan kriteria keandalan dari tiap alternatif
# urutan: yamaha, honda, suzuki, kawasaki

```

```

MPBe = [
    [1, 60/80, 1, 60/80],
    [80/60, 1, 80/60, 1],
    [1, 60/80, 1, 60/80],
    [80/60, 1, 80/60, 1]
]
print(f"Matriks perbandingan: {MPBe}\n")

```

```

# normalisasi
MPBe_norm = normalize_comparation(MPBe)
print(f"Matriks perbandingan setelah normalisasi: {MPBe_norm}\n")

# perhitungan bobot
We = weight(MPBe_norm)
print(f"Bobot: {We}\n")

# uji validitas bobot
validity_check(MPBe,We)

```

Setelah semua bobot divalidasi, perankingan dapat dilakukan. Bobot akhir dari masing-masing alternatif adalah hasil dari penjumlahan hasil kali bobot kriteria pada alternatif dengan bobot dari kriteria tersebut.

$$\begin{aligned}
 W_{total} &= [Wg \quad Wa \quad We] \\
 W_{total} &= \begin{bmatrix} 0,1611 & 0,2771 & 0,2143 \\ 0,2771 & 0,4658 & 0,2857 \\ 0,0960 & 0,0960 & 0,2143 \\ 0,4658 & 0,1611 & 0,2857 \end{bmatrix} \\
 W_{final} &= W_{total} \times Wk \\
 W_{final} &= \begin{bmatrix} 0,1611 & 0,2771 & 0,2143 \\ 0,2771 & 0,4658 & 0,2857 \\ 0,0960 & 0,0960 & 0,2143 \\ 0,4658 & 0,1611 & 0,2857 \end{bmatrix} \times \begin{bmatrix} 0,3202 \\ 0,5572 \\ 0,1226 \end{bmatrix} \\
 W_{final} &= \begin{bmatrix} 0,2323 \\ 0,3833 \\ 0,1105 \\ 0,2739 \end{bmatrix}
 \end{aligned}$$

Untuk melakukan perhitungan bobot akhir, diperlukan sebuah fungsi untuk mengalikan matriks bobot kriteria pada alternatif dengan matriks bobot kriteria. Berikut adalah fungsi perhitungan bobot akhir:

```
def final_weight(W_alt, W_crit):  
    """Menghitung bobot akhir tiap alternatif"""  
    # mengalikan bobot alternatif dan kriteria sesuai dengan prinsip matriks  
    W_fin = W_alt.T @ W_crit  
    return W_fin
```

Fungsi tersebut menghasilkan sebuah vektor dengan ukuran sesuai dengan jumlah alternatif. Penerapan fungsi tersebut pada kode adalah sebagai berikut:

```
# menyatukan semua bobot alternatif dalam satu matriks  
W_total = np.array([Wg, Wa, We])  
  
# menghitung bobot akhir  
W_final = final_weight(W_total, Wk)  
print(f"Bobot akhir: {W_final}\n")  
  
# memunculkan alternatif terbaik  
max_index = np.argmax(W_final)  
print(f"Alternatif terbaik adalah {A[max_index]} dengan nilai  
{W_final[max_index]}")
```

Berdasarkan hasil perhitungan, motor terbaik yang akan Dinda pilih adalah Honda dengan bobot akhir 0,3833.

IV. TUGAS PRAKTIKUM

1. Dari studi kasus pada langkah praktikum, tentukan motor terbaik jika ada kriteria baru, yaitu harga motor. Harga dari tiap motor adalah sebagai berikut:

Yamaha : 16

Honda : 30

Suzuki : 15

Kawasaki : 40

Buatlah sebuah sistem pendukung keputusan dengan menggunakan kasus lain yang dapat dilakukan dengan menggunakan metode AHP.