

**Laporan Project Akhir**  
**Sistem Pendukung Keputusan**  
**(AHP) – Kelompok 8**



Dosen :

**Dr Arna Fariza S.Kom., M.Kom.**

Disusun oleh:

**Asy Syaffa Khoirunnisa - 3121500040**

**Muhammad Daniel Krisna Halim Putra - 3121500047**

**Nabila Balqis Azzahra - 3121500059**

**D3 IT B**

**PROGRAM STUDI D3 TEKNIK INFORMATIKA**

**Politeknik Elektronika Negeri Surabaya**

**2021/2022**

# PROGRAM MADM AHP (ANALYTICAL HIERACHY PROCESS) DALAM BAHASA PYTHON

## Deskripsi Masalah :

Manajer pembelian Suroboyo Bus bertanggung jawab untuk memutuskan perusahaan mana yang tersedia untuk membeli 50 bus untuk rute baru. Setelah studi kelayakan awal dan analisis merek, model beberapa bus, dan proses yang terkait dengan pembelian, ia mempersempit daftar alternatif keputusannya menjadi tiga opsi;

1. **Proyek 1:** Pembelian model Eksekutif,
2. **Proyek 2:** Pembelian model Medium, dan
3. **Proyek 3:** Pembelian model Ekonomi.

Suroboyo Bus memutuskan bahwa kriteria berikut relevan untuk proses keputusan pembeliannya:

1. **Mil per galon (MPG)**
2. **Kenyamanan**
3. **Gaya**
4. **Harga**

## Penjelasan

### Cara menjalankan program

Sebelumnya, pada sebuah pemrograman pasti terdapat libraries. Libraries sendiri yaitu kumpulan modul terkait berisi kumpulan kode yang dapat digunakan berulang kali dalam program yang berbeda. Library ini membuat Pemrograman Python lebih sederhana dan nyaman bagi programmer. Disini kami menyediakan 2 cara mengeksekusi program, yaitu:

1. Menggunakan <https://colab.research.google.com/>. Masuk pada website pilih file kemudian upload notebook, maka tinggal upload file dengan format .ipynb,
2. Menggunakan Compiler seperti Visual Studio Code, sebelum menjalankan program, wajib wajib menginstall PIP, PIP merupakan program untuk manajemen paket di Python. Tugasnya untuk menginstal, menghapus, upgrade paket Python, dll. Masuk terlebih dahulu ke git bash/cmd, lalu ketik *python get-pip.py*. Setelah selesai install library-librarynya dengan mengetikkan *Pip install pandas* dan juga *pip install matplotlib*. Maka program sudah bisa dijalankan karena library yang dibutuhkan sudah memenuhi.

Pandas sendiri berfungsi mengolah suatu data seperti teknik join, distinct, group by, agregasi, dan teknik lainnya seperti pada SQL. Bedanya, ini dilakukan pada tabel. Kelebihan dari library ini juga dapat membaca file dari berbagai format seperti .txt, .csv, dan .tsv. Sedangkan matplotlib berfungsi untuk memvisualisasikan data secara 2D maupun 3D dan menghasilkan gambar berkualitas yang bahkan dapat kamu simpan dalam berbagai format gambar, seperti JPEG dan PNG.

```
[1] import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

**Tabel 1: Penilaian Verbal ke Peringkat Numerik**

```
Rating_dic = {'Tingkat Kepentingan': ['Mutlak lebih penting', 'Sangat lebih penting hingga mutlak lebih penting',
                                     'Sangat lebih penting', 'Lebih penting hingga sangat lebih penting',
                                     'Lebih penting', 'Cukup penting hingga lebih penting',
                                     'Cukup penting', 'Sama penting hingga cukup penting', 'Sama penting' ],
              'Peringkat Numerik': [9,8,7,6,5,4,3,2,1]}

Rating_comparison = pd.DataFrame(Rating_dic).set_index('Tingkat Kepentingan')
Rating_comparison
```

Kita deklarasikan variabel `Rating_dic` dengan menampung tipe data string lalu pada 'Tingkat Keputusan' akan menampung list data tablenya yang nantinya akan dimasukkan/dikombinasikan dengan library `panda`. Kemudian adda juga `Peringkat Numerik` yang menampung number untuk disandingkan dengan list data Tingkat kepentingan. Selanjutnya bisa dilihat pada variable `Rating_comparison`, disini variabelnya menampung `pd`, `pd` sendiri adalah `pandas` yang sudah di aliaskan. `pd` akan dihubungkan dengan class `DataFrame` yang menampung variabel `Rating_dic` tadi, lalu jangan lupa juga membuat method `set_index` yang berfungsi memberikan awalan kolom yang ditampilkan yaitu disini Tingkat Kepentingan. Maka ketika `Rating_comparison` dicetak akan menghasilkan:

Tingkat Kepentingan	
Mutlak lebih penting	9
Sangat lebih penting hingga mutlak lebih penting	8
Sangat lebih penting	7
Lebih penting hingga sangat lebih penting	6
Lebih penting	5
Cukup penting hingga lebih penting	4
Cukup penting	3
Sama penting hingga cukup penting	2
Sama penting	1

**Tabel 2: Perbandingan Berpasangan Empat Kriteria Soal Pemilihan Bus**

```
comparison_dic = {'Perbandingan Berpasangan': ['Harga-MPG', 'Harga-Kenyamanan', 'Harga-Gaya', 'MPG-Kenyamanan', 'MPG-Gaya', 'Kenyamanan-Gaya'],
                  'Kriteria Terpenting': ['Harga', 'Harga', 'Harga', 'Kenyamanan', 'Gaya', 'Gaya'],
                  'Seberapa Penting': ['Cukup penting', 'Sama penting hingga cukup penting', 'Sama penting hingga cukup penting',
                                       'Cukup penting hingga lebih penting', 'Cukup penting hingga lebih penting',
                                       'Sama penting hingga cukup penting'],
                  'Peringkat Numerik': [3, 2, 2, 4, 4, 2]}

criteria_comparison = pd.DataFrame(comparison_dic).set_index('Perbandingan Berpasangan')
criteria_comparison
```

Masuk pada tabel ke 2 kita bandingkan berpasangan dengan 4 kriteria soal pemilihan terdapat Perbandingan Berpasangan, Kriteria Terpenting, Seberapa Penting, dan juga Peringkat Numerik. Sama halnya dengan tadi, kita juga deklarasikan variable `comparison_dic` yang menampung tipe data string dan didalamnya terdapat list data dan terdapat tambahan kolom table dan perubahan nama table. Berikut outputnya:

Kriteria Terpenting		Seberapa Penting	Peringkat Numerik
Perbandingan Berpasangan			
Harga-MPG	Harga	Cukup penting	3
Harga-Kenyamanan	Harga	Sama penting hingga cukup penting	2
Harga-Gaya	Harga	Sama penting hingga cukup penting	2
MPG-Kenyamanan	Kenyamanan	Cukup penting hingga lebih penting	4
MPG-Gaya	Gaya	Cukup penting hingga lebih penting	4
Kenyamanan-Gaya	Gaya	Sama penting hingga cukup penting	2

- **Konversi Penilaian Verbal ke Peringkat Numerik**

```

Numeric_dic = {'Perbandingan Berpasangan': ['Harga', 'Kenamanan', 'Gaya', 'MPG'],
               'Harga': [1, 1/2, 1/2, 1/3],
               'Kenamanan': [2, 1, 2, 1/4],
               'Gaya': [2, 1/2, 1, 1/4],
               'MPG': [3, 4, 4, 1]}

Numeric_comparison = pd.DataFrame(Numeric_dic).set_index('Perbandingan Berpasangan')
Numeric_comparison

```

Disini kita masuk pada penilaian verbal ke peringkat numerik, Kita inisialisasikan Numeric\_dic yang berisi tipe data string dan diberikan value didalamnya. Kemudian kita deklarasi lagi Numeric\_comparison dengan isi class DataFrame yang berisi Numeric\_dic dan kita set\_index untuk menampilkan Perbandingan berpasangan pada awal table.

	Harga	Kenamanan	Gaya	MPG
Perbandingan Berpasangan				
Harga	1.000000	2.00	2.00	3
Kenamanan	0.500000	1.00	0.50	4
Gaya	0.500000	2.00	1.00	4
MPG	0.333333	0.25	0.25	1

- Menghitung bobot untuk setiap kriteria

```

criteria_weight = Numeric_comparison.copy()
for col in ['Harga', 'Kenamanan', 'Gaya', 'MPG']:
    criteria_weight[col] = criteria_weight[col] / criteria_weight[col].sum()
criteria_weight

```

Kita buat variable criteria\_weight yang mana menampung sebuah salinan objek baru dari Numeric\_comparison yang menampung DataFrame dari Numeric\_dic. Lalu kita berikan for pada column supaya perintah for yang digunakan bisa melakukan iterasi dari sebuah nilai sequence atau data koleksi pada python seperti List, Tuple, String dan lain-lain. Disini kita for pada Harga, kenyamanan, gaya, MPG yang nantinya pada criteria\_weight akan dimasukkan column dan akan di Sum() atau dijumlahkan seluruh anggota list arraynya. Dan hasilnya sebagai berikut:

	Harga	Kenamanan	Gaya	MPG
Perbandingan Berpasangan				
Harga	0.428571	0.380952	0.533333	0.250000
Kenamanan	0.214286	0.190476	0.133333	0.333333
Gaya	0.214286	0.380952	0.266667	0.333333
MPG	0.142857	0.047619	0.066667	0.083333

```

Price_weight = (criteria_weight['Harga'][0] + criteria_weight['Kenamanan'][0] + criteria_weight['Gaya'][0] + criteria_weight['MPG'][0])/4
Comfort_weight = (criteria_weight['Harga'][1] + criteria_weight['Kenamanan'][1] + criteria_weight['Gaya'][1] + criteria_weight['MPG'][1])/4
Style_weight = (criteria_weight['Harga'][2] + criteria_weight['Kenamanan'][2] + criteria_weight['Gaya'][2] + criteria_weight['MPG'][2])/4
MPG_weight = (criteria_weight['Harga'][3] + criteria_weight['Kenamanan'][3] + criteria_weight['Gaya'][3] + criteria_weight['MPG'][3])/4

```

Disini kita buat Price\_weight, Comfort\_weight, Style\_weight, MPG\_weight yang akan menampung rumus untuk mendapatkan bobot. Yang mana criteria\_weight akan di berikan variable global supaya bisa mengambil Harga yang ada pada list list sebelumnya dan diberikan array 0 untuk memanggil kolom 1 dan dijumlahkan dengan kenyamanan pada array 0 juga, dijumlahkan lagi dengan gaya serta MPG lalu terakhir dibagi dengan 4 karena total perbandingan ada 4. Begitu juga berikutnya.

```

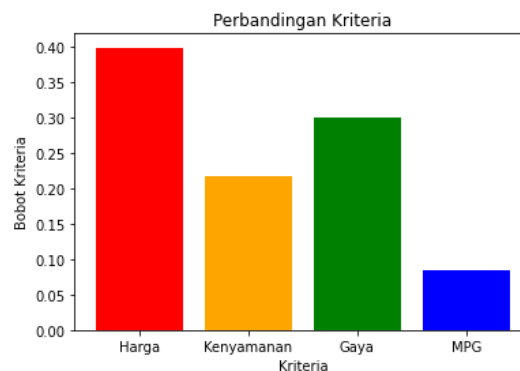
print(f'Bobot Harga : {Price_weight:.3f}\nBobot Kenamanan : {Comfort_weight:.3f}\nBobot Gaya : {Style_weight:.3f}\nBobot MPG : {MPG_weight:.3f}')

```

Lanjut kita cetak menggunakan print dan Bobot harga akan mencetak price\_weight dengan F mengacu pada "Format desimal titik mengambang". Angka .3 menunjukkan pembulatan ke 3 tempat setelah titik desimal. Begitu juga seterusnya.

```
plt.bar(['Harga','Kenamanan','Gaya','MPG'],[Price_weight, Comfort_weight, Style_weight, MPG_weight], color = ['red', 'orange', 'Green', 'blue'])
plt.title('Perbandingan Kriteria')
plt.xlabel('Kriteria')
plt.ylabel('Bobot Kriteria')
```

Kita visualisasikan data ke 2D berbentuk diagram bar dengan syntax plt adalah alias dari library matplotlib. Kita isi bar-nya dengan Harga, kenyamanan, gaya, MPG, lalu diisi dengan price\_weight, Comfort\_weight, Style\_weight, MPG\_weight diberikan juga warna red, orange, green, blue. Kita buat titlenya 'Perbandingan Kriteria', label x atau horizontal diisi dengan kriteria dan y vertikal diisi label Bobot Kriteria. Maka akan mencetak:



Hasil bobot di atas menunjukkan bahwa kriteria Harga merupakan kriteria yang paling penting dari semua kriteria, dengan bobot 0.398, disusul kriteria Gaya dengan bobot 0.299, disusul kriteria Kenyamanan dengan bobot 0.218, dan kriteria MPG dengan bobot 0.085.

**Tabel 3: Perbandingan Berpasangan yang menunjukkan Preferensi untuk proyek berdasarkan kriteria**

- Kriteria Harga**

```
PRICE_dic = {'HARGA': ['Proyek 1','Proyek 2','Proyek 3'],
              'Proyek 1': [1,3,4],
              'Proyek 2': [1/3,1,2],
              'Proyek 3': [1/4,1/2,1]}

PRICE_comparison = pd.DataFrame(PRICE_dic).set_index('HARGA')
PRICE_comparison
```

Deklarasikan PRICE\_dic yang akan memuat string dari HARGA dan isi dari column HARGA ada Proyek 1, Proyek 2, Proyek 3 . lalu pada Proyek 1, 2, 3 diisi dengan hasil perbandingan. Pada variable PRICE\_comparison akan diisi dengan DataFrame dari PRICE\_dic dan diawali dengan column HARGA. Hasil yang keluar sebagai berikut:

	Proyek 1	Proyek 2	Proyek 3
HARGA			
Proyek 1	1	0.333333	0.25
Proyek 2	3	1.000000	0.50
Proyek 3	4	2.000000	1.00

```
PRICE_criteria = PRICE_comparison.copy()
for col in ['Proyek 1', 'Proyek 2', 'Proyek 3']:
    PRICE_criteria[col] = PRICE_criteria[col] / PRICE_criteria[col].sum()
PRICE_criteria
```

Dilanjutlan kita copy untuk objek baru dari PRICE\_comparison dan kita berikan for pada column supaya perintah for yang digunakan bisa melakukan iterasi dari sebuah nilai sequence atau data koleksi pada python seperti List, Tuple, String dan lain-lain. Disini kita for pada Proyek 1, 2, dan 3 yang nantinya pada PRICE\_criteria akan dimasukkan column dan akan di Sum() atau dijumlahkan seluruh anggota list arraynya. Dan hasilnya sebagai berikut:

	Proyek 1	Proyek 2	Proyek 3
HARGA			
Proyek 1	0.125	0.1	0.142857
Proyek 2	0.375	0.3	0.285714
Proyek 3	0.500	0.6	0.571429

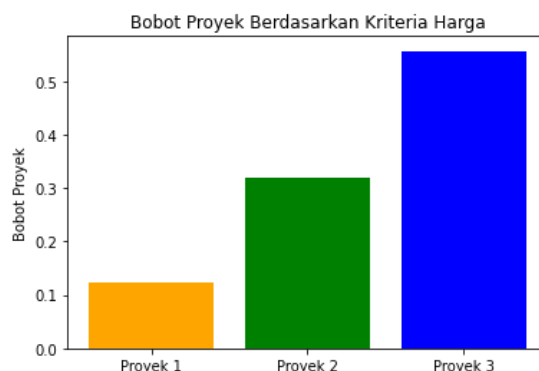
```
PRICE_P1 = (PRICE_criteria['Proyek 1'][0] + PRICE_criteria['Proyek 2'][0] + PRICE_criteria['Proyek 3'][0])/3
PRICE_P2 = (PRICE_criteria['Proyek 1'][1] + PRICE_criteria['Proyek 2'][1] + PRICE_criteria['Proyek 3'][1])/3
PRICE_P3 = (PRICE_criteria['Proyek 1'][2] + PRICE_criteria['Proyek 2'][2] + PRICE_criteria['Proyek 3'][2])/3
print(f'Bobot Proyek Berdasarkan Kriteria Harga\n-----\nProyek 1 : {PRICE_P1:.3f}\nProyek 2: {PRICE_P2:.3f}\nProyek 3 : {PRICE_P3:.3f}')
```

Kita buat PRICE\_P1, P2, dan P3 yang akan menampung rumus untuk mendapatkan bobot proyek berdasarkan kriteria harga. Yang mana PRICE\_criteria akan di berikan variable global supaya bisa mengambil Harga yang ada pada list list sebelumnya dan diberikan array 0 untuk memanggil kolom 1 dan dijumlahkan dengan Proyek 2 pada array 0 juga, dijumlahkan lagi dengan Proyek 3 serta dibagi dengan 3 karena total HARGA ada 3. Begitu juga berikutnya. Dan cetak menggunakan print, panggil PRICE\_P1, P2, dan P3 berikan .3F yang mana dengan F mengacu pada "Format desimal titik mengambang". Angka .3 menunjukkan pembulatan ke 3 tempat setelah titik desimal. Begitu juga seterusnya.

```
Bobot Proyek Berdasarkan Kriteria Harga
-----
Proyek 1 : 0.123
Proyek 2: 0.320
Proyek 3 : 0.557
```

```
plt.bar(['Proyek 1', 'Proyek 2', 'Proyek 3'], [PRICE_P1, PRICE_P2, PRICE_P3], color = ['orange', 'green', 'blue'])
plt.title('Bobot Proyek Berdasarkan Kriteria Harga')
plt.ylabel('Bobot Proyek')
```

Kita visualisasikan data ke 2D berbentuk diagram bar dengan syntax plt adalah alias dari library matplotlib. Kita isi barnya dengan Proyek 1, Proyek 2, Proyek 3, diisi dengan PRICE\_P1, PRICE\_P2, PRICE\_P3, berikan warna orange, green, dan blue lalu berikan title 'Bobot Proyek Berdasarkan Kriteria Harga', dan y label dengan naman Bobot Proyek. Maka akan mencetak:



Hasil di atas menunjukkan bahwa Proyek 3, pembelian model Ekonomi, memiliki bobot tertinggi 0.557. Ini berarti jika 3 proyek dianalisis berdasarkan kriteria Harga saja, Proyek 3 akan menjadi kandidat

terbaik untuk diseleksi.

- **Kriteria MPG**

```
MPG_dic = {'MPG': ['Proyek 1', 'Proyek 2', 'Proyek 3'],
            'Proyek 1': [1,4,6],
            'Proyek 2': [1/4,1,3],
            'Proyek 3': [1/6,1/3,1]}

MPG_comparison = pd.DataFrame(MPG_dic).set_index('MPG')

MPG_criteria = MPG_comparison.copy()
for col in ['Proyek 1', 'Proyek 2', 'Proyek 3']:
    MPG_criteria[col] = MPG_criteria[col] / MPG_criteria[col].sum()

MPG_P1 = (MPG_criteria['Proyek 1'] [0] + MPG_criteria['Proyek 2'] [0] + MPG_criteria['Proyek 3'] [0]) / 3
MPG_P2 = (MPG_criteria['Proyek 1'] [1] + MPG_criteria['Proyek 2'] [1] + MPG_criteria['Proyek 3'] [1]) / 3
MPG_P3 = (MPG_criteria['Proyek 1'] [2] + MPG_criteria['Proyek 2'] [2] + MPG_criteria['Proyek 3'] [2]) / 3

print(f'Bobot Proyek Berdasarkan Kriteria MPG\n-----\nProyek 1 : {MPG_P1:.3f}\nProyek 2 : {MPG_P2:.3f}\nProyek 3 : {MPG_P3:.3f}')
```

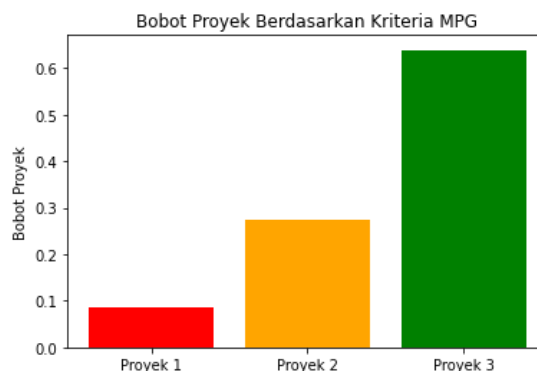
Deklarasikan MPG\_dic yang akan memuat string dari MPG dan isi dari column MPG ada Proyek 1, Proyek 2, Proyek 3. lalu pada Proyek 1, 2, 3 diisi dengan hasil perbandingan. Pada variable MPG\_comparison akan diisi dengan DataFrame dari MPG\_dic dan diawali dengan column MPG. Dilanjutlan kita copy untuk objek baru dari MPG\_comparison dan kita berikan for pada column supaya perintah for yang digunakan bisa melakukan iterasi dari sebuah nilai sequence atau data koleksi pada python seperti List, Tuple, String dan lain-lain. Disini kita for pada Proyek 1, 2, dan 3 yang nantinya pada MPG\_criteria akan dimasukkan column dan akan di Sum() atau dijumlahkan seluruh anggota list arraynya. Cetak menggunakan print, panggil MPG\_P1, P2, dan P3 berikan .3F yang mana dengan F mengacu pada "Format desimal titik mengambang". Angka .3 menunjukkan pembulatan ke 3 tempat setelah titik desimal. Begitu juga seterusnya.

Hasil yang keluar sebagai berikut:

```
Bobot Proyek Berdasarkan Kriteria MPG
-----
Proyek 1 : 0.087
Proyek 2 : 0.274
Proyek 3 : 0.639
```

```
plt.bar(['Proyek 1', 'Proyek 2', 'Proyek 3'], [MPG_P1, MPG_P2, MPG_P3], color = ['red', 'orange', 'green'])
plt.title('Bobot Proyek Berdasarkan Kriteria MPG')
plt.ylabel('Bobot Proyek')
```

Visualisasikan data ke 2D berbentuk diagram bar dengan menggunakan library matplotlib alias plt. Kita isi barnya dengan Proyek 1, Proyek 2, Proyek 3, diisi dengan MPG\_P1, MPG\_P2, MPG\_P3, berikan warna red, orange, green. Berikan title 'Bobot Proyek Berdasarkan Kriteria MPG, dan berikan y vertikal dengan label Bobot Proyek. Maka akan menghasilkan diagram data:



Hasil di atas menunjukkan bahwa Proyek 3, pembelian model Ekonomi, memiliki bobot tertinggi 0.639. Ini berarti jika 3 proyek dianalisis berdasarkan kriteria MPG saja, Proyek 3 akan menjadi kandidat terbaik untuk diseleksi.

- **Kriteria Kenyamanan**



```
COMFORT_dic = {'KENYAMANAN': ['Proyek 1','Proyek 2','Proyek 3'],
               'Proyek 1': [1,1/2,1/8],
               'Proyek 2': [2,1,1/6],
               'Proyek 3': [8,6,1,]}

COMFORT_comparison = pd.DataFrame(COMFORT_dic).set_index('KENYAMANAN')

COMFORT_criteria = COMFORT_comparison.copy()
for col in ['Proyek 1','Proyek 2','Proyek 3']:
    COMFORT_criteria[col] = COMFORT_criteria[col] / COMFORT_criteria[col].sum()

COMFORT_P1 = (COMFORT_criteria['Proyek 1'][0] + COMFORT_criteria['Proyek 2'][0] + COMFORT_criteria['Proyek 3'][0])/3
COMFORT_P2 = (COMFORT_criteria['Proyek 1'][1] + COMFORT_criteria['Proyek 2'][1] + COMFORT_criteria['Proyek 3'][1])/3
COMFORT_P3 = (COMFORT_criteria['Proyek 1'][2] + COMFORT_criteria['Proyek 2'][2] + COMFORT_criteria['Proyek 3'][2])/3

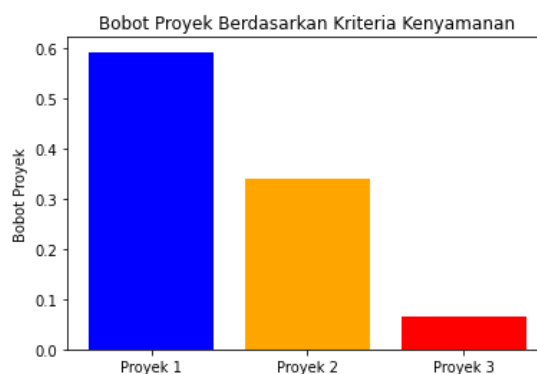
print(f'Bobot Proyek Berdasarkan Kriteria Kenyamanan\n-----\nProyek 1 : {COMFORT_P1:.3f}\nProyek 2 : {COMFORT_P2:.3f}\nProyek 3 : {COMFORT_P3:.3f}')
```

COMFORT\_dic dideklarasikan lalu, akan memuat string dari COMFORT yang berisi Proyek 1, Proyek 2, Proyek 3. Isikan pada semua Proyek dengan hasil perbandingan. Variable COMFORT\_comparison akan diisi dengan DataFrame dari COMFORT\_dic dan diawali dengan column COMFORT. Copy untuk objek baru dari COMFORT\_comparison menggunakan syntax copy() dan berikan for di column agar perintah for yang dipakai bisa melakukan iterasi dari sebuah nilai sequence atau data koleksi pada python seperti List, Tuple, String dan lain-lain. Nantinya pada COMFORT\_criteria akan dimasukkan column dan akan di jumlahkan dengan seluruh anggota list arraynya. Gunakan Print, panggil COMFORT\_P1, P2, dan P3 berikan .3F yang mana dengan F mengacu pada "Format desimal titik mengambang". Angka .3 menunjukkan pembulatan ke 3 tempat setelah titik desimal. Begitu juga seterusnya. Maka hasil outputnya:

```
Bobot Proyek Berdasarkan Kriteria Kenyamanan
-----
Proyek 1 : 0.593
Proyek 2 : 0.341
Proyek 3 : 0.065
```

```
plt.bar(['Proyek 1','Proyek 2','Proyek 3'],[COMFORT_P1,COMFORT_P2,COMFORT_P3], color = ['blue', 'orange', 'red'])
plt.title('Bobot Proyek Berdasarkan Kriteria Kenyamanan')
plt.ylabel('Bobot Proyek')
```

Data akan divisualisasikan ke bentuk diagram bar 2D dengan menggunakan library matplotlib. Isi barnya sebagai Proyek 1, Proyek 2, Proyek 3, diisi COMFORT\_P1, COMFORT\_P2, COMFORT\_P3, dengan warna blue, orange, red. Title dari diagramnya 'Bobot Proyek Berdasarkan Kriteria Kenyamanan, tidak lupa diberi label vertikal dengan nama Bobot Proyek. Maka akan tercetak diagram sebagai berikut:



Hasil di atas menunjukkan bahwa Proyek 1, pembelian model Eksekutif, memiliki bobot tertinggi sebesar 0.593. Ini berarti jika 3 proyek dianalisis berdasarkan kriteria Kenyamanan saja, Proyek 1 akan menjadi kandidat terbaik untuk diseleksi.

- **Kriteria Gaya**



```

STYLE_dic = {'GAYA': ['Proyek 1','Proyek 2','Proyek 3'],
              'Proyek 1': [1,3,1/4],
              'Proyek 2': [1/3,1,1/7],
              'Proyek 3': [4,7,1,]}

STYLE_comparison = pd.DataFrame(STYLE_dic).set_index('GAYA')

STYLE_criteria = STYLE_comparison.copy()
for col in ['Proyek 1','Proyek 2','Proyek 3']:
    STYLE_criteria[col] = STYLE_criteria[col] / STYLE_criteria[col].sum()

STYLE_P1 = (STYLE_criteria['Proyek 1'][0] + STYLE_criteria['Proyek 2'][0] + STYLE_criteria['Proyek 3'][0])/3
STYLE_P2 = (STYLE_criteria['Proyek 1'][1] + STYLE_criteria['Proyek 2'][1] + STYLE_criteria['Proyek 3'][1])/3
STYLE_P3 = (STYLE_criteria['Proyek 1'][2] + STYLE_criteria['Proyek 2'][2] + STYLE_criteria['Proyek 3'][2])/3

print(f'Bobot Proyek Berdasarkan Kriteria Gaya\n-----\nProyek 1 : {STYLE_P1:.3f}\nProyek 2 : {STYLE_P2:.3f}\nProyek 3 : {STYLE_P3:.3f}')

```

Buat deklarasi STYLE\_dic, kemudian akan memuat string dari COMFORT berisi Proyek 1, Proyek 2, Proyek 3 . Masukkan pada semua Proyek hasil perbandingannya. Variable STYLE\_comparison akan diisi dengan DataFrame dari STYLE\_dic dan diawali dengan column STYLE. Buat salinan objek baru dari STYLE\_comparison memakai syntax copy(). Selanjutnya For di column berfungsi supaya bisa melakukan iterasi dari sebuah nilai sequence atau data koleksi pada python seperti List, Tuple, String dan lain-lain. Nantinya pada STYLE\_criteria akan dimasukkan column yang nantinya di jumlahkan dengan seluruh anggota list arraynya. Print untuk mencetak dengan memanggil variable STYLE\_P1, P2, dan P3 berikan .3F yang mana dengan F mengacu pada "Format desimal titik mengambang". Angka .3 menunjukkan pembulatan ke 3 tempat setelah titik desimal. Begitu juga seterusnya. Hasil yang tercetak sebagai berikut:

```

Bobot Proyek Berdasarkan Kriteria Gaya
-----
Proyek 1 : 0.265
Proyek 2 : 0.656
Proyek 3 : 0.080

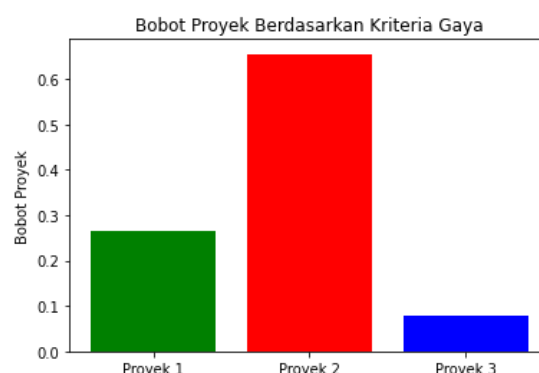
```

```

plt.bar(['Proyek 1','Proyek 2','Proyek 3'],[STYLE_P1,STYLE_P2,STYLE_P3], color = ['green', 'red', 'blue'])
plt.title('Bobot Proyek Berdasarkan Kriteria Gaya')
plt.ylabel('Bobot Proyek')

```

Agar tercetak data diagram gunakan syntax plt disertai dengan bar, plt sendir diambil dari library matplotlib. Pada bar masukkan Proyek 1, Proyek 2, Proyek 3, isi dengan STYLE\_P1, STYLE\_P2, STYLE\_P3, dengan warna blue, orange, red. Title dari diagramnya 'Bobot Proyek Berdasarkan Kriteria Gaya, beri label pada y vertikal dengan nama Bobot Proyek. Dat Diagram akan terlihat berikut:



Hasil di atas menunjukkan bahwa Proyek 2, pembelian model Medium, memiliki bobot tertinggi 0.656. Ini berarti jika 3 proyek dianalisis berdasarkan kriteria Gaya saja, Proyek 2 akan menjadi kandidat terbaik untuk diseleksi.

## - Keempat Kriteria

```

Project_1 = Price_weight*PRICE_P1 + Comfort_weight*COMFORT_P1 + MPG_weight*MPG_P1 + Style_weight*STYLE_P1
Project_2 = Price_weight*PRICE_P2 + Comfort_weight*COMFORT_P2 + MPG_weight*MPG_P2 + Style_weight*STYLE_P2
Project_3 = Price_weight*PRICE_P3 + Comfort_weight*COMFORT_P3 + MPG_weight*MPG_P3 + Style_weight*STYLE_P3

print(f'Bobot Proyek Berdasarkan 4 Kriteria\n-----\nProyek 1 : {Project_1:.3f}\nProyek 2 : {Project_2:.3f}\nProyek 3 : {Project_3:.3f}')

```

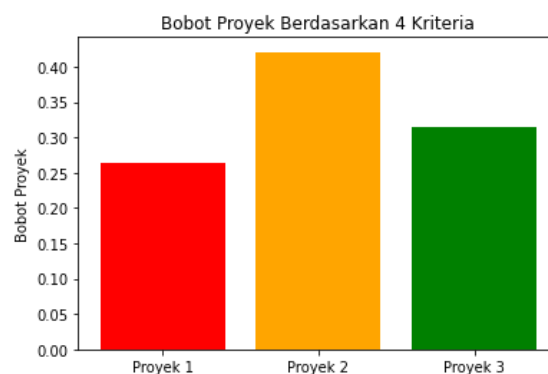
Dari Project 1, Project 2, Project 3 kita berikan value Price\_weight (diambil dari Perbandingan Berpasangan 4) yang akan dikalikan dengan PRICE\_P1 (mengambil data dari Kriteria Harga) lalu

dijumlahkan dengan Comfort\_weight (diambil dari Perbandingan Berpasangan 4) dikalikan dengan COMFORT\_P1 (diambil data dari Kriteria Kenyamanan) ditambahkan dengan MPG\_weight (diambil dari Perbandingan Berpasangan 4) dikalikan dengan MPG\_P1 (mengambil data dari Kriteria MPG) dan terakhir ditambahkan dengan Style\_weight (diambil dari Perbandingan Berpasangan 4) dikalikan dengan STYLE\_P1 (mengambil data dari Kriteria Gaya). Kita Print mencetak hasil, tinggal panggil variable STYLE\_P1, P2, dan P3 berikan .3F yang mana dengan F mengacu pada "Format desimal titik mengambang". Angka .3 menunjukkan pembulatan ke 3 tempat setelah titik desimal. Hasil yang tercetak:

```
Bobot Proyek Berdasarkan 4 Kriteria
-----
Proyek 1 : 0.265
Proyek 2 : 0.421
Proyek 3 : 0.314
```

```
plt.bar(['Proyek 1','Proyek 2','Proyek 3'],[Project_1,Project_2,Project_3], color = ['red', 'orange', 'green'])
plt.title('Bobot Proyek Berdasarkan 4 Kriteria')
plt.ylabel('Bobot Proyek')
```

Membuat data diagram dengan menggunakan library matplotlib yang dialiaskan menjadi plt supaya ringkas. Pada bar masukkan Proyek 1, Proyek 2, dan Proyek 3. Kemudian pada setiap Proyek diisi dengan STYLE\_P1, STYLE\_P2, STYLE\_P3, berikan warna pada setiap Proyek supaya mudah membedakan dengan blue, orange, red. Berikan Title dari data diagramnya 'Bobot Proyek Berdasarkan Kriteria Gaya, terakhir berikan nama Bobot Proyek pada label dengan posisi vertikal (y). Alhasil Data Diagram akan nampak sebagai berikut:



Hasil akhir di atas menunjukkan bahwa proyek terbaik berdasarkan kombinasi keempat kriteria adalah Proyek 2, pembelian model Medium.

## Hasil Menggunakan Compiler

```
Dell Latitude7480@DESKTOP-7MGRFRG MINGW64 /d/N/PENS/Semester 3/Sistem Pengambilan Keputusan/Proyek Akhir
$ py ahp_kelompok_8.py
Bobot Harga : 0.398
Bobot Kenyamanan : 0.218
Bobot Gaya : 0.299
Bobot MPG : 0.085
Bobot Proyek Berdasarkan Kriteria Harga
-----
Proyek 1 : 0.123
Proyek 2: 0.320
Proyek 3 : 0.557
Bobot Proyek Berdasarkan Kriteria MPG
-----
Proyek 1 : 0.087
Proyek 2 : 0.274
Proyek 3 : 0.639
Bobot Proyek Berdasarkan Kriteria Kenyamanan
-----
Proyek 1 : 0.593
Proyek 2 : 0.341
Proyek 3 : 0.065
Bobot Proyek Berdasarkan Kriteria Gaya
-----
Proyek 1 : 0.265
Proyek 2 : 0.656
Proyek 3 : 0.080
Bobot Proyek Berdasarkan 4 Kriteria
-----
Proyek 1 : 0.265
Proyek 2 : 0.421
Proyek 3 : 0.314
```