

## Laporan Observasi – 3

Aditya Januar Widiyanto | 1301170039 | IF-41-12

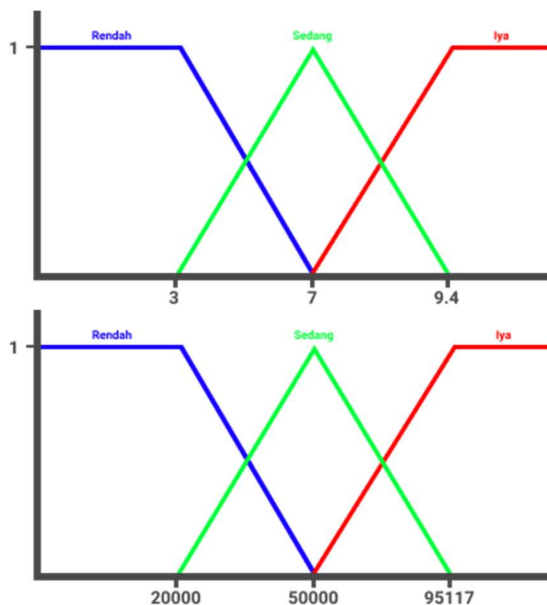
### Pendahuluan

Logika *Fuzzy* adalah suatu bentuk logika yang memiliki banyak nilai, di mana nilai kebenarannya berupa dua angka *real* yaitu 1 dan 0. Ini digunakan untuk menangani konsep kebenaran parsial, di mana nilai kebenaran dapat berkisar antara sepenuhnya benar dan sepenuhnya salah.

Istilah logika *fuzzy* diperkenalkan dengan proposal teori himpunan *fuzzy* 1965 oleh *Lotfi Zadeh*. Namun logika *fuzzy* telah dipelajari sejak tahun 1920-an, sebagai logika dengan nilai tak terbatas — terutama oleh *Lukasiewicz* dan *Tarski*.

### Isi

Pertama-tama kita membuat grafik terlebih dahulu untuk *follower* dan *engagement*:



Saya memilih batasannya ada tiga, yaitu rendah, sedang dan tinggi untuk masing-masing *follower* dan *engagement*. Untuk *follower* sendiri saya memilih angka 20000, 50000 dan 95117 karena dari angka-angka tersebut memenuhi kriteria untuk dibagi menjadi tiga area yang dipilih tadi. Sama

untuk pemilihan angka untuk *engagement* sendiri yaitu 3, 7, dan 9.4 karena angka-angka tersebut memenuhi kriteria untuk membagi daerah grafik menjadi tiga bagian. Dari grafik tersebut kita mencari nilai-nilainya dengan rumusan:

$$\mu(x) = \frac{-(x - c)}{(c - b)} = \frac{(c - x)}{(c - b)}$$

$$\mu(x) = \frac{(x - a)}{(b - a)}$$

Setelah itu saya melakukan perumusan *rule*-nya yang saya pilih:

<i>follower</i>   <i>engagement</i>	rendah	sedang	tinggi
rendah	tidak	tidak	tidak
sedang	iya	tidak	tidak
tinggi	iya	iya	tidak

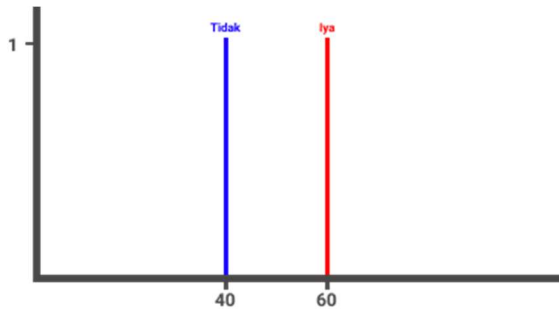
```
def inferensi(fr, fs, ft, er, es, et, pengikut, engage):  
    iya = [0, 0, 0]  
    tidak = [0, 0, 0, 0, 0, 0]  
    if (fr > 0) and (er > 0):  
        tidak[0] = min(fr, er)  
    if (fr > 0) and (es > 0):  
        iya[0] = min(fr, es)  
    if (fr > 0) and (et > 0):  
        iya[1] = min(fr, et)  
    if (fs > 0) and (er > 0):  
        tidak[1] = min(fs, er)  
    if (fs > 0) and (es > 0):  
        tidak[2] = min(fs, es)  
    if (fs > 0) and (et > 0):  
        iya[2] = min(fs, et)  
    if (ft > 0) and (er > 0):  
        tidak[3] = min(ft, er)  
    if (ft > 0) and (es > 0):  
        tidak[4] = min(ft, es)  
    if (ft > 0) and (et > 0):  
        tidak[5] = min (ft, et)  
  
    return max(iya), max(tidak)
```

Untuk tahap selanjutnya ialah pemilihan penggunaan metode *mamdani* atau metode *sugeno* untuk *defuzzifikasi*. Di sini saya memilih metode *sugeno* karena lebih mudah untuk diimplementasikan dan

dipahami. Dengan rumusan sebagai berikut:

$$y^* = \frac{(\text{nilai max } a \times \text{titik } a) + (\text{nilai max } b \times \text{titik } b)}{\text{nilai max } a + \text{nilai max } b}$$

Dari pengimplementasian di atas, didapatkan grafiknya yaitu:



Dari grafik di atas, saya memilih angka 40 dan 60 karena angka tersebut merupakan dua buah angka yang mendekati setengah dari 100 tapi tidak benar-benar di tengahnya.

Maka dari itu hasil keluaran atau *output* dari program ini adalah:

```
PS C:\Users\Aditya J. Widiyanto> & "C:/Users/Aditya J. Widiyanto/Tupro 3/IF4112_1301170039.py"
orang yang terpilih, orang ke- 2
Nilai Y* : 47.84
orang yang terpilih, orang ke- 5
Nilai Y* : 42.5
orang yang terpilih, orang ke- 6
Nilai Y* : 41.6
orang yang terpilih, orang ke- 8
Nilai Y* : 42.0
orang yang terpilih, orang ke- 11
Nilai Y* : 40.0
orang yang terpilih, orang ke- 27
Nilai Y* : 40.5
orang yang terpilih, orang ke- 28
Nilai Y* : 43.5
orang yang terpilih, orang ke- 34
Nilai Y* : 44.5
orang yang terpilih, orang ke- 36
Nilai Y* : 42.06
orang yang terpilih, orang ke- 39
Nilai Y* : 60.0
orang yang terpilih, orang ke- 48
Nilai Y* : 57.5
orang yang terpilih, orang ke- 51
Nilai Y* : 40.51
orang yang terpilih, orang ke- 67
Nilai Y* : 42.81
orang yang terpilih, orang ke- 74
Nilai Y* : 56.5
orang yang terpilih, orang ke- 76
Nilai Y* : 42.1
```

Di program utama, semua fungsi-fungsi tadi di panggil dan di proses untuk menghasilkan

perhitungan matematis yang menghasilkan nilai-nilai yang terbaik dari yang terbaik.

## Penutup

Kesimpulan dari program ini ialah mencari dua puluh orang atau *ID* untuk diidentifikasi dan memilih yang terbaik untuk di keluarkan dan di masukkan ke fail .csv dari 100 data yang tersedia.