

**LAPORAN TUGAS PEMROGRAMAN
PENGANTAR KECERDASAN BUATAN
FUZZY LOGIC**



Disusun Oleh :

Kelompok 1

Anyelir Belia Azzahra (1301200048)

Risma Amaliyah Mahmudah (1301204087)

Muhammad Rafi Irfansyah (1301204500)

Program Studi Informasi

Fakultas Informatika

Telkom University

2021/2022

BAB 1

PENDAHULUAN DAN HASIL OBSERVASI

Logika Fuzzy adalah suatu logika yang memiliki nilai kekaburan atau kesamaran antara benar dan salah. Dalam teori logika fuzzy, suatu nilai bisa bernilai benar dan salah secara bersamaan. Namun berapa besar kemungkinan benar dan salahnya suatu nilai tergantung pada bobot keanggotaan yang dimilikinya.

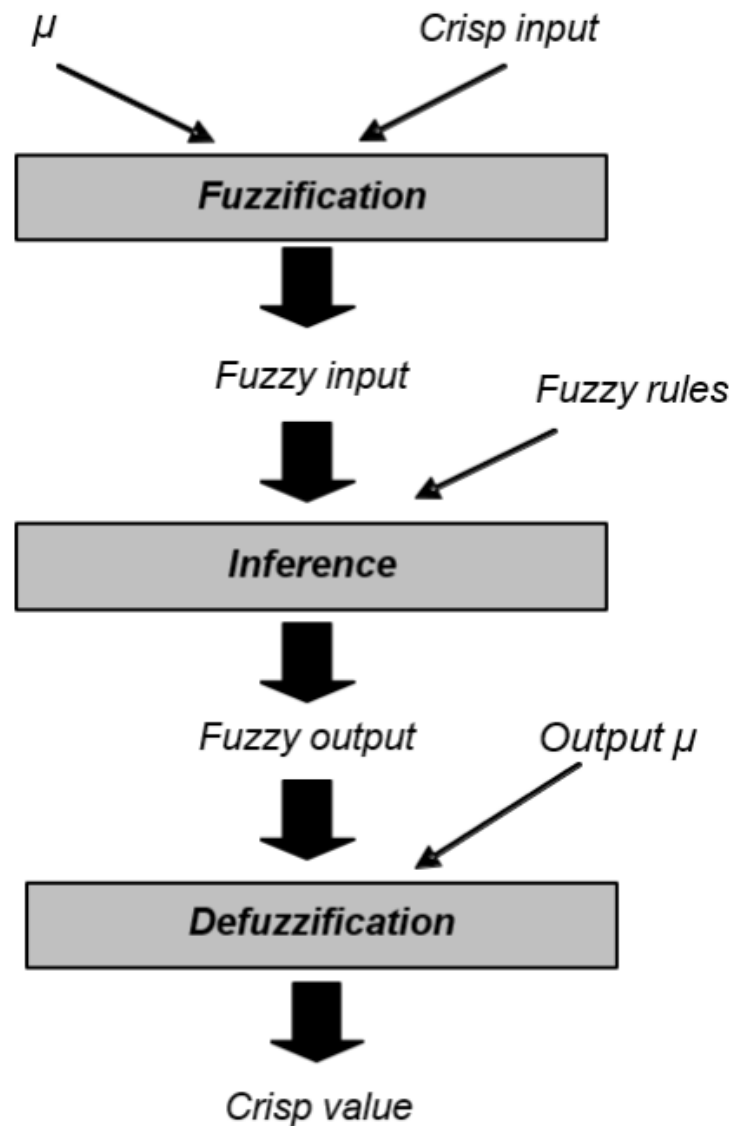
Logika fuzzy merupakan logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”, dan lain lain. Oleh karena itu, semua ini dapat mempunyai nilai keanggotaan 0 atau 1. Akan tetapi ada di antara 0 dan 1. Artinya, bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya, Hasan & Widiastuti (dalam Syahnandar, 2017).

Terdapat tiga proses utama pada sistem Fuzzy, yaitu: Fuzzification, Inference, dan Defuzzification. Pada proses Fuzzification memproses crisp input menjadi nilai fuzzy sesuai fungsi keanggotaan yang telah dibangun, pada proses inference menentukan fuzzy output sesuai aturan yang telah ditentukan, dan pada proses Defuzzification mengubah kembali nilai fuzzy output menjadi crisp value.

Metode output atau metode defuzzifikasi yang digunakan pada sistem fuzzy ada dua, yaitu: Center of Gravity (Mamdani Style) dan Constant Defuzzification (Takagi-sugeno-Style). Fuzzy metode Takagi-Sugeno merupakan metode inferensi fuzzy untuk aturan yang direpresentasikan dalam bentuk IF – THEN, Lamrabet, Ech-charqy, Tissir, & Haoussi (dalam Mukaromah, 2019) dimana output (konsekuen) sistem tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linear, Civelek (dalam Mukaromah, 2019).

Tugas pemrograman kedua kami kali ini diberikan file bengkel.xls berupa himpunan data 100 bengkel mobil yang ada di kota Bandung dengan dua atribut: Kualitas servis (bilangan real 1-100; semakin tinggi semakin baik) dan harga (bilangan real 1-10; semakin tinggi semakin mahal). Pada tugas kali ini kami akan membangun sebuah sistem berbasis fuzzy logic untuk memilih 10 bengkel terbaik di kota Bandung. Sistem membaca masukan file bengkel.xlsx dan mengeluarkan output berupa sebuah file peringkat.xlsx yang berisi 10 nomor/ID bengkel terbaik beserta skor-nya (Output Defuzzification).

Adapun hal yang diobservasi dalam penelitian ini yaitu proses Fuzzifikasi, Inferensi, dan Defuzzifikasi. Hal tersebut dapat digambarkan pada diagram berikut:



Dimana,

- Fuzzification : Merubah nilai crisp value (1 atau 0) menjadi fuzzy value.
- Inference : Menentukan fuzzy output menggunakan fuzzy rules.
- Defuzzification : Merubah fuzzy value menjadi crisp value kembali.

BAB 2 IMPLEMENTASI

1. Fuzzifikasi

Pada proses fuzzifikasi, kami mendefinisikan jumlah dan nama linguistik setiap input beserta bentuk dan batas fungsi keanggotaan input untuk mendapatkan crisp value dari masing-masing data yang ada.

1.1. Jumlah dan Nama Linguistik Setiap Input

Pada sistem rekomendasi bengkel, kami mendefinisikan jumlah dan nama linguistik yaitu Harga dan Servis. Kami menggunakan 2 input serta 4 variabel linguistik pada harga dan servis.

1.1.1 Harga

```
def fuzzy_harga(x):  
    harga = {'Sangat Murah': 0, 'Murah': 0, 'Mahal': 0, 'Sangat Mahal': 0}
```

Dalam variabel Harga kami menggunakan 4 nilai linguistik yaitu: Sangat Murah, Murah, Mahal, Sangat Mahal.

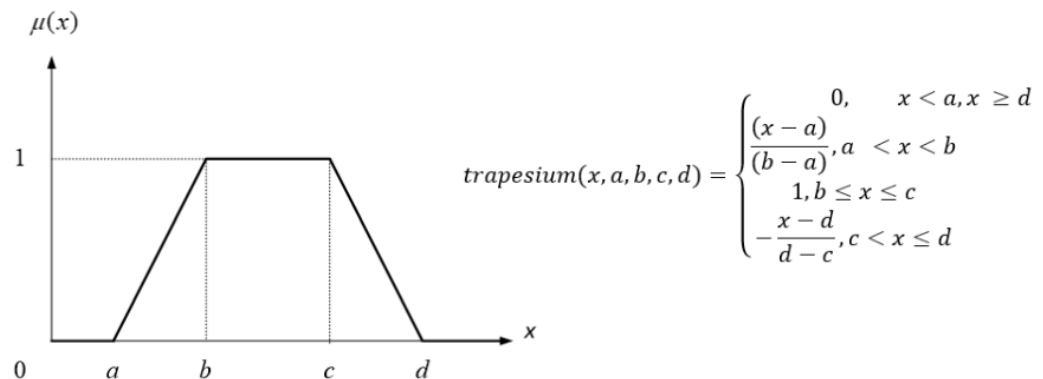
1.1.2 Servis

```
def fuzzy_servis(x):  
    servis = {'Sangat Kurang': 0, 'Kurang': 0, 'Baik': 0, 'Sangat Baik': 0}
```

Dalam variabel Servis kami menggunakan 4 nilai linguistik yaitu: Sangat Kurang, Kurang, Baik, Sangat Baik.

1.2. Bentuk dan Batas Fungsi Keanggotaan Input

Pada sistem bengkel, kami menggunakan fungsi trapesium untuk variabel Harga dan servis. Fungsi trapesium karena memiliki bentuk datar di bagian tengah sehingga memiliki rentang yang bisa dianggap sama untuk setiap nilai yang ada. Penerapan fungsi trapesium dapat digunakan dalam rumus berikut ini :



1.3. Batas Fungsi Keanggotaan Input

Fungsi keanggotaan harga:

- Nilai harga ≤ 4 bernilai Sangat Murah
- Nilai $5 \leq \text{harga} \leq 7$ bernilai Murah
- Nilai $8 \leq \text{harga} \leq 9$ bernilai Mahal
- Nilai harga > 9 bernilai Sangat Mahal

Fungsi keanggotaan servis:

- Nilai servis ≤ 40 bernilai Sangat Kurang
- Nilai $50 \leq \text{servis} \leq 70$ bernilai Kurang dengan nilai dari 41 - 79
- Nilai $80 \leq \text{servis} \leq 90$ bernilai Baik dengan nilai dari 71 - 94
- Nilai servis ≥ 95 bernilai Sangat Baik

1.4. Fungsi Keanggotaan Harga

Fungsi keanggotaan harga digunakan untuk mengelompokkan harga dengan batas batas yang telah ditentukan.

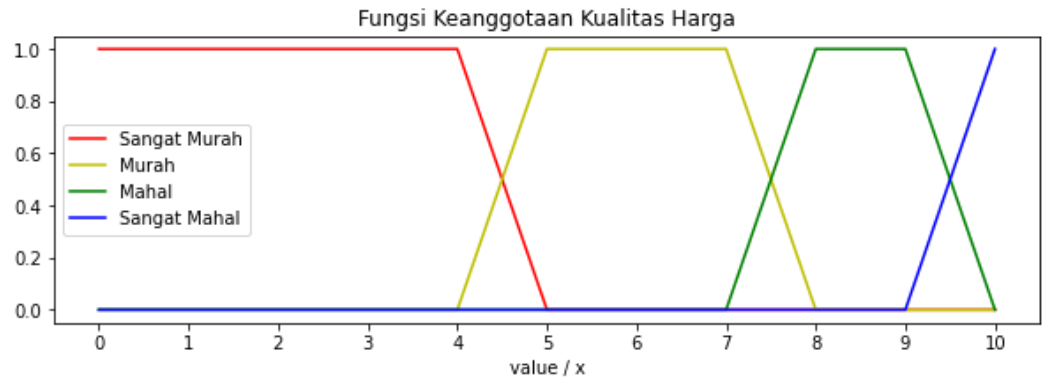
```
[ ] def fuzzy_harga(x):
    harga = {'Sangat Murah': 0, 'Murah': 0, 'Mahal': 0, 'Sangat Mahal': 0}

    a, b, c, d, e, f = 4, 5, 7, 8, 9, 10

    # Garis Datar
    if x <= a:
        harga['Sangat Murah'] = 1
    if b <= x <= c:
        harga['Murah'] = 1
    if d <= x <= e:
        harga['Mahal'] = 1
    if x >= f:
        harga['Sangat Mahal'] = 1

    # Garis Miring (Gradient)
    if a < x <= b:
        harga['Sangat Murah'] = -(x - b) / (b - a)
    if a < x < b:
        harga['Murah'] = (x - a) / (b - a)
    if c < x <= d:
        harga['Murah'] = -(x - d) / (d - c)
    if c < x < d:
        harga['Mahal'] = (x - c) / (d - c)
    if e < x < f:
        harga['Mahal'] = -(x - f) / (f - e)
    if e < x <= f:
        harga['Sangat Mahal'] = (x - e) / (f - e)

    return harga
```



Gambar di atas merupakan bentuk implementasi dan visualisasi dari fungsi keanggotaan input untuk variabel harga yang telah dibuat. Terdapat 4 nilai linguistik yaitu: Sangat Murah untuk garis berwarna merah, Murah untuk garis berwarna kuning, Mahal untuk garis berwarna hijau, Sangat Mahal untuk garis berwarna biru. Hasil visualisasi di atas mengikuti pola rumus trapesium yang telah dijelaskan sebelumnya.

1.5. Fungsi Keanggotaan Servis

Fungsi keanggotaan servis merupakan fungsi yang digunakan untuk mengelompokkan tingkat servis suatu bengkel dengan batas batas yang telah ditentukan.

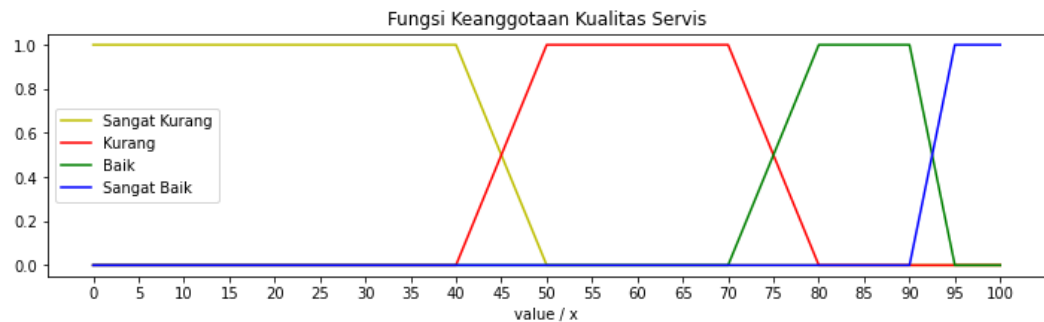
```
[ ] def fuzzy_servis(x):
    servis = {'Sangat Kurang': 0, 'Kurang': 0, 'Baik': 0, 'Sangat Baik': 0}

    a, b, c, d, e, f = 40, 50, 70, 80, 90, 95

    # Garis Datar
    if x <= a:
        servis['Sangat Kurang'] = 1
    if b <= x <= c:
        servis['Kurang'] = 1
    if d <= x <= e:
        servis['Baik'] = 1
    if x >= f:
        servis['Sangat Baik'] = 1

    # Garis Miring (Gradient)
    if a < x <= b:
        servis['Sangat Kurang'] = -(x - b) / (b - a)
    if a < x < b:
        servis['Kurang'] = (x - a) / (b - a)
    if c < x <= d:
        servis['Kurang'] = -(x - d) / (d - c)
    if c < x < d:
        servis['Baik'] = (x - c) / (d - c)
    if e < x < f:
        servis['Baik'] = -(x - f) / (f - e)
    if e < x <= f:
        servis['Sangat Baik'] = (x - e) / (f - e)

    return servis
```



Gambar di atas merupakan bentuk implementasi dan visualisasi dari fungsi keanggotaan input untuk variabel servis yang telah dibuat. Terdapat 4 nilai linguistik yaitu: Sangat Kurang untuk garis berwarna merah, Kurang untuk garis berwarna kuning, Baik untuk garis berwarna hijau, Sangat Baik untuk garis berwarna biru. Hasil visualisasi di atas mengikuti pola rumus trapesium yang telah dijelaskan sebelumnya.

2. Aturan Inferensi

```
# Untuk setiap rules merupakan Konjungsi.
# Contoh: IF Harga = Tidak Enak (1) ^ Pelayanan = Sangat Kurang (1) THEN Status = Tidak Direkomendasikan (1)

fuzzy_set_rules = {
    ('Sangat Murah', 'Sangat Kurang') : 'Tidak Direkomendasikan',
    ('Sangat Murah', 'Kurang') : 'Tidak Direkomendasikan',
    ('Sangat Murah', 'Baik') : 'Tidak Direkomendasikan',
    ('Sangat Murah', 'Sangat Baik') : 'Tidak Direkomendasikan',
    ('Murah', 'Sangat Kurang') : 'Tidak Direkomendasikan',
    ('Murah', 'Kurang') : 'Tidak Direkomendasikan',
    ('Murah', 'Baik') : 'Direkomendasikan',
    ('Murah', 'Sangat Baik') : 'Direkomendasikan',
    ('Mahal', 'Sangat Kurang') : 'Tidak Direkomendasikan',
    ('Mahal', 'Kurang') : 'Direkomendasikan',
    ('Mahal', 'Baik') : 'Direkomendasikan',
    ('Mahal', 'Sangat Baik') : 'Sangat Direkomendasikan',
    ('Sangat Mahal', 'Sangat Kurang') : 'Tidak Direkomendasikan',
    ('Sangat Mahal', 'Kurang') : 'Direkomendasikan',
    ('Sangat Mahal', 'Baik') : 'Sangat Direkomendasikan',
    ('Sangat Mahal', 'Sangat Baik') : 'Sangat Direkomendasikan',
}
```

Kami menggunakan 16 aturan inferensi karena setiap variabel dalam harga dan jasa memiliki 4 variabel linguistik, sehingga akan menghasilkan 16 aturan inferensi. Aturan inferensi kami didasarkan pada peringkat harga, yaitu, Sangat Murah, Murah, Mahal, dan Sangat Mahal, dan peringkat layanan, yaitu, Sangat Kurang, Kurang, Baik, dan Sangat Baik, dikelompokkan ke dalam bentuk keluaran fungsi keanggotaan seperti Tidak Direkomendasikan, Direkomendasikan, dan Rekomendasi kuat. Saat kami menggunakan logika gabung berikut untuk melakukan proses inferensi:

$$T(P \wedge Q) = \min\{T(P), T(Q)\}$$

Berikut merupakan konjungsi dari output fungsi keanggotaan yang dibuat :

Servis Sangat Kurang

- IF Harga = Sangat Mahal \wedge Servis = Sangat Kurang THEN Status = Tidak Direkomendasikan
- IF Harga = Mahal \wedge Servis = Sangat Kurang THEN Status = Tidak Direkomendasikan
- IF Harga = Murah \wedge Servis = Sangat Kurang THEN Status = Tidak Direkomendasikan
- IF Harga = Sangat Murah \wedge Servis = Sangat Kurang THEN Status = Tidak Direkomendasikan

Servis Kurang

- IF Harga = Sangat Mahal \wedge Servis = Kurang THEN Status = Tidak Direkomendasikan
- IF Harga = Mahal \wedge Servis = Kurang THEN Status = Tidak Direkomendasikan
- IF Harga = Murah \wedge Servis = Kurang THEN Status = Direkomendasikan
- IF Harga = Sangat Murah \wedge Servis = Kurang THEN Status = Direkomendasikan

Servis Baik

- IF Harga = Sangat Mahal \wedge Servis = Baik THEN Status = Tidak Direkomendasikan
- IF Harga = Mahal \wedge Servis = Baik THEN Status = Direkomendasikan
- IF Harga = Murah \wedge Servis = Baik THEN Status = Direkomendasikan
- IF Harga = Sangat Murah \wedge Servis = Baik THEN Status = Direkomendasikan

Servis Sangat Baik

- IF Harga = Sangat Mahal \wedge Servis = Sangat Baik THEN Status = Tidak Direkomendasikan
- IF Harga = Mahal \wedge Servis = Sangat Baik THEN Status = Direkomendasikan
- IF Harga = Murah \wedge Servis = Sangat Baik THEN Status = Sangat Direkomendasikan
- IF Harga = Sangat Murah \wedge Servis = Sangat Baik THEN Status = Sangat Direkomendasikan

Fungsi keanggotaan dapat juga dilihat dari tabel berikut ini :

		Servis			
		Sangat kurang	Kurang	Baik	Sangat Baik
Harga	Sangat Mahal	Tidak Direkomendasikan	Tidak Direkomendasikan	Tidak Direkomendasikan	Tidak Direkomendasikan
	Mahal	Tidak Direkomendasikan	Tidak Direkomendasikan	Direkomendasikan	Direkomendasikan
	Murah	Tidak Direkomendasikan	Direkomendasikan	Direkomendasikan	Sangat Direkomendasikan
	Sangat Murah	Tidak Direkomendasikan	Direkomendasikan	Sangat Direkomendasikan	Sangat Direkomendasikan

3. Proses Inferensi

```
inference_result = []

def inference(fuzzed):
    result = {'Tidak Direkomendasikan': 0, "Direkomendasikan": 0, "Sangat Direkomendasikan": 0}

    for Servis in fuzzed['Servis'].keys():
        for Harga in fuzzed['Harga'].keys():
            min_value = min(fuzzed['Harga'][Harga], fuzzed['Servis'][Servis])

            # Proses Conjunction Sesuai Fuzzy Rules pada variabel fuzzy_set_rules
            output = fuzzy_set_rules[(Harga, Servis)]

            if min_value > result[output]:
                result[output] = min_value

    return result
```

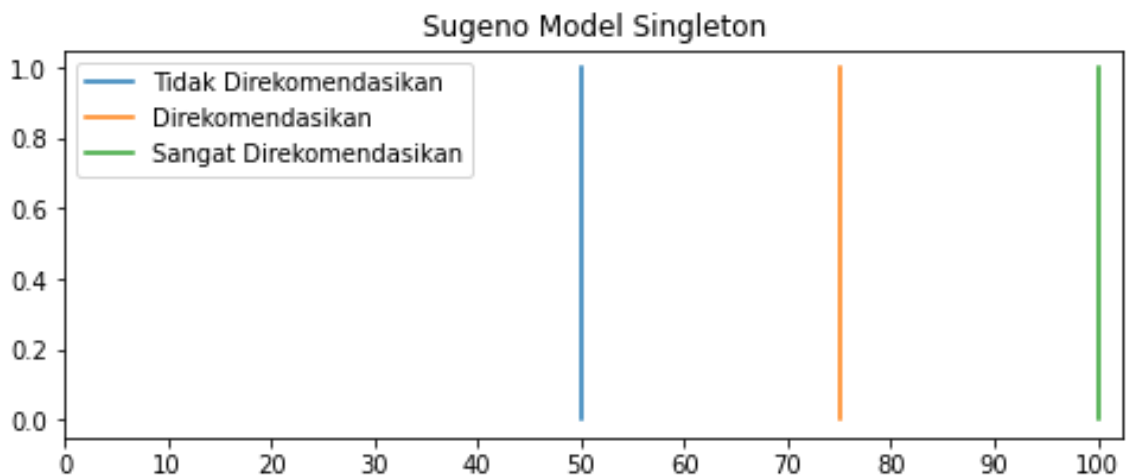
Proses inferensi dilakukan sesuai dengan aturan inferensi yang telah dirumuskan sebelumnya, dan setiap nilai harga dan nilai service dibuat dari data evaluasi bengkel yang telah dilakukan proses fuzzifikasi. Selama inferensi ini, salah satu keluaran fungsi keanggotaan diperoleh berupa status yang akan diperoleh bengkel, yaitu Tidak Direkomendasikan, Direkomendasikan, atau Sangat Direkomendasikan. Proses yang dilakukan pada saat proses inferensi adalah mengambil nilai minimum untuk setiap nilai pada hasil proses fuzzifikasi dengan cara membandingkan hasil crisp value fuzzifikasi dengan set rules yang ada. Berikut ini adalah contoh output dari program yang dibuat:

Contoh Hasil Inferensi :

```
{'Tidak Direkomendasikan': 1, 'Direkomendasikan': 0, 'Sangat Direkomendasikan': 0}
{'Tidak Direkomendasikan': 1, 'Direkomendasikan': 0, 'Sangat Direkomendasikan': 0}
{'Tidak Direkomendasikan': 1, 'Direkomendasikan': 0, 'Sangat Direkomendasikan': 0}
{'Tidak Direkomendasikan': 1, 'Direkomendasikan': 0, 'Sangat Direkomendasikan': 0}
{'Tidak Direkomendasikan': 1, 'Direkomendasikan': 0, 'Sangat Direkomendasikan': 0}
{'Tidak Direkomendasikan': 0, 'Direkomendasikan': 1.0, 'Sangat Direkomendasikan': 0}
{'Tidak Direkomendasikan': 0, 'Direkomendasikan': 1, 'Sangat Direkomendasikan': 0}
{'Tidak Direkomendasikan': 1.0, 'Direkomendasikan': 0, 'Sangat Direkomendasikan': 0}
{'Tidak Direkomendasikan': 0.5, 'Direkomendasikan': 0, 'Sangat Direkomendasikan': 0}
{'Tidak Direkomendasikan': 1, 'Direkomendasikan': 0, 'Sangat Direkomendasikan': 0}
```

4. Defuzzifikasi

Pada proses ini kami menggunakan logika fuzzy metode sugeno karena sistem fuzzy sugeno dapat memperbaiki kelemahan yang dimiliki oleh sistem fuzzy murni untuk menambah suatu perhitungan matematika sederhana sebagai bagian IF THEN. Defuzzifikasi sendiri merupakan langkah terakhir dalam suatu sistem logika fuzzy yang bertujuan mengkonversi setiap hasil dari inference engine yang diekspresikan dalam bentuk fuzzy set ke suatu bilangan real.



Di sini kami menggunakan metode sugeno model singleton agar mendapatkan hasil nilai konstan sehingga crisp output pada penelitian ini merupakan hasil perkalian

dari fungsi keanggotaan output nilai konstan singleton yang telah ditentukan yaitu: 50, 75, dan 100 sesuai pada grafik yang ada di atas.

```
deffuzy = {'Tidak Direkomendasikan': 50, "Direkomendasikan": 75, "Sangat Direkomendasikan": 100}

def defuzzification(inference, deffuzy):
    numerator, denominator = 0, 0

    for output in deffuzy.keys():
        numerator += inference[output] * deffuzy[output]
        denominator += inference[output]

    return numerator/denominator

final = []

for inference in inference_result:
    final.append(defuzzification(inference, deffuzy))
```

Pada metode sugeno ini digunakan untuk menentukan nilai rekomendasi suatu bengkel berdasarkan penilaian harga dan servis bengkel tersebut. Sehingga persamaan yang digunakan dalam metode defuzzifikasi tersebut adalah sebagai berikut:

$$z^* = \frac{\sum_{i=1}^l \mu B_i C_i}{\sum_{i=1}^l \mu B_i} \quad \begin{array}{l} c_i = \text{constant for } i^{\text{th}} \text{ linguistic} \\ \mu B_i = \text{membership for } i^{\text{th}} \text{ linguistic} \end{array}$$

Berikut merupakan contoh hasil metode defuzzifikasi sugeno yang telah kami buat:

1. Tidak Direkomendasikan: 1, Direkomendasikan: 0, Sangat Direkomendasikan: 0

$$z^* = \frac{1 * 50 + 0 * 75 + 0 * 100}{1 + 0 + 0} = 50$$

2. Tidak Direkomendasikan: 0, Direkomendasikan: 1, Sangat Direkomendasikan: 0

$$z^* = \frac{0 * 50 + 1 * 75 + 0 * 100}{0 + 1 + 0} = 75$$

3. Tidak Direkomendasikan: 0, Direkomendasikan: 0, Sangat Direkomendasikan: 1

$$z^* = \frac{0 * 50 + 0 * 75 + 1 * 100}{0 + 0 + 1} = 100$$

Adapun hasil dari fuzzifikasi menghasilkan sebuah crisp output yang dapat menjadi acuan untuk menentukan peringkat bengkel terbaik. Berikut merupakan contoh hasil crisp output pada program :

```
print("Contoh Hasil Defuzifikasi:\n")
for i in range(0, 10):
    print(f'Data Ke-{i+1} = {final[i]}')
```

Contoh Hasil Defuzifikasi:

```
Data Ke-1 = 50.0
Data Ke-2 = 50.0
Data Ke-3 = 50.0
Data Ke-4 = 50.0
Data Ke-5 = 50.0
Data Ke-6 = 75.0
Data Ke-7 = 75.0
Data Ke-8 = 50.0
Data Ke-9 = 50.0
Data Ke-10 = 50.0
```

BAB 3

KESIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan bahwa penggunaan fuzzy logic pada bagian fungsi keanggotaan bentuk trapesium melalui beberapa proses, seperti fuzzification, inference, dan defuzzification kita akan mendapatkan solusi yang optimal untuk memilih sepuluh bengkel terbaik di kota Bandung. Penggunaan metode defuzzifikasi sugeno dapat disebut sebagai jalan terbaik dan cocok untuk digunakan sebagai pengambil keputusan dalam pengerjaan tugas kali ini. Berikut hasil yang menandakan bahwa proses fuzzy logic yang kami buat sudah benar dan cocok dengan data yang dimiliki.

	id	servis	harga	result
23	24	100	9	100.0
41	42	94	10	100.0
68	69	86	10	100.0
30	31	74	9	75.0
15	16	82	6	75.0
82	83	69	8	75.0
24	25	61	10	75.0
69	70	78	8	75.0
21	22	79	9	75.0
94	95	63	8	75.0

Source Code Program :

https://colab.research.google.com/drive/1-HfKdW0iiIxtQ17ZQb-QqiY3L_qQ5-C?usp=sharing

Video presentasi program : <https://youtu.be/kaNbIIFJTHY>