

LAPORAN TUGAS BESAR

Diajukan Untuk Memenuhi Salah Satu Tugas Mata Kuliah Kecerdasan Buatan



Disusun oleh:

- | | |
|------------------------------|--------------|
| 1. Nikita Farah Andrea | (1305220013) |
| 2. Chacha Alisha Dewintasari | (1305220036) |
| 3. Jason Kusuma | (1305220095) |

**Program Studi Sains Data
Fakultas Informatika
Universitas Telkom**

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Dengan rasa syukur dan hormat, kami sebagai mahasiswa Program Studi Sains Data mengucapkan salam yang tulus. Sebagai bagian dari perkuliahan mata kuliah Kecerdasan Buatan, kami dengan penuh antusiasme ingin mempersembahkan laporan tugas besar kami mengenai *Reasoning Fuzzy Logic*.

Laporan ini disusun sebagai salah satu syarat untuk melengkapi penilaian tugas mata kuliah Kecerdasan Buatan di program studi Sains Data Fakultas Informatika Universitas Telkom.

Pada laporan ini kami akan membahas mengenai sistem *fuzzy logic* dalam penentuan data terbaik, dari dataset yang telah diberikan. Dataset yang digunakan dalam tugas kali ini adalah data bengkel mobil di kota Bandung dari file bengkel.xlsx.

Hasil tugas ini nantinya akan menunjukkan lima bengkel terbaik yang akan ditampilkan berdasarkan skor kelayakan yang ditentukan dengan metode inferensi dari kedua atribut pada dataset diatas.

Kami menyadari bahwa laporan tugas besar ini masih jauh dari kata sempurna, dan begitu banyak kekurangan. Oleh karena itu, kami mengharapkan adanya kritik dan saran yang membangun dari para pembaca, untuk perbaikan laporan di masa yang akan datang.

Akhir kata, kami berharap laporan tugas besar ini dapat memberikan manfaat dan kontribusi yang berarti bagi perkembangan ilmu kecerdasan buatan dan aplikasinya dalam pemecahan masalah dunia nyata.

Wassalamu'alaikum Wr. Wb.

Sabtu, 3 November 2023

Penyusun

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam kecerdasan buatan kita mempelajari cara untuk membuat komputer dapat berpikir dan bertindak selayaknya manusia. Salah satu cabang dari kecerdasan buatan yang berperan penting adalah *Reasoning* dengan *fuzzy logic*. *Reasoning fuzzy logic* sendiri merupakan proses pengambilan keputusan oleh komputer yang dilakukan menggunakan *fuzzy logic*.

Ide *fuzzy logic* sendiri pertama kali ditemukan oleh Lotfi Zadeh dari University of California, Barkley pada tahun 1960. *Fuzzy logic* adalah salah satu dari banyak cabang logika, khususnya pada kecerdasan buatan yang memungkinkan komputer untuk menangani konsep yang tidak pasti. Logika ini menggunakan nilai keanggotaan yang digunakan untuk mewakili tingkat kebenaran dari suatu hal. Nilai keanggotaan yang digunakan dalam *fuzzy logic* berkisar antara rentang nilai 0 hingga 1,

Dalam kecerdasan buatan *fuzzy logic* biasanya digunakan untuk meniru cara manusia berpikir dan mengambil keputusan. Logika ini berbeda pada logika pada umumnya yang hanya memiliki nilai kebenaran 0 dan 1, tetapi juga memiliki tingkat kebenaran menengah pada range 0 hingga 1. Oleh karena itu *fuzzy logic* biasanya digunakan untuk rekayasa pengambilan keputusan yang tidak pasti

Pada tugas besar ini, nantinya akan dijelaskan mengenai sistem berbasis *fuzzy logic* yang akan digunakan untuk memilih lima bengkel terbaik. Sistem tersebut akan menggunakan dua variabel yaitu, “kualitas servis” dan “harga”, selain itu untuk menentukan bengkel-bengkel terbaik akan digunakan juga metode inferensi untuk menentukan skor kelayakan dari tiap bengkel.

Skor kelayakan dari tiap bengkel akan didapatkan dengan menggunakan metode *defuzzification*, metode ini merupakan kebalikan dari metode *fuzzy* yang akan mengubah nilai *fuzzy* menjadi nilai *crisp*.

Output dari tugas besar ini akan berupa informasi lima bengkel dengan skor kelayakan terbaik, beserta atribut ID bengkel, kualitas servis, dan harga.

1.2 Rumusan Masalah

Berikut adalah rumusan masalah yang akan dibahas:

1. Jumlah dan nama linguistik setiap atribut *input*

2. Bentuk dan batas fungsi keanggotaan *input*
3. Aturan Inferensi
4. Metode *Defuzzification*
5. Bentuk dan batas fungsi keanggotaan *output*

1.3 Tujuan

Dapat menemukan solusi optimal dalam konteks permasalahan yang diberikan menggunakan *Fuzzy Logic* dan dapat:

1. Mempelajari penggunaan *fuzzy logic*
2. Memahami konsep *fuzzy logic* dalam penyelesaian masalah di dunia nyata
3. Memahami cara kerja *fuzzy logic* untuk melakukan *reasoning*
4. Dapat menggunakan inferensi dalam *fuzzy logic* untuk pengambilan keputusan

1.4 Manfaat

- Meningkatkan pemahaman tentang dunia yang tidak pasti. Logika fuzzy dapat digunakan untuk memodelkan dunia nyata yang penuh dengan ketidakpastian, seperti kondisi cuaca, kualitas udara, atau perilaku manusia. Dengan mempelajari fuzzy logic, kita dapat lebih memahami bagaimana dunia nyata bekerja dan bagaimana membuat keputusan yang tepat dalam kondisi yang tidak pasti.
- Meningkatkan kemampuan berpikir kritis dan memecahkan masalah. Logika fuzzy mengajarkan kita untuk berpikir dalam hal derajat kebenaran, bukan hanya benar atau salah. Hal ini dapat membantu kita untuk lebih objektif dalam menilai informasi dan membuat keputusan yang lebih baik.
- Memperluas pengetahuan tentang kecerdasan buatan. Logika fuzzy adalah salah satu cabang penting dari kecerdasan buatan (AI). Dengan mempelajari fuzzy logic, kita dapat meningkatkan pemahaman kita tentang AI dan mengembangkan aplikasi AI yang lebih canggih.

BAB II PERANCANGAN

Untuk memperoleh skor bengkel terbaik berdasarkan nilai kualitas servis dan harga, kita akan menggunakan algoritma logika fuzzy yang melibatkan tahapan *Fuzzification*, *Inference*, dan *Defuzzification*.

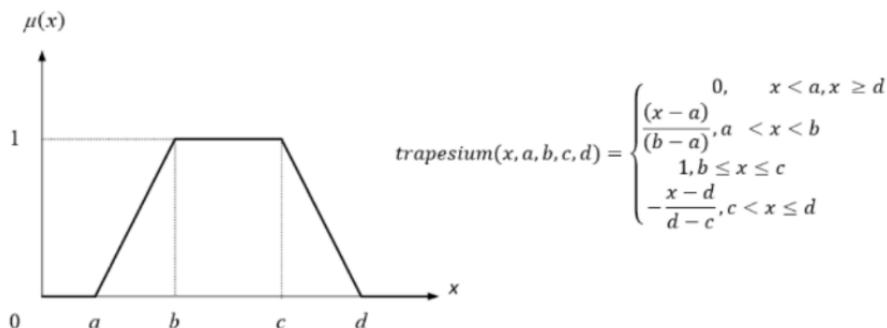
2.1 *Fuzzification*

Dalam proses fuzzifikasi, kami mendefinisikan nilai dan variabel linguistik beserta bentuk dan batas fungsi keanggotaan data input untuk mendapatkan *crisp value*.

2.1.1 Nilai dan Variabel linguistik

Dalam data bengkel terdapat dua variabel linguistik yaitu kualitas servis dan harga. Dalam variabel linguistik kualitas servis terdapat tiga nilai linguistik yaitu ‘Jelek’, ‘Biasa’ dan dalam variabel linguistik harga terdapat tiga nilai linguistik yaitu ‘Mahal’, ‘Sedang’ dan ‘Murah’.

2.1.2 Bentuk dan Batas Fungsi Keanggotaan



Kami menerapkan fungsi trapesium untuk variabel linguistik kualitas servis dan harga. Trapesium memiliki empat parameter yang menentukan bentuknya, yaitu batas bawah kiri (a), batas bawah kanan (d), batas atas kiri (b) dan batas atas kanan (c). Fungsi trapesium dapat dengan baik merepresentasikan interval fuzzy atau rentang fuzzy yang tidak hanya memiliki batas bawah dan batas atas tetapi juga dapat memiliki batas bawah dan batas atas yang sedikit kurang relevan atau signifikan.

Batasan fungsi keanggotaan untuk variabel kualitas servis:

- Jika $value \leq 40$ maka memiliki fungsi keanggotaan “Jelek”

- Jika $30 \leq value \leq 70$ maka memiliki fungsi keanggotaan “Biasa”
- Jika $60 \leq value \leq 100$ maka memiliki fungsi keanggotaan “Bagus”

Batasan fungsi keanggotaan untuk variabel harga:

- Jika $value \leq 5$ maka memiliki fungsi keanggotaan “Murah”
- Jika $3 \leq value \leq 9$ maka memiliki fungsi keanggotaan “Sedang”
- Jika $6 \leq value \leq 10$ maka memiliki fungsi keanggotaan “Mahal”

2.2 Inference

Bagian Inferensi adalah langkah-langkah logika yang digunakan untuk mengambil keputusan berdasarkan aturan-aturan yang telah ditentukan (Fuzzy Set). Pada bagian inferensi ini kami menggunakan metode Mamdani dengan menggunakan logika aturan fuzzy dan agregasi nilai fuzzy untuk menghasilkan output fuzzy.

Berikut adalah bentuk tabel dari Fuzzy Set:

		Servis		
		Jelek	Biasa	Bagus
Murah	Tidak Direkomendasikan	Direkomendasikan	Sangat Direkomendasikan	
Sedang	Tidak Direkomendasikan	Tidak Direkomendasikan	Sangat Direkomendasikan	
Harga	Mahal	Tidak Direkomendasikan	Tidak Direkomendasikan	Direkomendasikan

Kami menggunakan 9 aturan inferensi karena terdapat 3 nilai linguistik dari masing-masing variabel linguistik harga dan kualitas servis. Aturan inferensi diatur berdasarkan penilaian kualitas servis ‘Jelek’, ‘Biasa’ dan ‘Bagus’ serta penilaian harga yaitu ‘Murah’, ‘Sedang’, dan ‘Mahal’ menjadi suatu bentuk output fungsi keanggotaan seperti ‘Tidak Direkomendasikan’, ‘Direkomendasikan’ dan ‘Sangat Direkomendasikan’. Kami menggunakan logika konjungsi untuk melakukan proses inferensi. Adapun contoh dari proses inferensi adalah sebagai berikut:

- IF Servis = Jelek(1) AND harga = Murah(1) THEN status = Tidak Direkomendasikan(1)
- IF Servis = Biasa(0.25) AND harga = Murah(0.13) THEN status = Direkomendasikan(0.13)

Kemudian akan di cek ke 9 aturan inferensi dan akan mendapatkan hasil dari setiap status untuk setiap iterasi dari data. Dalam satu iterasi jika terdapat status yang sama dengan nilai bukan nol, maka akan dipilih nilai terbesarnya.

2.3 **Defuzzification**

Defuzzifikasi merupakan langkah terakhir dalam suatu sistem logika fuzzy dengan tujuan untuk mengkonversi setiap hasil dari inferensi yang masih diekspresikan dalam bentuk fuzzy menjadi suatu bilangan real

Pada proses Defuzzifikasi kami menggunakan metode centroid. Nilai cripis dihasilkan dengan menghitung titik tengah dari area dibawah kurva himpunan fuzzy. Centroid dihitung dengan menggunakan rumus

$$\text{Centroid} = \frac{(nilaiTidakDirekomendasikan \times 40) + (nilaiDirekomendasikan \times 70) + (nilaiSangatDirekomendasikan \times 100)}{\text{Total Nilai}}$$

Pada rumus di atas, nilai 40, 70 dan 100 mewakili pusat dari masing masing interval untuk label ‘Tidak Direkomendasikan’, ‘Direkomendasikan’ dan ‘Sangat Direkomendasikan’

BAB III IMPLEMENTASI

1. Import Data & Import Library

✓ Import Data

```
[ ] import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from google.colab import files

[ ] upload = files.upload()
data = pd.read_excel('bengkel.xlsx')

data.head()

Choose Files no files selected Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving bengkel.xlsx to bengkel (1).xlsx
```

	id	servis	harga
0	1	58	7
1	2	54	1
2	3	98	2
3	4	52	4
4	5	11	4

2. Fuzzification

Terdapat 2 variabel linguistik yaitu servis dan harga.

a) Fungsi Keanggotaan Kualitas Servis

```
[ ] # Fungsi Keanggotaan Kualitas Servis
def fuzzificationService(value):
    kualitasServis = {"Jelek": 0, "Biasa": 0, "Bagus": 0}
    a,b,c,d,e = 30, 40, 60, 70, 100

    # garis datar
    if value <= a:
        kualitasServis["Jelek"] = 1
    if value >= b and value <= c:
        kualitasServis["Biasa"] = 1
    if value >= d and value <= e:
        kualitasServis["Bagus"] = 1

    #garis miring
    if value > a and value <= b:
        kualitasServis["Jelek"] = max(0, min(-(value - b) / (b - a), 1))
    if value > a and value <= b:
        kualitasServis["Biasa"] = max(0, min((value - a) / (b - a), 1))
    if value > c and value < d:
        kualitasServis["Biasa"] = max(0, min(-(value - d) / (d - c), 1))
    if value > c and value < d:
        kualitasServis["Bagus"] = max(0, min((value - c) / (d - c), 1))

    return kualitasServis
```

Dalam fungsi ini terdapat variabel kualitasServis yang menyimpan nilai keanggotaan untuk tiga himpunan fuzzy, yaitu “Jelek”, “Biasa”, “Bagus”. Sedangkan variabel a,b,c,d,e digunakan untuk mendefinisikan nilai-nilai batas untuk membagi domain nilai input menjadi tiga himpunan dalam fuzzy.

Jika nilai (value) berada dalam rentang tertentu, maka nilai keanggotaan himpunan fuzzy diatur menjadi 1. Namun, jika nilai (value) berada diantara

dua titik batas nilai keanggotaan untuk himpunan fuzzy diatur menggunakan fungsi linear (garis miring) untuk menentukan seberapa dekat nilai tersebut dengan batas atas atau batas bawah himpunan fuzzy yang bersangkutan.

Fungsi *fuzzificationService* ini akan mengembalikan nilai dari variabel kualitasServis yang berisi informasi tentang keanggotaan kualitas servis (“Jelek”, “Biasa”, “Bagus”) berdasarkan nilai yang diberikan ke dalam fungsi.

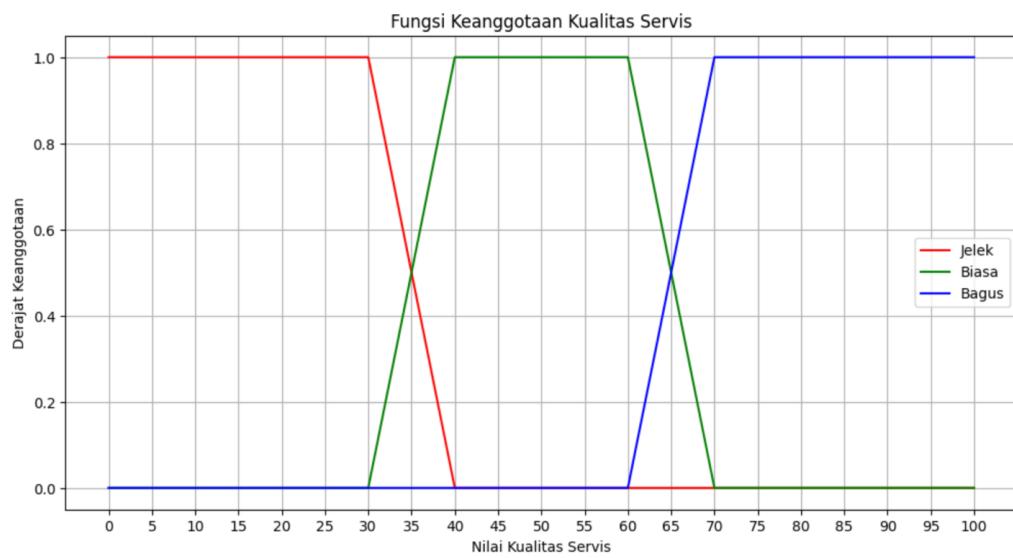
Berikut adalah diagram nilai kualitas servis menggunakan fungsi trapesium,

```
plt.figure(figsize=(12, 6))
plt.title('Fungsi Keanggotaan Kualitas Servis')

plt.plot(range(101),[fuzzificationService(x)["Jelek"] for x in range(101)], "r")
plt.plot(range(101),[fuzzificationService(x)[ "Biasa"] for x in range(101)], "g")
plt.plot(range(101),[fuzzificationService(x)[ "Bagus"] for x in range(101)], "b")

plt.plot([0, 0, "r", linewidth = 1.5, label = "Jelek"]
plt.plot([0, 0, "g", linewidth = 1.5, label = "Biasa"]
plt.plot([0, 0, "b", linewidth = 1.5, label = "Bagus"]

plt.xticks(np.arange(0, 105, 5.0))
plt.xlabel('Nilai Kualitas Servis')
plt.ylabel('Derajat Keanggotaan')
plt.legend()
plt.grid(True)
plt.show()
```



b) Fungsi Keanggotaan Harga

```
[ ] # Fungsi Keanggotaan Harga
def fuzzificationPrice(value):
    harga = {"Murah": 0,"Sedang": 0,"Mahal": 0}

    a,b,c,d,e = 3,5,6,8,10

    # garis datar
    if value <= a:
        harga["Murah"] = 1
    if value >= b and value <= c:
        harga["Sedang"] = 1
    if value >= d and value <= e:
        harga["Mahal"] = 1

    # garis miring
    if value > a and value < b:
        harga["Murah"] = max(0, min(-(value - b) / (b - a), 1))
    if value > a and value < c:
        harga["Sedang"] = max(0, min((value - a) / (b - a), 1))
    if value > c and value < d:
        harga["Sedang"] = max(0, min(-(value - d) / (d - c), 1))
    if value > c and value < e:
        harga["Mahal"] = max(0, min((value - c) / (d - c), 1))

    return harga
```

Dalam fungsi ini terdapat variabel harga yang menyimpan nilai keanggotaan untuk tiga himpunan fuzzy, yaitu “Murah”, “Sedang”, “Mahal”. Sedangkan variabel a,b,c,d,e digunakan untuk mendefinisikan nilai-nilai batas untuk membagi domain nilai input menjadi tiga himpunan dalam fuzzy.

Jika nilai (value) berada dalam rentang tertentu, maka nilai keanggotaan himpunan fuzzy diatur menjadi 1. Namun, jika nilai (value) berada diantara dua titik batas nilai keanggotaan untuk himpunan fuzzy diatur menggunakan fungsi linear (garis miring) untuk menentukan seberapa dekat nilai tersebut dengan batas atas atau batas bawah himpunan fuzzy yang bersangkutan.

Fungsi *fuzzificationPrice* ini akan mengembalikan nilai dari variabel harga yang berisi informasi tentang keanggotaan kualitas servis (“Murah”, “Sedang”, “Mahal”) berdasarkan nilai yang diberikan ke dalam fungsi.

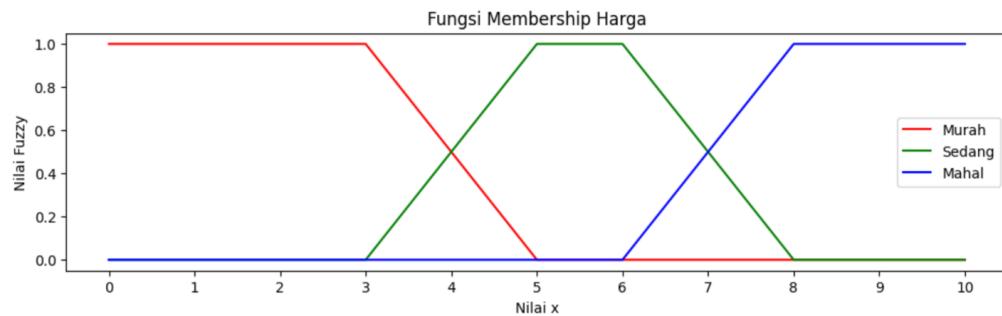
Berikut adalah diagram nilai harga menggunakan fungsi trapesium,

```
[ ] # Diagram nilai harga
plt.figure(figsize=(12,3))
plt.title("Fungsi Membership Harga")

plt.plot(range(11),[fuzzificationPrice(x)__["Murah"] for x in range(11)], "r")
plt.plot(range(11),[fuzzificationPrice(x)__["Sedang"] for x in range(11)], "g")
plt.plot(range(11),[fuzzificationPrice(x)__["Mahal"] for x in range(11)], "b")

plt.plot(0, 0, "r", linewidth = 1.5, label = "Murah")
plt.plot(0, 0, "g", linewidth = 1.5, label = "Sedang")
plt.plot(0, 0, "b", linewidth = 1.5, label = "Mahal")

plt.xticks(list(range(11)))
plt.xlabel("Nilai x")
plt.ylabel("Nilai Fuzzy")
plt.legend()
plt.show()
```



c) Proses *Fuzzification*

```
[ ] hasilFuzzifikasi = []

for servis, harga in zip(data["servis"], data["harga"]):
    nilaiFuzzy = {"Kualitas Servis": 0, "Harga": 0}
    nilaiFuzzy["Kualitas Servis"] = fuzzificationService(servis)
    nilaiFuzzy["Harga"] = fuzzificationPrice(harga)

    hasilFuzzifikasi.append(nilaiFuzzy)
```

Melalui perulangan ‘for’ nilai servis dan harga diambil dari setiap baris data yang berada di kolom servis dan harga secara bersamaan dengan menggunakan fungsi zip untuk menggabungkan keduanya.

Proses ini berakhir dengan hasilFuzzifikasi yang berisi hasil dari setiap nilai yang telah diproses melalui fungsi fuzzifikasi. List ini akan berisi kamus-kamus yang memuat nilai keanggotaan untuk kualitas servis dan harga dari setiap pasangan nilai servis dan harga

Berikut adalah tampilan hasil dari proses *fuzzyfication*,

```

▶ # Menampilkan hasil dari proses Fuzzifikasi 5 data awal
i = 0
for i in range(5):
    print("Kualitas Servis: ", data["servis"][i])
    print("Harga: ", data["harga"][i])
    print("Nilai Kualitas Servis: ", hasilFuzzifikasi[i]["Kualitas Servis"])
    print("Nilai Harga: ", hasilFuzzifikasi[i]["Harga"], end = "\n\n")

Kualitas Servis: 58
Harga: 7
Nilai Kualitas Servis: {'Jelek': 0, 'Biasa': 1, 'Bagus': 0}
Nilai Harga: {'Murah': 0, 'Sedang': 0.5, 'Mahal': 0.5}

Kualitas Servis: 54
Harga: 1
Nilai Kualitas Servis: {'Jelek': 0, 'Biasa': 1, 'Bagus': 0}
Nilai Harga: {'Murah': 1, 'Sedang': 0, 'Mahal': 0}

Kualitas Servis: 98
Harga: 2
Nilai Kualitas Servis: {'Jelek': 0, 'Biasa': 0, 'Bagus': 1}
Nilai Harga: {'Murah': 1, 'Sedang': 0, 'Mahal': 0}

Kualitas Servis: 52
Harga: 4
Nilai Kualitas Servis: {'Jelek': 0, 'Biasa': 1, 'Bagus': 0}
Nilai Harga: {'Murah': 0.5, 'Sedang': 0.5, 'Mahal': 0}

Kualitas Servis: 11
Harga: 4
Nilai Kualitas Servis: {'Jelek': 1, 'Biasa': 0, 'Bagus': 0}
Nilai Harga: {'Murah': 0.5, 'Sedang': 0.5, 'Mahal': 0}

```

3. Inference

```

[ ] fuzzySet = {
    ('Murah', 'Jelek'): 'Tidak Direkomendasikan',
    ('Sedang', 'Jelek'): 'Tidak Direkomendasikan',
    ('Mahal', 'Jelek'): 'Tidak Direkomendasikan',
    ('Murah', 'Biasa'): 'Direkomendasikan',
    ('Sedang', 'Biasa'): 'Tidak Direkomendasikan',
    ('Mahal', 'Biasa'): 'Tidak Direkomendasikan',
    ('Murah', 'Bagus'): 'Sangat Direkomendasikan',
    ('Sedang', 'Bagus'): 'Sangat Direkomendasikan',
    ('Mahal', 'Bagus'): 'Direkomendasikan',
}

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

    for servis_label in fuzzed['Kualitas Servis']:
        for harga_label in fuzzed['Harga']:
            result["Nilai Fuzzy"] = {"Kualitas Servis": fuzzed["Kualitas Servis"], "Harga": fuzzed["Harga"]}

            # Assuming 'Fuzzy Harga' should be 'Harga'
            min_value = min(fuzzed["Harga"][harga_label], fuzzed['Kualitas Servis'][servis_label])

            # Proses Konjungsi sesuai fuzzy set rules
            output_label = fuzzySet[(harga_label, servis_label)]

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

    return result

```

Variabel `fuzzySet` merupakan sebuah kamus yang berisi aturan inferensi untuk sistem logika fuzzy. Aturan-aturan ini menyatakan hasil atau rekomendasi berdasarkan pasangan nilai harga dan kualitas servis.

Pada fungsi `interference`, ia hanya menerima satu parameter, yaitu `fuzzed`. Parameter ini diasumsikan sebagai kamus yang berisi nilai-nilai fuzzy untuk kualitas servis dan harga.

Di dalam fungsi interference terdapat dua kali perulangan menggunakan for, fungsi ini melakukan iterasi melalui label-label dari nilai fuzzifikasi kualitas servis dan harga. Kemudian, menyimpan nilai-nilai tersebut di dalam kamus result.

Variabel min_value mengambil nilai minimum antara nilai fuzzifikasi harga dan kualitas servis yang sedang dievaluasi. Sedangkan, variabel output_label mencari aturan inferensi yang sesuai berdasarkan pasangan label dari harga dan kualitas servis.

Jika nilai min_value kurang dari output_label maka akan diperbarui nilai pada kamus result sesuai dengan hasil inferensi yang didapatkan dari aturan inferensi dengan nilai fuzzy minimum.

Fungsi ini akan mengembalikan hasil dari proses inferensi yang telah disimpan di dalam kamus result.

Berikut adalah hasil dari proses interference pada 5 data paling awal,

```
[ ] # Proses Inference
hasilInterference = []

for fuzzed in hasilFuzzifikasi:
    hasilInterference.append(interference(fuzzed))

# Menampilkan hasil dari proses Interference 5 data awal
i = 0
for i in range(5):
    print({key: value for key, value in hasilInterference[i].items() if key in ["Tidak Direkomendasikan", "Direkomendasikan", "Sangat Direkomendasikan"]})

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

4. Defuzzyfication

```
[ ] def defuzzification(interference_result):
    nilai_direkomendasikan = interference_result["Direkomendasikan"]
    nilai_tidak_direkomendasikan = interference_result["Tidak Direkomendasikan"]
    nilai_sangat_direkomendasikan = interference_result["Sangat Direkomendasikan"]

    total_nilai = nilai_direkomendasikan + nilai_tidak_direkomendasikan + nilai_sangat_direkomendasikan

    if total_nilai != 0:
        centroid = (nilai_tidak_direkomendasikan * 40 + nilai_direkomendasikan * 70 + nilai_sangat_direkomendasikan * 100.0) / total_nilai
    else:
        centroid = 0

    return centroid
```

Fungsi defuzzification hanya menerima satu parameter yaitu interference_result, yang diasumsikan berisi hasil dari proses interferensi pada sistem logika fuzzy.

Pada variabel total_nilai akan menyimpan nilai yang didapatkan dari hasil inferensi untuk setiap label yang direkomendasikan, tidak direkomendasikan, dan sangat direkomendasikan dari hasil proses interferensi yang diberikan ke dalam

fungsi. Dan variabel total_nilai akan menghitung total nilai dari ketiga kategori hasil inferensi.

Jika total_nilai bukan 0, maka akan dilakukan perhitungan centroid. Rumus ini menghitung nilai centroid dengan mengalikan setiap nilai inferensi dengan titik-titik centroid yang diasosiasikan dengan label masing-masing, kemudian dibagi dengan total nilai dari semua hasil inferensi. Namun, jika total_nilai adalah 0, maka centorid bernilai 0.

Fungsi ini akan mengembalikan nilai centroid yang telah dihitung sebagai hasil dari proses defuzzifikasi.

Selanjutnya adalah proses defuzzification,

```
[ ] # Proses Deffuzification
hasilDeffuzifikasi = []

for interference_result in hasilInterference:
    hasilDeffuzifikasi.append(defuzzification(interference_result))
```

Kode ini bertujuan untuk melakukan proses deffuzifikasi pada setiap hasil interferensi yang ada dalam list hasilInterference, dan hasil deffuzifikasi tersebut disimpan dalam list hasilDeffuzifikasi. Setelah selesai iterasi, list hasilDeffuzifikasi akan berisi nilai-nilai konkret yang telah dihasilkan dari proses deffuzifikasi, yang dapat digunakan untuk melakukan interpretasi atau pengambilan keputusan lebih lanjut dalam aplikasi logika fuzzy tersebut.

Berikut adalah hasil dari proses *defuzzyfication* pada 5 data paling awal,

```
▶ # Menampilkan hasil deffuzifikasi 5 data awal
i = 0
for i in range(5):
    print("Hasil Deffuzifikasi:", hasilDeffuzifikasi[i])

Hasil Deffuzifikasi: 40.0
Hasil Deffuzifikasi: 70.0
Hasil Deffuzifikasi: 100.0
Hasil Deffuzifikasi: 55.0
Hasil Deffuzifikasi: 40.0
```

5. Output

Berikut adalah hasil akhir dari proses implementasi,

```

[ ] data['Skor Kelayakan'] = hasilDefuzzifikasi
data = data.sort_values(by=['Skor Kelayakan', 'servis', 'harga'], ascending=[False, False, True])[:5]
data[['id', 'servis', 'harga', 'Skor Kelayakan']].to_excel('peringkat.xls', index=False, header=True, engine='openpyxl')
data

  id  servis  harga  Skor Kelayakan
2   3      98     2       100.0
90  91      98     3       100.0
51  52      94     3       100.0
33  34      93     4       100.0
91  92      83     3       100.0

```

Kode tersebut bertujuan untuk mengurutkan skor kelayakan secara descending, servis secara descending, dan harga secara ascending untuk memperoleh 5 bengkel terbaik yang dinilai berdasarkan skor kelayakan yang tinggi, kualitas servis yang tinggi, dan harga yang murah.

BAB IV KESIMPULAN

Pada laporan ini telah dijelaskan langkah langkah penggunaan *fuzzy logic* dalam mencari lima bengkel terbaik. Dalam pencarinya digunakan variabel input yaitu kualitas servis dan harga, serta variabel kualitas servis yang menampung nilai keanggotaan. Terdapat tiga tahapan utama dalam sistem ini, yaitu *fuzzification*, *inference*, dan *defuzzification*.

Pada tahapan fuzzifikasi akan diubah input yang berupa nilai *crisp* menjadi nilai *fuzzy*. Kemudian untuk tahapan inferensi akan digunakan inferensi dengan metode mamdani, pada model ini nilai *fuzzy* dari input akan digabungkan menggunakan logika konjungsi untuk menghasilkan nilai *fuzzy* dari output yang diinginkan. Selanjutnya tahapan terakhir defuzzifikasi, pada tahapan ini nilai *fuzzy* yang dihasilkan akan diubah kembali menjadi nilai *crisp*. Metode *crisp* yang paling sering digunakan adalah menggunakan metode *centroid*.

Hasil yang dihasilkan dari proses ini akan berupa data dari lima bengkel dengan nilai kelayakan terbaik berdasarkan hasil pengujian dengan metode fuzzifikasi. Dari hasil tersebut dapat dilihat kelima bengkel terbaik adalah sebagai berikut :

Bengkel 3, dengan kualitas servis 98, harga 2, dan skor kelayakan 100

Bengkel 91, dengan kualitas servis 98, harga 3, dan skor kelayakan 100

Bengkel 52, dengan kualitas servis 94, harga 3, dan skor kelayakan 100

Bengkel 34, dengan kualitas servis 93, harga 4, dan skor kelayakan 100

Bengkel 92, dengan kualitas servis 83, harga 3, dan skor kelayakan 100

Dengan demikian sistem *fuzzy logic* yang dibuat dalam tugas besar kali ini dapat dijadikan cara yang efektif dalam mencari bengkel dengan kualitas terbaik. Selain itu sistem ini juga dapat menangani kriteria lain dalam mencari penilaian yang disesuaikan dengan kebutuhan pengguna.

PERAN ANGGOTA KELOMPOK

Jason Kusuma	: Source Code, Laporan dan ppt bab 1 & 4
Chacha Alisha Dewintasari	: Source Code, Laporan dan ppt bab 2
Nikita Farah Andrea	: Source Code, Laporan dan ppt bab 3