

LAPORAN : TUGAS PEMROGRAMAN 02 IMPLEMENTASI FUZZY LOGIC DALAM PEMERINGKATAN RESTORAN

Kelompok A - IF4302

Kaenova Mahendra Auditma (1301190324)¹

Adhe Akram Azhari (1301194063)²

Elita Aurora Az Zahra (1301194127)³

Pengantar Kecerdasan Buatan (CII2M3-IF-43-02)

Informatika, FT, Universitas Telkom. 2021.

email:

¹kaenova@student.telkomuniversity.ac.id

²adheakramazhari@student.telkomuniversity.ac.id

³elitaaurora@student.telkomuniversity.ac.id

1. PENDAHULUAN

Tugas Pemrograman 02 merupakan salah satu dari ketiga proyek tugas yang ada pada Mata Kuliah Pengantar Kecerdasan Buatan. Pada tugas ini, kami diminta untuk membuat suatu algoritma dimana akan memeringkatkan 10 restoran terbaik dari 100 restoran berdasarkan penilaian makanan dan pelayanan dari restoran tersebut. Dengan data-data tersebut kami diminta untuk menerapkan algoritma *Fuzzy Logic*.

Fuzzy Logic merupakan salah satu algoritma yang ada di dalam kecerdasan buatan. Algoritma ini merupakan salah satu konsep dari algoritma *Reasoning Logic* dimana akan mengembalikan nilai derajat kebenaran dari input-input yang tidak pasti. Hal ini biasa diterapkan untuk menyelesaikan masalah-masalah yang sifatnya kualitatif.

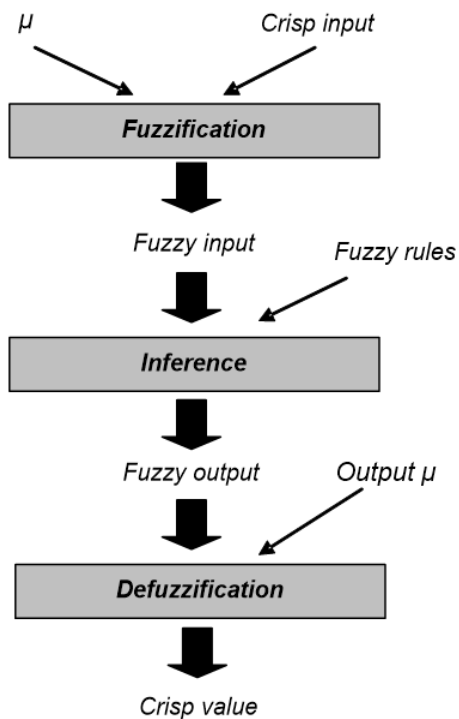
Dalam penyelesaian algoritma ini ada beberapa hal yang perlu diobservasi seperti:

- Jumlah dan Nama Linguistik setiap input.
- Bentuk dan Batas Fungsi Keanggotaan Input.
- Aturan Inferensi.
- Metode Defuzzifikasi.
- Bentuk dan Batas Fungsi Keanggotaan Output.

Hal-hal tersebut yang menjadi faktor sehingga mempengaruhi hasil akhir dari pemeringkatan yang akan dilakukan oleh algoritma *Fuzzy Logic* ini.

2. RANCANGAN ALGORITMA

Fuzzy logic merupakan suatu cara untuk memetakan suatu ruang input ke dalam ruang output. Secara umum, *fuzzy logic* cocok untuk menyelesaikan masalah-masalah yang sulit didefinisikan dengan model matematis atau masalah yang mengandung ketidakpastian. Keuntungan dari *fuzzy logic* yaitu sistem ini mempunyai kemampuan dalam penalaran yang sama dengan penalaran manusia karena sistem ini punya kemampuan untuk memberikan respon berdasarkan informasi yang kualitatif, tidak akurat, dan ambigu.



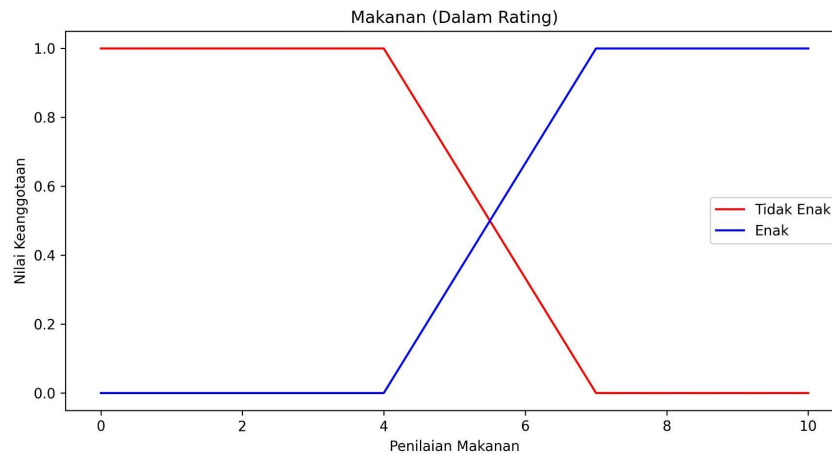
Gambar 1.1

Overview fuzzy logic

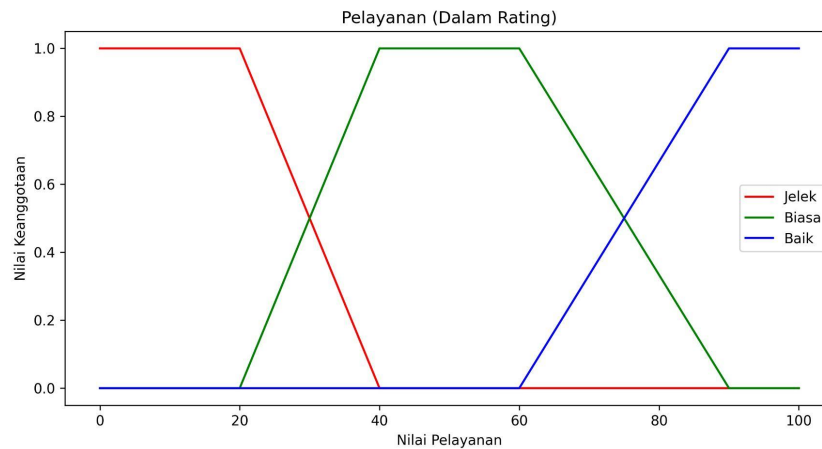
Dapat dilihat pada **Gambar 1.1**, tahapan pada *fuzzy logic* ada tiga, yaitu *fuzzification*, *inference*, dan *defuzzification*. Pada *fuzzification* terdapat inputan berupa μ dan crisp input, kemudian crisp input diubah menjadi *fuzzy input* yang berupa nilai linguistik. Pada *inference* kita melakukan penalaran menggunakan *fuzzy input* dan *fuzzy rules* yang telah ditentukan sehingga menghasilkan *fuzzy output*. Terdapat dua model *inference* yang bisa digunakan, yaitu model mamdani (intuitif mengikuti perasaan manusia) dan model sugeno (lebih sesuai untuk sistem kontrol). Langkah yang terakhir yaitu *defuzzification* yaitu terdapat dua input yaitu output μ dan *fuzzy output* yang diubah menjadi *crisp value* berdasarkan fungsi keanggotaan yang telah ditentukan.

2.1. Fuzzifikasi

Fuzzifikasi merupakan tahap pertama dari algoritma *Fuzzy Logic*. Pada tahap ini, kami merubah nilai *crisp input* dengan menggunakan μ menjadi *fuzzy input* yang berupa nilai linguistik. Pada soal, terdapat dua atribut yaitu kualitas makanan dengan skala rating bilangan *real* 0-10 dan kualitas pelayanan dengan skala rating bilangan *real* 0-100. Pada tahap ini, kami menentukan sendiri bentuk dan batasan pada fungsi kualitas makanan dan kualitas pelayanan.



Gambar 2.1
Grafik Fungsi Keanggotaan Kualitas Makanan



Gambar 2.2
Grafik Fungsi Keanggotaan Kualitas Pelayanan

Pada **Gambar 2.1** merupakan grafik fungsi keanggotaan dari kualitas makanan. Kami menentukan bentuk trapesium dengan batasan makanan tidak enak pada skala 0 sampai 4 bilangan *real* dan makanan enak dengan skala 7 sampai 10 bilangan *real*. Pada **Gambar 2.2** merupakan grafik fungsi kualitas pelayanan. Disini kami menentukan bentuk trapesium dengan batasan kualitas pelayanan jelek pada skala 0 sampai 20 bilangan *real*, batasan kualitas pelayanan biasa pada skala 40 sampai 60 bilangan *real*, dan batasan kualitas pelayanan baik dengan skala 90 sampai 100 bilangan *real*.

Pada fuzzifikasi, kami melakukan konversi nilai pada kualitas makanan dan kualitas pelayanan dengan nilai yang sudah ditentukan menjadi nilai fuzzy menggunakan rumus yang sudah ditentukan sesuai dengan bentuk trapesium.

2.2. Inferensi

Inferensi merupakan tahap kedua dari algoritma *Fuzzy Logic*. Pada tahap ini kami membandingkan pelayan dan makanan untuk menentukan nilai bagus atau buruk. Dengan syarat dimana nilai dari fuzzifikasi sudah diinputkan. Pada tahap ini kami menentukan sendiri hasil dari

perbandingan makanan dan pelayanan. Contoh jika pelayan jelek dan makanan tidak enak maka tentu saja kesimpulannya buruk.

MAKANAN/PELAYANAN	Jelek	Biasa	Baik
Tidak enak	Buruk	Buruk	Buruk
Enak	Buruk	Bagus	Bagus

Gambar 2.3

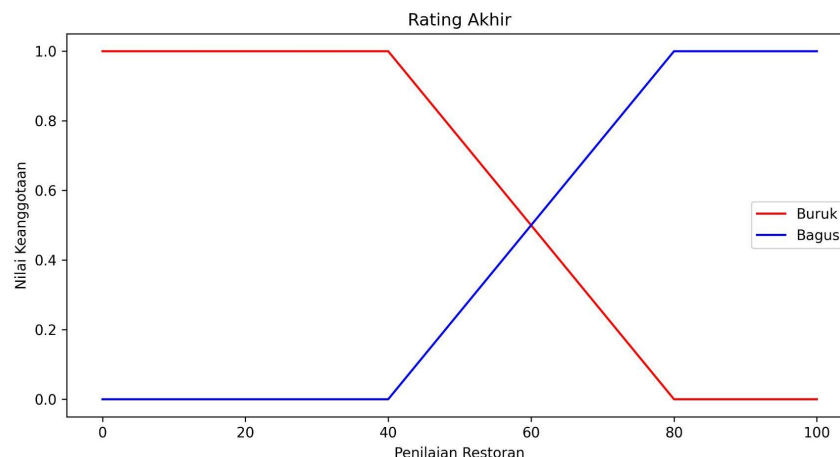
Tabel Inferensi

Pada **Gambar 2.3** dapat dilihat tabel tersebut telah memperbandingkan antara pelayanan dan makanan (tidak enak/jelek, tidak enak/biasa, tidak enak/baik, enak/jelek, enak/biasa, enak/baik) sehingga dapat menghasilkan nilai buruk atau bagus.

2.3. Defuzzifikasi

Defuzzifikasi merupakan tahap terakhir dari algoritma *Fuzzy Logic* ini, dimana setelah didapatkan hasil inferensi atau *Fuzzy Output* pada tahap sebelumnya, di sini kami merubah nilai-nilai yang ada pada *Fuzzy Output* tersebut menjadi nilai yang pasti dan jelas. Pada soal yang diberikan, kami diminta untuk menjadikan nilai-nilai yang jelas yaitu dari skala 1-100.

Untuk melakukan defuzzifikasi ini dibutuhkan fungsi keanggotaan. Pada langkah ini kami menentukan sendiri bagaimana bentuk dan batas fungsi keanggotaan tersebut. Kami menyepakati pada langkah inferensi bahwa akan ada dua nilai linguistik pada fungsi keanggotaan ini, yaitu *bagus* dan *buruk* seperti. Serta fungsi keanggotaan seperti berikut:

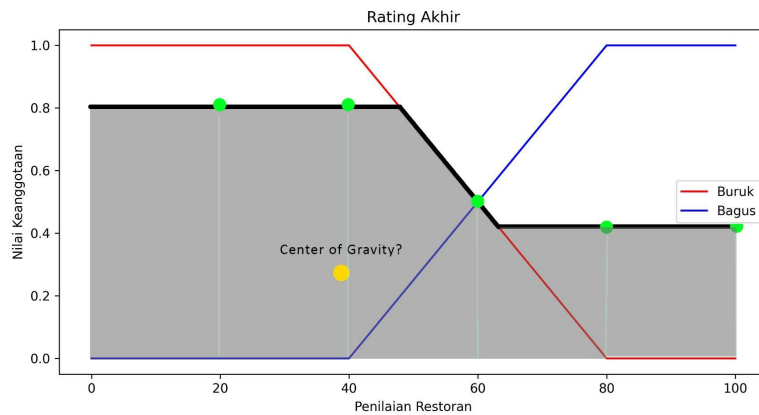


Gambar 2.4

Grafik Fungsi Keanggotaan Output

Pada **Gambar 2.3** dapat terlihat bahwa bentuk dari fungsi ini merupakan bentuk trapesium, dimana nilai buruk memiliki derajat keanggotaan bernilai 1 dari 1-40 lalu akan secara linear turun hingga bernilai 0 pada rating bernilai 80 dan akan tetap bernilai 0 hingga rating 100. Sebaliknya dari nilai buruk, yaitu bagus, dimana memiliki derajat keanggotaan bernilai 0 dari 1-40 lalu akan secara linear naik hingga bernilai 1 pada rating bernilai 80 dan akan tetap bernilai 1 hingga rating 100.

Perhitungan nilai akhir yang kami gunakan pada langkah defuzzifikasi ini menggunakan model Mamdani. Model Mamdani ini menggunakan metode *centroid* pada langkah defuzzifikasi. Metode ini bekerja dengan mencari titik berat dari fungsi keanggotaan suatu restoran.



Gambar 2.5

Grafik Fungsi Keanggotaan Output (Sampling = 5 ; $\mu_{\text{Buruk}} = 0.8$; $\mu_{\text{Bagus}} = 0.4$)

Kasus perhitungan metode centroid pada algoritma kami serupa dengan **Gambar 2.4**. Pada gambar tersebut terlihat jika adanya hasil dari inferensi salah satu restoran memberikan nilai keanggotaan buruk sebesar 0.8 dan nilai keanggotaan bagus 0.4 maka akan dihitung titik berat dari grafik fungsi keanggotaan tersebut dengan perhitungan seperti di bawah:

$$\text{Titik Berat} = \frac{20 \cdot 0.8 + 40 \cdot 0.8 + 60 \cdot 0.45 + 80 \cdot 0.4 + 100 \cdot 0.4}{3(0.8) + 0.45 + 2(0.4)} \quad (2.6)$$

Pada algoritma yang kami gunakan, sampling yang diatur yaitu sebesar 20 tahap yang artinya titik penilaian restoran yang di observasi dimulai dari 5 hingga 100 dengan besar pertambahan sebesar 5. Dari nilai titik berat tersebut maka akan menjadi nilai akhir atau *crisp output* dari suatu restoran. Hal ini dilakukan kepada seluruh restoran sehingga didapatkan nilai-nilai *crisp output* dari setiap restoran, sehingga dapat diperingkatkan 10 restoran dengan nilai *crisp output* terbaik.

3. IMPLEMENTASI ALGORITMA DAN HASIL PROGRAM

Implementasi dari semua hal yang dijelaskan pada bagian 2 akan dibuat pada bahasa pemrograman Python. Dengan menggunakan bahasa pemrograman Python, kami tidak memisahkan bagian fungsi-fungsi yang dibuat dengan fungsi utama, melainkan setelah mendefinisikan fungsi tersebut kami langsung menggunakannya. Untuk melihat pemrograman secara terpisah, bisa membuka source code [main.py](#).

Kami hanya menggunakan beberapa *library* luar yang digunakan hanya untuk *input/output* data dan pembuatan grafik.

3.1. Membaca File dan Definisi Kelas Restoran

Tahap pertama dari pengimplementasian *Fuzzy Logic* ini yaitu memasukkan data-data yang dibutuhkan, yaitu penilaian restoran dan penilaian makanan dari setiap restoran. Setelah itu kami juga membuat kelas bernama restoran yang mana akan membantu kami dalam mengolah data-data pada setiap restoran. Setiap data-data restoran yang berasal dari data eksternal tersebut akan dimasukkan ke dalam list dengan kelas restoran.

```

import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_excel('../data/raw/restoran.xlsx')
df.head()

class restoran:
    def __init__(self):
        self.id_restoran = 0
        self.pelayanan = 0
        self.makanan = 0

        # Fuzzifikasi (Fuzzy Input)
        self.makanan_enak = -1          # Range [7, 10]
        self.makanan_tidak enak = -1    # Range [1, 4]

        self.pelayanan_baik = -1        # Range [90, 100]
        self.pelayanan_biasa = -1       # Range [40, 60]
        self.pelayanan_jelek = -1       # Range [1, 20]

        # Inferensi (Fuzzy Output)
        self.bagus = []                 # Masukkan Semua Nilai Yang Mungkin untuk Bagus
        self.buruk = []                 # Masukkan Semua Nilai Yang Mungkin untuk Buruk
        self.bagus_akhir = 0            # Nilai Terkecil dari List Bagus
        self.buruk_akhir = 0           # Nilai Terkecil dari List Buruk

        # Defuzzifikasi (Crisp Input)
        self.nilai_akhir = 0

    def tampilData(self):
        print('id restoran: {} \n nilai pelayanan: {} \n nilai makanan: {} \n nilai
akhir: {} \n makanan_enak: {} \n makanan_tidak enak: {} \n pelayanan_baik: {} \n
pelayanan_biasa: {} \n pelayanan_jelek: {} \n persentase bagus: {} \n persentase
buruk: {}'.format(self.id_restoran, self.pelayanan, self.makanan, self.nilai_akhir,
self.makanan_enak, self.makanan_tidak enak, self.pelayanan_baik, self.pelayanan_biasa,
self.pelayanan_jelek, self.bagus_akhir, self.buruk_akhir))

restoran_data = []

#Masukkan data dari dataframe ke dalam List restoran data
for i in range(len(df)):
    restoran_temp = restoran()
    restoran_temp.id_restoran = df.iloc[i][0]
    restoran_temp.pelayanan = df.iloc[i][1]
    restoran_temp.makanan = df.iloc[i][2]
    restoran_data.append(restoran_temp)

```

3.2. Fuzzifikasi

Fungsi fuzzifikasi ini dibuat berdasarkan **Gambar 2.1** dan **Gambar 2.2** dengan melakukan percabangan. Input dari fungsi ini yaitu nilai kualitas makanan dan kualitas

pelayanan yang ada pada class restoran yang kemudian dibandingkan dengan fungsi keanggotaan yang sudah ditentukan sebelumnya. Kemudian output dari fungsi fuzzifikasi yaitu tingkat kualitas makanan dan tingkat kualitas pelayanan beserta dengan nilai *fuzzy* yang sudah ditentukan berdasarkan rumus yang sudah ditentukan.

```
def fuzzifikasi(restoran_temp):
    for i in range (len(restoran_temp)):
        # Perhitungan pelayanan
        if restoran_temp[i].pelayanan >= 1 and restoran_temp[i].pelayanan <= 20:
            restoran_temp[i].pelayanan_jelek = 1
        elif restoran_temp[i].pelayanan > 20 and restoran_temp[i].pelayanan < 40:
            restoran_temp[i].pelayanan_jelek = -(restoran_temp[i].pelayanan - 40)/(40 -
20)
            restoran_temp[i].pelayanan_biasa = (restoran_temp[i].pelayanan - 20)/(40 -
20)
        elif restoran_temp[i].pelayanan >= 40 and restoran_temp[i].pelayanan <= 60:
            restoran_temp[i].pelayanan_biasa = 1
        elif restoran_temp[i].pelayanan > 60 and restoran_temp[i].pelayanan < 90:
            restoran_temp[i].pelayanan_biasa = -(restoran_temp[i].pelayanan - 90)/(90 -
60)
            restoran_temp[i].pelayanan_baik = (restoran_temp[i].pelayanan - 60)/(90 -
60)
        elif restoran_temp[i].pelayanan >= 90 and restoran_temp[i].pelayanan <= 100:
            restoran_temp[i].pelayanan_baik = 1

        if restoran_temp[i].makanan >= 1 and restoran_temp[i].makanan <= 4:
            restoran_temp[i].makanan_tidakenak = 1
        elif restoran_temp[i].makanan > 4 and restoran_temp[i].makanan < 7:
            restoran_temp[i].makanan_tidakenak = -(restoran_temp[i].makanan - 7)/(7 - 4)
            restoran_temp[i].makanan_enak = (restoran_temp[i].makanan - 4)/(7 - 4)
        elif restoran_temp[i].makanan >= 7 and restoran_temp[i].makanan <= 10:
            restoran_temp[i].makanan_enak = 1

    return restoran_temp

# Melakukan Fuzzifikasi terhadap restoran_data
restoran_data = fuzzifikasi(restoran_data)
```

3.3. Inferensi

Pada fungsi ini kita mengacu terhadap **Gambar 2.3** yang dimana kita menentukan nilai bagus atau buruk. Dan tentunya makanan dan pelayanan merupakan nilai input dari fungsi ini dan outputnya adalah bagus atau buruk.

```
def inferensi(restoran_temp):
    for i in range (len(restoran_temp)):
        if restoran_temp[i].makanan_tidakenak != -1:
            if restoran_temp[i].pelayanan_jelek != -1:
                restoran_temp[i].buruk.append(min(restoran_temp[i].makanan_tidakenak, restoran_temp[i].pelaya
```

```

nan_jelek))
    if restoran_temp[i].pelayanan_biasa != -1:

restoran_temp[i].buruk.append(min(restoran_temp[i].makanan_tidakenak, restoran_temp[i].pelaya
nan_biasa))
    if restoran_temp[i].pelayanan_baik != -1:
        restoran_temp[i].buruk.append(min(restoran_temp[i].makanan_tidakenak,
restoran_temp[i].pelayanan_baik))
    if restoran_temp[i].makanan_enak != -1:
        if restoran_temp[i].pelayanan_jelek != -1:

restoran_temp[i].buruk.append(min(restoran_temp[i].makanan_enak, restoran_temp[i].pelayanan_j
elek))
    if restoran_temp[i].pelayanan_biasa != -1:

restoran_temp[i].bagus.append(min(restoran_temp[i].makanan_enak, restoran_temp[i].pelayanan_b
iasa))
    if restoran_temp[i].pelayanan_baik != -1:
        restoran_temp[i].bagus.append(min(restoran_temp[i].makanan_enak,
restoran_temp[i].pelayanan_baik))

    if len(restoran_temp[i].bagus) != 0:
        restoran_temp[i].bagus_akhir = max(restoran_temp[i].bagus)

    if len(restoran_temp[i].buruk) != 0:
        restoran_temp[i].buruk_akhir = max(restoran_temp[i].buruk)

    return restoran_temp

# Melakukan inferensi terhadap restoran_data
restoran_data = inferensi(restoran_data)

```

3.4. Defuzzifikasi

Berdasarkan bagian 2.4 setiap restoran akan dihitung titik berat dari setiap restoran. Pada fungsi ini memiliki input berupa banyaknya sampling yang digunakan dan restoran yang dihitung. Output yang dihasilkan adalah restoran yang pada variable nilai_akhir yang merupakan *crisp output* dari suatu restoran.

```

def defuzzifikasi(sample: int, rest: restoran) -> restoran:
    #Parameter fungsi keanggotaan output
    parm = {
        "batas_bawah" : 1,
        "buruk_c" : 40,
        "buruk_d" : 80,
        "bagus_a" : 40,
        "bagus_b" : 80,
        "batas_atas": 100
    }

```



```

        pembilang = 0
        penyebut = 0
        increment = 100 // sample
        for i in (range(sample-0)):
            bilangan = (i+1) * increment
            if bilangan <= parm['buruk_c']:          #Pengecekan jika berada dari titik
buruk_c ke bawah
                pembilang += bilangan*rest.buruk_akhir
                penyebut += rest.buruk_akhir
            elif bilangan >= parm['bagus_b']:        #Pengecekan jika berada dari titik
bagus_b ke atas
                pembilang += bilangan*rest.bagus_akhir
                penyebut += rest.bagus_akhir
            else:                                    #Pengecekan jika berada di antara /
transisi
                nilai_asli_buruk = -(bilangan - parm['buruk_d']) / (parm['buruk_d'] -
parm['buruk_c'])
                nilai_asli_bagus = (bilangan - parm['bagus_a']) / (parm['bagus_b'] -
parm['bagus_a'])

                nilai_terkecil_buruk = min([nilai_asli_buruk, rest.buruk_akhir])
                nilai_terkecil_bagus = min([nilai_asli_bagus, rest.bagus_akhir])

                nilai_max = max([nilai_terkecil_buruk, nilai_terkecil_bagus])

                pembilang += bilangan*nilai_max
                penyebut += nilai_max

        rest.nilai_akhir = pembilang // penyebut

        return rest

# Melakukan defuzzifikasi terhadap data_restoran
for i in range(len(restoran_data)):
    restoran_data[i] = defuzzifikasi(20, restoran_data[i])

```

3.5. Output/File Export

Bagian ini merupakan bagian terakhir dari algoritma yang harus kami buat. Setelah mendapatkan *crisp output* kita bisa memeringkatkan 10 restoran yang memiliki nilai tertinggi dengan menggunakan fungsi NilaiAkhirSort(). Setelah didapatkan 10 restoran terakhir kami merubah dari isi list menjadi *dataframe* lalu mengoutputkan ke dalam file *excel* bernama 'peringkat.xls'

```

def NilaiAkhirSort(restoran_temp: [restoran()]):
    restoran_temp = sorted(restoran_temp, key=lambda x: x.nilai_akhir, reverse= 1)
    return restoran_temp

# Melakukan sorting terhadap restoran_data
restoran_data = NilaiAkhirSort(restoran_data)

```

```

# Memasukkan 10 restoran dengan nilai tertinggi ke dalam List data_akhir
data_akhir = []
for i in range(10):
    data_temp = []
    data_temp.append(restoran_data[i].id_restoran)
    data_temp.append(restoran_data[i].nilai_akhir)
    data_temp.append(restoran_data[i].pelayanan)
    data_temp.append(restoran_data[i].makanan)
    data_akhir.append(data_temp)

# Memasukkan List data_akhir ke dalam dataframe df_akhir
df_akhir = pd.DataFrame(data_akhir, columns=['id', 'nilai_akhir', 'pelayanan', 'makanan'])

# Membuat excel dari dataframe df_akhir
df_akhir.to_excel('../data/processed/peringkat.xls', encoding='xlwt')

df_akhir

```

4. KESIMPULAN

Pada tahap fuzzifikasi dapat dilihat pada **Gambar 2.1** dan **Gambar 2.2** disini kami melakukan inputan yang dimana kualitas makanan di batasi dengan rating 0-10 dan kualitas pelayanan dibatasi dengan rating 0-100, pada tahap inferensi bisa dilihat pada **Gambar 2.3** akan diterima inputan dari fuzzifikasi lalu akan dibandingkan dari kualitas makanan dan kualitas pelayan sehingga kita dapat menyimpulkan bagus atau buruk. Selanjutnya pada tahap terakhir atau defuzzifikasi bisa dilihat pada **Gambar 2.4** dan **Gambar 2.5** setelah kami mendapatkan hasil dari inferensi atau *Fuzzy Output* kami merubah nilai tersebut menjadi nilai yang jelas dengan skala yang sudah di tentukan yakni 1-100.

Dari hasil algoritma atau program yang kami buat sudah memenuhi syarat yang dimana pada tahap pertama adalah Fuzzifikasi lalu tahap kedua adalah Inferensi dan tahap terakhir adalah Defuzzifikasi. Kesimpulan yang dapat diambil adalah dengan menggunakan *fuzzy logic* masalah terhadap suatu data yang bersifat kualitatif dapat kita ubah menjadi penilaian yang pasti. Dalam kasus ini, kami dapat merubah berdasarkan inputan yang bersifat kualitatif, yaitu penilaian makanan dan pelayanan, hingga menjadi satu penilaian yang pasti terhadap restoran tersebut.

Lampiran

Video Penjelasan oleh Kaenova Mahendra Auditama: <https://youtu.be/uJfs4-bQxh0>

Video Penjelasan oleh Adhe Akram Azhari: <https://youtu.be/2ax07TOoJ94>

Video Penjelasan oleh Elita Aurora Az Zahra: <https://youtu.be/6Rq5L04l9MY>

Notebooks Program: https://kaenova.github.io/AI_Tupro2/

Source Code Program: https://github.com/kaenova/AI_Tupro2

Referensi

Novák, V.; Perfilieva, I.; Močkoř, J. (1999). Mathematical principles of fuzzy logic. Dordrecht: Kluwer Academic.

"Fuzzy Logic | by Fakultas Informatika, School of Computing, Telkom University by Suyanto"