

LAPORAN OBSERVASI FUZZY LOGIC

ditujukan untuk memenuhi tugas mata kuliah Sistem Cerdas

Dosen Pengampu Azka Khoirunnisa



oleh :

Pratama Azmi Atmajaya (1303180096)

Ruben (1303184084)

Rama Ahmadi Ariayudha (1303184036)

IT - 42 - 01

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

BANDUNG

2021

KATA PENGANTAR

Alhamdulillah, segala puja dan puji senantiasa kita ucapkan atas berkaat dan rahmatnya sehingga kita dapat menyelesaikan tugas yang diberikan kepada kami. Shalawat dan salam kita curahkan kepada baginda nabi kita Muhammad SAW. semoga kita selalu mendapatkan syafaat Beliau di Yaumul Mahsyar.

Adapun tujuan utama dalam penulisan laporan ini, yaitu untuk memenuhi salah satu tugas dari mata kuliah Sistem Cerdas dengan judul laporan “Fuzzy Logic”.

Kami ucapkan terima kasih kepada Ibu Azka Khoirunnisa selaku dosen pembimbing dan kepada semua pihak yang sudah membantu dalam penulisan laporan ini dari awal hingga selesai.

Kami menyadari bahwa Laporan kami masih banyak kekurangan. Oleh sebab itu, kami mengharapkan kritik dan saran agar terciptanya laporan yang lebih baik di masa yang akan datang. Semoga laporan yang kami buat dapat bermanfaat dan memenuhi nilai tugas mata kuliah Sistem Cerdas.

DAFTAR ISI

Kata Pengantar	1
Daftar Isi	2
Bab I Pendahuluan	3
A. Latar Belakang	3
B. Analisa Masalah	3
C. Rumusan Masalah	3
D. Tujuan Observasi	3
E. Manfaat Observasi	3
Bab II Hasil Observasi	4
A. Pemecahan Masalah	4
B. Analisis Masalah dan Penjelasan Teknik Strategi	4
C. Implementasi	15
D. Proses yang Dibangun	15
E. Output Program	22
Bab III Penutupan	23
A. Kesimpulan	23
B. Daftar Pustaka	23
C. Tautan Google Drive	23

BAB I

PENDAHULUAN

A. Latar Belakang

Fuzzy logic adalah sebuah logika yang bernilai benar atau salah. Pada teori *fuzzy logic*, nilai bisa berupa benar atau salah secara bersamaan. Ada beberapa keberadaan dan kesalahan pada sesuatu yang bergantung pada bobot anggota yang dimilikinya. *Fuzzy logic* mempunyai derajat keanggotaan dalam rentang 0 sampai 1. Terdapat perbedaan dengan *digital logic* yang hanya memiliki dua nilai 1 atau 0. *Fuzzy logic* berguna untuk menerjemahkan besaran yang diekspresikan dengan bahasa linguistik, contoh penjualan alat elektronik dengan kualitas buruk, baik dan sangat baik. *Fuzzy logic* juga menunjukkan sampai mana nilai itu bernilai benar dan salah. *Classic logic* memiliki acuan yang berbeda, berupa *crisp/tegas*, nilai yang hanya mempunyai dua kemungkinan yaitu berupa keanggotaan himpunan atau tidak. Derajat anggota 0 artinya nilai bukan termasuk anggota himpunan dan 1 berarti nilai tersebut merupakan anggota himpunan.

B. Analisa Masalah

Kualitas suatu makanan dan minuman menjadi acuan utama dalam sebuah usaha, bahkan dalam pelayanan juga terdapat kualitas untuk menjamin baik buruknya. Ada kendala ketika kualitas itu menurun bahkan bisa jadi lebih parah, yaitu menyebabkan kerugian pada suatu perusahaan karena terlalu berlebihan. Fungsi dari pelatihan dan juga pengalaman diperlukan untuk mengatur agar kualitas tetap stabil atau bahkan meningkat.

C. Rumusan Masalah

1. Bagaimana mengimplementasikan metode *fuzzy logic* sebagai pengambil keputusan dalam mengukur sebuah kualitas pada restoran?
2. Bagaimana menguji metode *fuzzy* sebagai pendukung keputusan dalam mengukur sebuah kualitas pada restoran?

D. Tujuan Observasi

1. Mengimplementasikan metode *fuzzy logic* sebagai pengambil keputusan dalam mengukur sebuah kualitas pada restoran.
2. Menguji metode *fuzzy logic* sebagai pendukung keputusan dalam mengukur sebuah kualitas pada restoran.

E. Manfaat Observasi

1. Mengetahui cara implementasi dan parameter yang digunakan dalam *fuzzy logic* terhadap peristiwa nyata, serta manfaat dan kelebihan yang diperoleh dari penerapan pengolahan data dengan rentang nilai yang menyerupai suatu bisnis.

2. Mengetahui derajat kualitas dari sebuah bisnis restoran berdasarkan penilaian dari pelanggan terhadap aspek yang terlibat, sehingga dapat membantu dalam mengambil keputusan untuk menentukan letak kelebihan dan kekurangan, serta aspek yang paling dominan mempengaruhi kualitas bisnis.

BAB II

HASIL OBSERVASI

A. Pemecahan Masalah

Terdapat 100 himpunan data restoran yang terdapat pada kota A dan memiliki 3 atribut yaitu kualitas makanan (bilangan int 1 - 10) kualitas minuman (bilangan int 1 - 10) dan kualitas pelayanan (bilangan int 1 - 100) . Dengan menggunakan fuzzy logic dapat ditentukan 10 restoran terbaik di kota A.

B. Analisis masalah dan Penjelasan Teknik Strategi

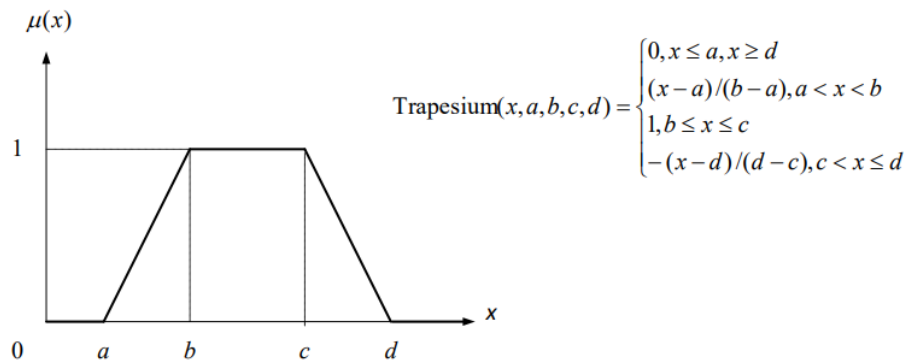
1. Jumlah dan Nama Linguistik setiap input

Pada setiap input, input minuman telah dibagi menjadi 3 nama linguistik yaitu buruk, sedang, baik. Sedangkan untuk nama linguistik makanan telah dibagi menjadi 3 yaitu buruk, sedang, baik. Sedangkan untuk nama linguistik pada input pelayanan yaitu buruk, baik, dan sangat baik. Kami memilih menggunakan 3 nama linguistik untuk setiap inputnya karena relatif sederhana namun efektif dan dapat menghasilkan nilai beragam.

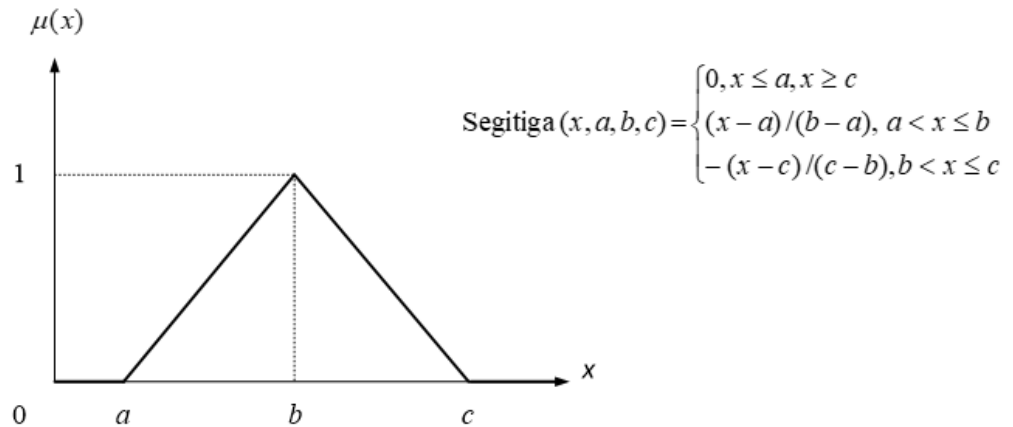
2. Bentuk dan Batas Fungsi Keanggotaan Input

Pada bentuk dan fungsi keanggotaan kami menggunakan fungsi trapesium dan segitiga .

Fungsi Trapesium :

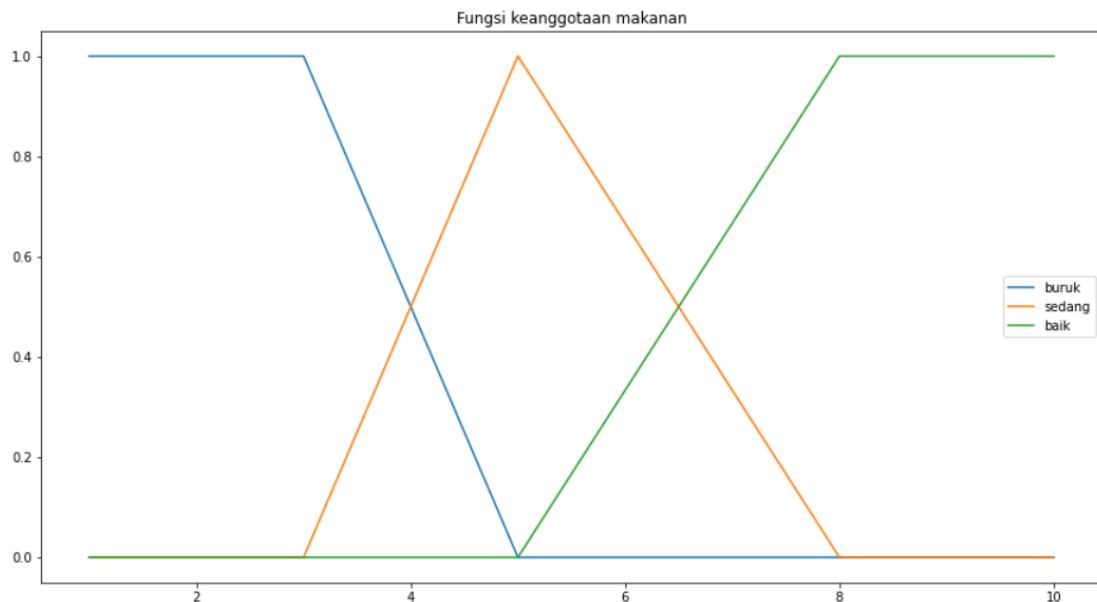


Fungsi segitiga :



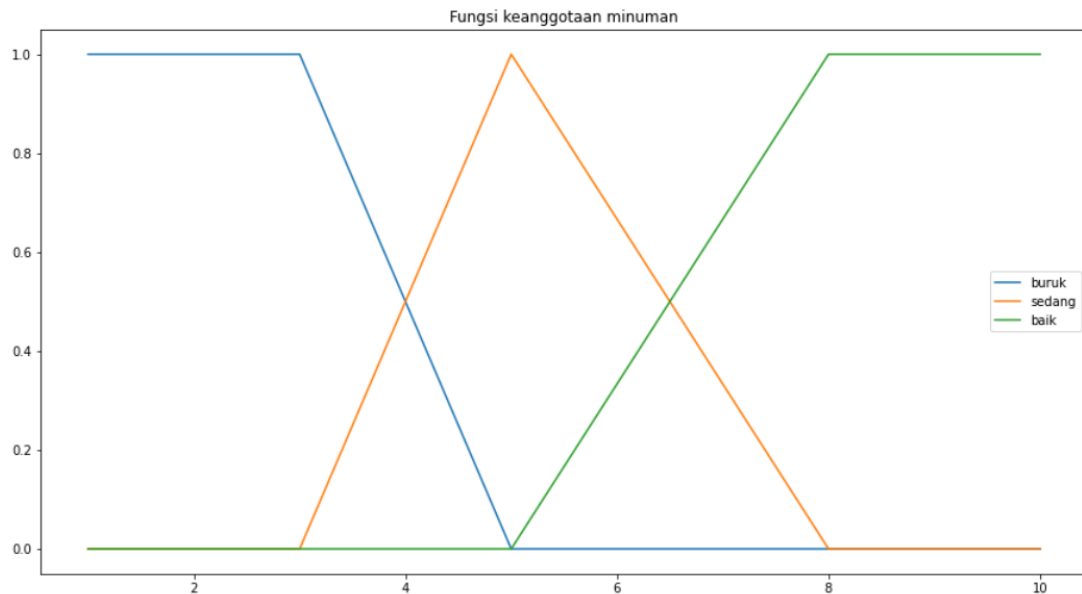
a. Makanan

Berikut adalah visualisasi fungsi keanggotaan makanan , dengan axis x merepresentasikan range nilai x , dan y merepresentasikan derajat keanggotaan dari makanan. Dengan beberapa kategori, yaitu buruk, sedang, dan baik.



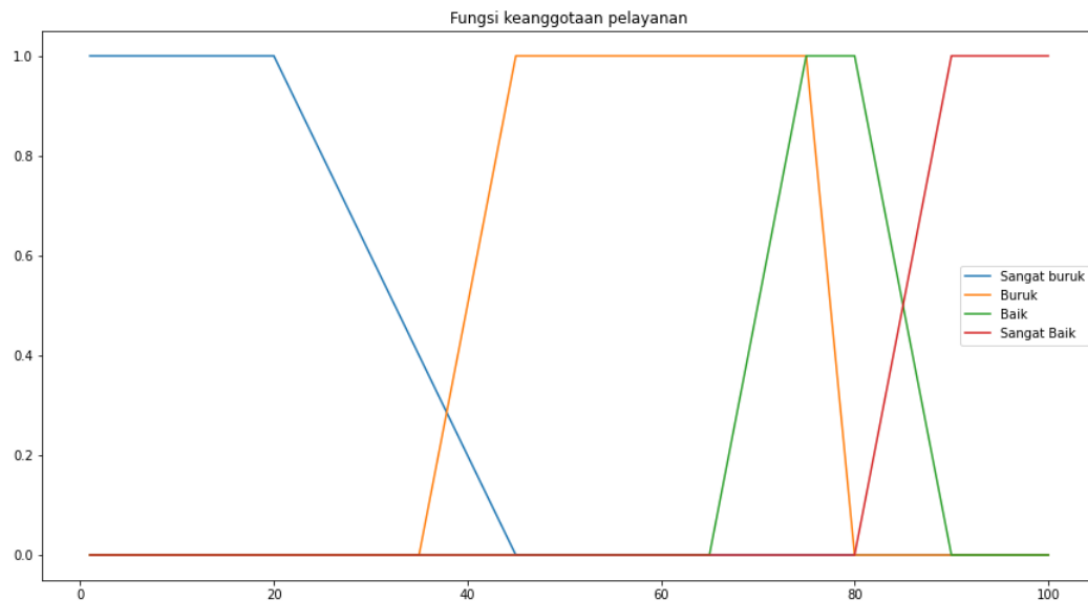
b. Minuman

Berikut adalah visualisasi fungsi keanggotaan minuman, dengan axis x merepresentasikan range nilai x , dan y merepresentasikan derajat keanggotaan dari minuman . Dengan beberapa kategori, yaitu buruk, sedang, dan baik.



c. Pelayanan

Berikut adalah visualisasi fungsi keanggotaan pelayanan, dengan axis x merepresentasikan range nilai x , dan y merepresentasikan derajat keanggotaan dari pelayanan . Dengan beberapa kategori , yaitu Sangat Buruk , Buruk , Baik, dan Sangat Baik.



3. Aturan Inferensi

Memiliki tiga input yaitu:

1. Makanan
2. Minuman
3. Pelayanan

Untuk merepresentasikan kualitas setiap input (skor) dan nilai kualitas (NK) atau penilaian akhir, digunakan rentang nilai integer sebagai berikut:

1. 0
Buruk untuk makanan dan minuman, sangat buruk untuk pelayanan, menghasilkan sangat rendah untuk NK
2. 1
Sedang untuk makanan dan minuman, buruk untuk pelayanan, menghasilkan nilai rendah untuk NK
3. 2
Baik untuk makanan dan minuman, baik untuk pelayanan, menghasilkan nilai tinggi untuk NK
4. 3
Sangat baik pada pelayanan, menghasilkan nilai sangat tinggi untuk NK

Input tersebut diolah melalui serangkaian aturan untuk dikategorikan ke NK sebagai berikut:

Makanan\ Minuman		Pelayanan			
		Sangat Buruk	Buruk	Baik	Sangat baik
Makanan \ Minuman	Buruk \ Buruk	Sangat Rendah	Sangat Rendah	Sangat Rendah	Sangat Rendah
	Sedang \ Buruk	Sangat Rendah	Sangat Rendah	Sangat Rendah	Rendah
	Baik \ Buruk	Sangat Rendah	Rendah	Rendah	Tinggi
	Buruk \ Baik	Sangat Rendah	Rendah	Rendah	Tinggi
	Buruk \ Sedang	Sangat Rendah	Sangat Rendah	Sangat Rendah	Rendah
	Sedang \ Sedang	Rendah	Rendah	Tinggi	Tinggi
	Sedang \ Baik	Rendah	Rendah	Tinggi	Sangat Tinggi
	Baik \ Sedang	Rendah	Rendah	Tinggi	Sangat Tinggi
	Baik \ Baik	Rendah	Tinggi	Sangat Tinggi	Sangat Tinggi

- 1) IF Makanan = Buruk AND Minuman = Buruk AND Pelayanan = Sangat Buruk THEN NK = Sangat Rendah
- 2) IF Makanan = Buruk AND Minuman = Buruk AND Pelayanan = Buruk THEN NK = Sangat Rendah
- 3) IF Makanan = Buruk AND Minuman = Buruk AND Pelayanan = Baik THEN NK = Sangat Rendah
- 4) IF Makanan = Buruk AND Minuman = Buruk AND Pelayanan = Sangat Baik THEN NK = Sangat Rendah
- 5) IF Makanan = Sedang AND Minuman = Buruk AND Pelayanan = Sangat Buruk THEN NK = Sangat Rendah
- 6) IF Makanan = Sedang AND Minuman = Buruk AND Pelayanan = Buruk THEN NK = Sangat Rendah
- 7) IF Makanan = Sedang AND Minuman = Buruk AND Pelayanan = Baik THEN NK = Sangat Rendah
- 8) IF Makanan = Sedang AND Minuman = Buruk AND Pelayanan = Sangat Baik THEN NK = Rendah
- 9) IF Makanan = Baik AND Minuman = Buruk AND Pelayanan = Sangat Buruk THEN NK = Sangat Rendah
- 10) IF Makanan = Baik AND Minuman = Buruk AND Pelayanan = Buruk THEN NK = Rendah
- 11) IF Makanan = Baik AND Minuman = Buruk AND Pelayanan = Baik THEN NK = Rendah

- 12) IF Makanan = Baik AND Minuman = Buruk AND Pelayanan = Sangat Baik THEN NK = Tinggi
- 13) IF Makanan = Buruk AND Minuman = Baik AND Pelayanan = Sangat Buruk THEN NK = Sangat Rendah
- 14) IF Makanan = Buruk AND Minuman = Baik AND Pelayanan = Buruk THEN NK = Rendah
- 15) IF Makanan = Buruk AND Minuman = Baik AND Pelayanan = Baik THEN NK = Rendah
- 16) IF Makanan = Buruk AND Minuman = Baik AND Pelayanan = Sangat Baik THEN NK = Tinggi
- 17) IF Makanan = Buruk AND Minuman = Sedang AND Pelayanan = Sangat Buruk THEN NK = Sangat Rendah
- 18) IF Makanan = Buruk AND Minuman = Sedang AND Pelayanan = Buruk THEN NK = Sangat Rendah
- 19) IF Makanan = Buruk AND Minuman = Sedang AND Pelayanan = Baik THEN NK = Sangat Rendah
- 20) IF Makanan = Buruk AND Minuman = Sedang AND Pelayanan = Sangat Baik THEN NK = Rendah
- 21) IF Makanan = Sedang AND Minuman = Sedang AND Pelayanan = Sangat Buruk THEN NK = Rendah
- 22) IF Makanan = Sedang AND Minuman = Sedang AND Pelayanan = Buruk THEN NK = Rendah
- 23) IF Makanan = Sedang AND Minuman = Sedang AND Pelayanan = Baik THEN NK = Tinggi
- 24) IF Makanan = Sedang AND Minuman = Sedang AND Pelayanan = Sangat Baik THEN NK = Tinggi
- 25) IF Makanan = Sedang AND Minuman = Baik AND Pelayanan = Sangat Buruk Then NK = Rendah
- 26) IF Makanan = Sedang AND Minuman = Baik AND Pelayanan = Buruk THEN NK = Rendah

- 27) IF Makanan = Sedang AND Minuman = Baik AND Pelayanan = Baik
THEN NK = Tinggi
- 28) IF Makanan = Sedang AND Minuman = Baik AND Pelayanan = Sangat
Baik THEN NK = Sangat Tinggi
- 29) IF Makanan = Baik AND Minuman = Sedang AND Pelayanan = Sangat
Buruk THEN NK = Rendah
- 30) IF Makanan = Baik AND Minuman = Sedang AND Pelayanan = Buruk
THEN NK = Rendah
- 31) IF Makanan = Baik AND Minuman = Sedang AND Pelayanan = Baik
THEN NK = Tinggi
- 32) IF Makanan = Baik AND Minuman = Sedang AND Pelayanan = Sangat
Baik THEN NK = Sangat Tinggi
- 33) IF Makanan = Baik AND Minuman = Baik AND Pelayanan = Sangat
Buruk THEN NK = Rendah
- 34) IF Makanan = Baik AND Minuman = Baik AND Pelayanan = Buruk
Then NK = Tinggi
- 35) IF Makanan = Baik AND Minuman = Baik AND Pelayanan = Baik
Then NK = Sangat Tinggi
- 36) IF Makanan = Baik AND Minuman = Baik AND Pelayanan = Sangat
Baik THEN NK = Sangat Tinggi

Rules diatas dapat dikonversi menjadi kodingan sebagai berikut :

```
def RulesNgab(makanan, minuman, pelayanan):
    if (makanan == 0 and minuman == 0 and pelayanan == 0):
        return 0
    elif (makanan == 0 and minuman == 0 and pelayanan == 1):
        return 0
    elif(makanan == 0 and minuman == 0 and pelayanan == 2 ):
        return 1
    elif(makanan == 0 and minuman == 0 and pelayanan == 3):
        return 1

    elif(makanan == 1 and minuman == 0 and pelayanan == 0):
        return 0
```

```

elif(makanan == 1 and minuman == 0 and pelayanan == 1):
    return 0
elif(makanan == 1 and minuman == 0 and pelayanan == 2):
    return 0
elif(makanan == 1 and minuman == 0 and pelayanan == 3):
    return 1
#baik / buruk
elif(makanan == 2 and minuman == 0 and pelayanan == 0):
    return 0
elif(makanan == 2 and minuman == 0 and pelayanan == 1):
    return 1
elif(makanan == 2 and minuman == 0 and pelayanan == 2):
    return 1
elif(makanan == 2 and minuman == 0 and pelayanan == 3):
    return 2

# buruk/ baik
elif(makanan == 0 and minuman == 2 and pelayanan == 0):
    return 0
elif(makanan == 0 and minuman == 2 and pelayanan == 1):
    return 1
elif(makanan == 0 and minuman == 2 and pelayanan == 2):
    return 1
elif(makanan == 0 and minuman == 2 and pelayanan == 3):
    return 2

#Buruk/sedang
elif(makanan == 0 and minuman == 1 and pelayanan == 0):
    return 0
elif(makanan == 0 and minuman == 1 and pelayanan == 1):
    return 0
elif(makanan == 0 and minuman == 1 and pelayanan == 2):
    return 0
elif(makanan == 0 and minuman == 1 and pelayanan == 3):
    return 1

#Sedang / Sedang
elif (makanan == 1 and minuman == 1 and pelayanan == 0):
    return 1

```

```

elif (makanan == 1 and minuman == 1 and pelayanan == 1):
    return 1
elif (makanan == 1 and minuman == 1 and pelayanan == 2):
    return 2
elif (makanan == 1 and minuman == 1 and pelayanan == 3):
    return 2
#Sedang / Baik
elif(makanan == 1 and minuman == 2 and pelayanan == 0):
    return 1
elif(makanan == 1 and minuman == 2 and pelayanan == 1):
    return 1
elif(makanan == 1 and minuman == 2 and pelayanan == 2):
    return 2
elif(makanan == 1 and minuman == 2 and pelayanan == 3):
    return 3

#Baik / Sedang
elif(makanan == 2 and minuman == 1 and pelayanan == 0):
    return 1
elif(makanan == 2 and minuman == 1 and pelayanan == 1):
    return 1
elif(makanan == 2 and minuman == 1 and pelayanan == 2):
    return 2
elif(makanan == 2 and minuman == 1 and pelayanan == 3):
    return 3
#baik / baik
elif(makanan == 2 and minuman == 2 and pelayanan == 0):
    return 1
elif(makanan == 2 and minuman == 2 and pelayanan == 1):
    return 2
elif(makanan == 2 and minuman == 2 and pelayanan == 2):
    return 3
elif(makanan == 2 and minuman == 2 and pelayanan == 3):
    return 3

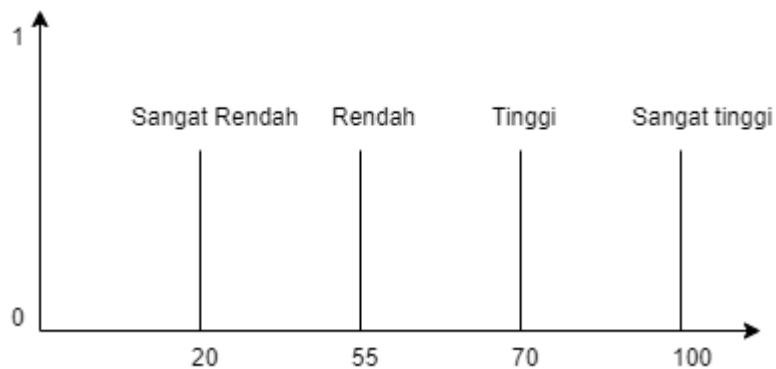
```

4. Metode Defuzzifikasi

Tujuan dari defuzzifikasi adalah menghasilkan nilai variabel solusi yang diinginkan dari suatu daerah konsekuen fuzzy, untuk metode yang digunakan adalah metode sugeno.

Berikut adalah fungsi Defuzzifikasi yang dibangun :

```
def defuzzySugeno(sgtRendah, rendah, tinggi, sgtTinggi):  
    totalValue = 0  
    pembagi = 0  
  
    if (sgtRendah != 0):  
        totalValue += sgtRendah * 20  
        pembagi += sgtRendah  
    if (rendah != 0):  
        totalValue += rendah * 55  
        pembagi += rendah  
    if (tinggi != 0):  
        totalValue += tinggi * 70  
        pembagi += tinggi  
    if (sgtTinggi != 0):  
        totalValue += sgtTinggi * 100  
        pembagi += sgtTinggi  
  
    return totalValue / pembagi
```

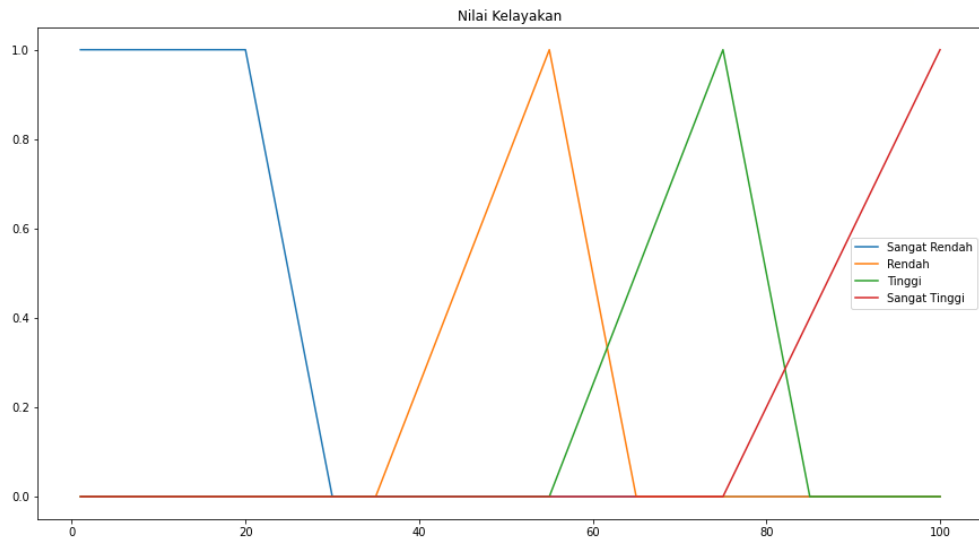


5. Bentuk dan Batas Fungsi Keanggotaan Output (sesuai metode defuzzifikasi)

Berikut adalah batas fungsi keanggotaan untuk Nilai kelayakan .

```
class Kelayakan():  
  
    def __init__(self,x):  
        self.valueOfX = x  
    def kelayakanSangatRendah(self):  
        return fungsiTrapesium(self.valueOfX, -0.001, 0, 20 ,30)  
    def kelayakanRendah(self):  
        return fungsiSegitiga(self.valueOfX, 20, 55, 65)  
    def kelayakanTinggi(self):  
        return fungsiSegitiga(self.valueOfX, 55, 75, 85)  
    def kelayakanSangatTinggi(self):  
        return fungsiTrapesium(self.valueOfX, 75, 100, 100.001, 100.001)
```

```
plt.figure(figsize=(15,8))  
plt.plot(kelayakan,[Kelayakan(x).kelayakanSangatRendah() for x in kelayakan],label="Sangat Rendah")  
plt.plot(kelayakan,[Kelayakan(x).kelayakanRendah() for x in kelayakan],label="Rendah")  
plt.plot(kelayakan,[Kelayakan(x).kelayakanTinggi() for x in kelayakan],label="Tinggi")  
plt.plot(kelayakan,[Kelayakan(x).kelayakanSangatTinggi() for x in kelayakan],label="Sangat Tinggi")  
plt.title("Nilai Kelayakan")  
plt.legend()
```



C. Implementasi

1. Bahasa pemrograman

Pada sistem fuzzy logic yang dibuat, disini kami menggunakan bahasa pemrograman *python ver 3.x*. Bahasa pemrograman *Python* merupakan bahasa yang mudah dipelajari dan cocok untuk scientific computing sehingga, sangat mudah untuk mengaplikasikan memiliki struktur yang sederhana sehingga mudah diaplikasikan.

2. Library

Library yang digunakan adalah *pandas* untuk manipulasi data dan membuat data, sedangkan untuk *numpy* manipulasi array, Selain itu, *matplotlib* dan *seaborn* digunakan untuk plotting luaran, dan fungsi keanggotaan .

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

D. Proses yang Dibangun

1. Membaca file

Hal yang pertama adalah membentuk dataset dengan state tertentu, karena dataset telah terbentuk sebelumnya , maka kami menggunakan `!gdown --id idfile` , dengan begitu dataset didapatkan dengan mudah. Selanjutnya adalah membaca dataset tersebut.

```
# Eksekusi Cell ini jika ingin generate new dataset
!gdown --id 11aRsQOlcon_MwuCfs6Wtf8x4vqpeuQI8
np.random.seed(100)
def randomizeMinuman(totalRow):
    return np.random.randint(low = 1, high = 11,
                             size= totalRow)
def randomizeMakanan(totalRow):
    return np.random.randint(low = 1, high = 11,
                             size = totalRow)
def randomizePelayanan(totalRow):
    return np.random.randint(low = 1, high = 101,
                             size = totalRow)
def createDataset():
    df = pd.DataFrame({"id":range(0,100)})
    df["NilaiMinuman"] = randomizeMinuman(len(df))
    df["NilaiMakanan"] = randomizeMakanan(len(df))
    df["NilaiPelayanan"] = randomizePelayanan(len(df))
    return df
#createDataset().to_csv("Dataset.csv",index=False)
df = pd.read_csv("Dataset.csv")
```


2. Fungsi keanggotaan

2.1 Fungsi Keanggotaan makanan

Berikut adalah representasi kelas untuk Makanan, pada kelas ini terdapat beberapa method untuk memberikan nilai fungsi keanggotaan untuk masing masing kategori makanan .

```
class Makanan():
    def __init__(self, x):
        self.valueOfX = x
    def makananEnak(self):
        return fungsiTrapeسيوم(self.valueOfX, 5, 8, 10, 10.001)
    def makananSedang(self):
        return fungsiSegitiga(self.valueOfX, 3, 5, 8)
    def makananNggaEnak(self):
        return fungsiTrapeسيوم(self.valueOfX, -0.001, 0 ,3, 5)
```

2.2 Fungsi Keanggotaan Minuman

Berikut adalah representasi kelas untuk Minuman, pada kelas ini terdapat beberapa method untuk memberikan nilai fungsi keanggotaan untuk masing masing kategori minuman .

```
class Minuman():
    def __init__(self,x):
        self.valueOfX = x
    def minumanEnak(self):
        return fungsiTrapeسيوم(self.valueOfX, 5, 8, 10, 10.001)
    def minumanSedang(self):
        return fungsiSegitiga(self.valueOfX, 3, 5, 8)
    def minumanNggaEnak(self):
        return fungsiTrapeسيوم(self.valueOfX, -0.001, 0 , 3, 5)
```

2.2 Fungsi Keanggotaan Pelayanan

Berikut adalah representasi kelas untuk Pelayanan, pada kelas ini terdapat beberapa method untuk memberikan nilai fungsi keanggotaan untuk masing masing kategori pelayanan.

```

class Pelayanan():
    def __init__(self, x):
        self.valueOfX = x
    def pelayananSangatBaik(self):
        return fungsiTrapesium(self.valueOfX, 75, 90, 100, 100.001)
    def pelayananBaik(self):
        return fungsiTrapesium(self.valueOfX, 45, 75, 80, 90)
    def pelayananBuruk(self):
        return fungsiTrapesium(self.valueOfX, 20, 45, 75, 80)
    def pelayananSangatBuruk(self):
        return fungsiTrapesium(self.valueOfX, -0.001, 0, 20, 45)

```

2.3 Fungsi Plot Keanggotaan untuk Penghasilan Function

Untuk melakukan visualisasi terhadap fungsi keanggotaan kami menggunakan matplotlib sebagai tools atau library untuk menghasilkan plotting agar mudah untuk memahami fungsi keanggotaan yang dihasilkan.

Makanan :

Untuk plot fungsi keanggotaan pada Makanan, kami membuat visualisasi menggunakan library matplotlib dan mengambil nilai makanan dari 1 sampai 10. Berikut adalah kodingan yang dibuat untuk menghasilkan visualisasinya.

```

makanan = np.arange(1, 11,1)
plt.figure(figsize=(15,8))
plt.plot(makanan,[Makanan(x).makananNggaEnak() for x in makanan],label = "buruk")
plt.plot(makanan, [Makanan(x).makananSedang() for x in makanan],label = "sedang")
plt.plot(makanan,[Makanan(x).makananEnak() for x in makanan],label="baik")
plt.title("Fungsi keanggotaan makanan")
plt.legend()

```

Minuman :

Untuk plot fungsi keanggotaan pada Makanan, kami membuat visualisasi menggunakan library matplotlib dan mengambil nilai makanan dari 1 sampai 10. Berikut adalah kodingan yang dibuat untuk menghasilkan visualisasinya.

```

minuman = np.arange(1, 11,1)
plt.figure(figsize=(15,8))
plt.plot(minuman,[Minuman(x).minumanNggaEnak() for x in minuman],label = "buruk")
plt.plot(minuman, [Minuman(x).minumanSedang() for x in minuman],label = "sedang")
plt.plot(minuman,[Minuman(x).minumanEnak() for x in minuman],label="baik")
plt.title("Fungsi keanggotaan minuman")
plt.legend()

```

Pelayanan :

Untuk plot fungsi keanggotaan pada Makanan, kami membuat visualisasi menggunakan library matplotlib dan mengambil nilai makanan dari 1 sampai 10. Berikut adalah kodingan yang dibuat untuk menghasilkan visualisasinya.

```

plt.figure(figsize=(15,8))
pelayanan= np.arange(1, 101,1)
plt.plot(pelayanan,[Pelayanan(x).pelayananSangatBuruk() for x in pelayanan],label="Sangat buruk")
plt.plot(pelayanan,[Pelayanan(x).pelayananBuruk() for x in pelayanan],label = "Buruk")
plt.plot(pelayanan,[Pelayanan(x).pelayananBaik() for x in pelayanan],label="Baik")
plt.plot(pelayanan,[Pelayanan(x).pelayananSangatBaik() for x in pelayanan],label="Sangat Baik")
plt.title("Fungsi keanggotaan pelayanan")
plt.legend()

```

2. Fuzzifikasi

Pada bagian fuzzifikasi, digunakan fungsi keanggotaan yang telah dibentuk, berikut adalah tahapan tahapan yang dilakukan untuk fuzzifikasi.

```

def fuzzification(makananScore , minumanScore , pelayananScore):

    MakananObj = Makanan(makananScore)
    MinumanObj = Minuman(minumanScore)
    PelayananObj = Pelayanan(pelayananScore)

    makananArr = np.array([round(m,2) for m in [MakananObj.makananNggaEnak(),MakananObj.makananSedang(),
                                                MakananObj.makananEnak()]])
    minumanArr = np.array([round(m,2) for m in [MinumanObj.minumanNggaEnak(),MinumanObj.minumanSedang(),
                                                MinumanObj.minumanEnak()]])
    pelayananArr = np.array([round(p,2) for p in [PelayananObj.pelayananSangatBuruk(),PelayananObj.pelayananBuruk(),
                                                PelayananObj.pelayananBaik(),PelayananObj.pelayananSangatBaik()]])

    hasilInference = inference(makananArr,minumanArr,pelayananArr)
    print("Hasil Inference : ")
    for i in range(len(hasilInference)):
        print(f"Nilai Kelayakan : {mapKualitas(i)} -> {hasilInference[i]}")

    return defuzzySugeno(*hasilInference)

```

3. Inferensi

Untuk melakukan inferensi , kami membuat sebuah logic untuk menyesuaikan kepada luaran yang minta, berikut adalah hasil tahap inferensi

yang dilakukan .

```
def inference(makananArrScore,minumanArrScore,pelayananArrScore):
    result = np.zeros(shape=(5),dtype=np.float)

    zerosMinuman = np.where(minumanArrScore == 0)[0] # Get The content ,i.e Index zeros Minuman
    zerosMakanan = np.where(makananArrScore == 0)[0]
    zerosPelayanan = np.where(pelayananArrScore == 0)[0]

    if(np.size(zerosPelayanan) <= 2):
        idxPelayanan = (-pelayananArrScore).argsort()[0]

        if (np.size(zerosMakanan) <= 1):
            idxMakanan = (-makananArrScore).argsort()[0:2]

            if(np.size(zerosMinuman) <= 1):
                idxMinuman = (-minumanArrScore).argsort()[0:2]
                result = gradeResult(idxMakanan, idxMinuman, idxPelayanan,makananArrScore,
                                     minumanArrScore,pelayananArrScore)

            elif(np.size(zerosMinuman) > 1):
                idxMinuman = (-minumanArrScore).argsort()[0:1]
                result = gradeResult(idxMakanan, idxMinuman,
                                     idxPelayanan,makananArrScore,
                                     minumanArrScore,pelayananArrScore)

        elif (np.size(zerosMakanan) > 1 ):
            idxMakanan = (-makananArrScore).argsort()[0:1]
```

0.4 selected

```

elif (np.size(zerosMakanan) > 1 ):
    idxMakanan = (-makananArrScore).argsort()[1:]

    if(np.size(zerosMinuman) <= 1):
        idxMinuman = (-minumanArrScore).argsort()[2]
        result = gradeResult(idxMakanan, idxMinuman, idxPelayanan,makananArrScore,
                               minumanArrScore,pelayananArrScore)

    elif(np.size(zerosMinuman) > 1):
        idxMinuman = (-minumanArrScore).argsort()[1]
        result = gradeResult(idxMakanan, idxMinuman,
                               idxPelayanan,makananArrScore,
                               minumanArrScore,pelayananArrScore)

elif(np.size(zerosPelayanan) >2):
    idxPelayanan = (-pelayananArrScore).argsort()[1:]

    if (np.size(zerosMakanan) <= 1):
        idxMakanan = (-makananArrScore).argsort()[2]

    if(np.size(zerosMinuman) <= 1):
        idxMinuman = (-minumanArrScore).argsort()[2]
        result = gradeResult(idxMakanan, idxMinuman, idxPelayanan,makananArrScore,
                               minumanArrScore,pelayananArrScore)

    elif(np.size(zerosMinuman) > 1):
        idxMinuman = (-minumanArrScore).argsort()[1]
        result = gradeResult(idxMakanan, idxMinuman,
                               idxPelayanan,makananArrScore,
                               minumanArrScore,pelayananArrScore)

```

```

elif (np.size(zerosMakanan) > 1 ):
    idxMakanan = (-makananArrScore).argsort()[1:]

    if(np.size(zerosMinuman) <= 1):
        idxMinuman = (-minumanArrScore).argsort()[2]
        result = gradeResult(idxMakanan, idxMinuman, idxPelayanan,makananArrScore,
                               minumanArrScore,pelayananArrScore)

    elif(np.size(zerosMinuman) > 1):
        idxMinuman = (-minumanArrScore).argsort()[1]
        result = gradeResult(idxMakanan, idxMinuman,
                               idxPelayanan,makananArrScore,
                               minumanArrScore,pelayananArrScore)

return result

```

4. Defuzzifikasi

Untuk mengubah hasil daripada inferensi tadi menjadi crisp value , maka perlu dilakukan defuzzifikasi , kami menggunakan metode defuzzifikasi sugeno. Berikut adalah kodingan untuk metode yang kami pakai

4.1 Sugeno

```
def defuzzySugeno(sgtRendah, rendah, tinggi, sgtTinggi):  
    totalValue = 0  
    pembagi = 0  
  
    if (sgtRendah != 0):  
        totalValue += sgtRendah * 20  
        pembagi += sgtRendah  
    if (rendah != 0):  
        totalValue += rendah * 55  
        pembagi += rendah  
    if (tinggi != 0):  
        totalValue += tinggi * 70  
        pembagi += tinggi  
    if (sgtTinggi != 0):  
        totalValue += sgtTinggi * 100  
        pembagi += sgtTinggi  
  
    return totalValue / pembagi
```

4.2 Fungsi Keanggotaan Defuzzifikasi

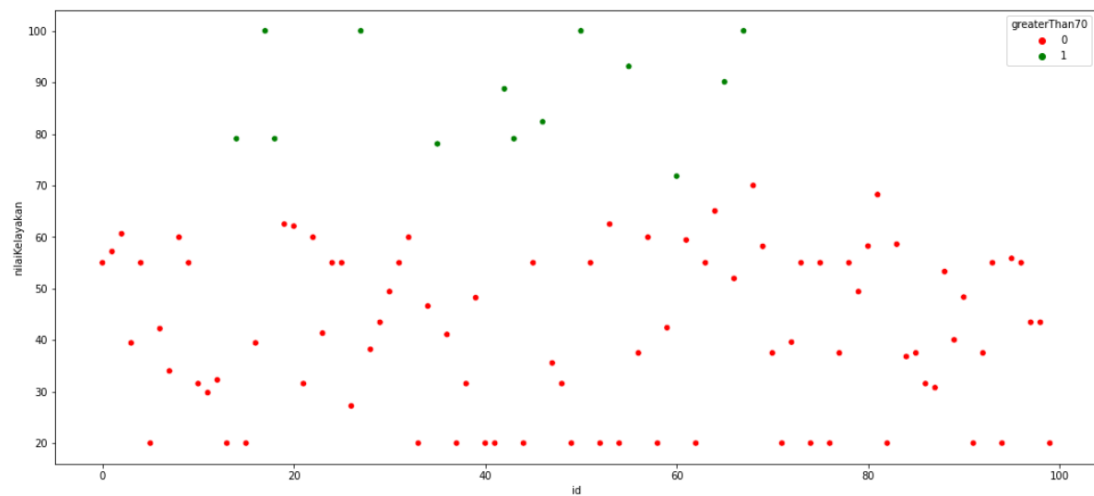
Berikut adalah kodingan plot daripada fungsi keanggotaan Kelayakan.

```
kelayakan= np.arange(1, 101,1)  
plt.figure(figsize=(15,8))  
plt.plot(kelayakan,[Kelayakan(x).kelayakanSangatRendah() for x in kelayakan],label="Sangat Rendah")  
plt.plot(kelayakan,[Kelayakan(x).kelayakanRendah() for x in kelayakan],label="Rendah")  
plt.plot(kelayakan,[Kelayakan(x).kelayakanTinggi() for x in kelayakan],label="Tinggi")  
plt.plot(kelayakan,[Kelayakan(x).kelayakanSangatTinggi() for x in kelayakan],label="Sangat Tinggi")  
plt.title("Nilai Kelayakan")  
plt.legend()
```

E. Output Program

Berikut adalah luaran dari pada program fuzzy yang telah dibuat, luaran ini terdiri dari 10 restoran terbaik beserta seluruh sebaran datanya.

	id	NilaiMinuman	NilaiMakanan	NilaiPelayananan	nilaiKelayakan
50	50	9	8	88	100.000000
27	27	8	10	83	100.000000
17	17	10	5	93	100.000000
67	67	10	8	80	100.000000
55	55	9	6	88	93.103448
65	65	7	5	88	90.100000
42	42	8	9	77	88.750000
46	46	9	8	66	82.352941
14	14	9	7	74	79.071856
18	18	7	9	69	79.071856



BAB III PENUTUP

A. Kesimpulan

Dari yang telah dipaparkan pada bab sebelumnya. Kita dapat mengetahui bahwasanya terdapat beberapa cara untuk menentukan fungsi, aturan bahkan metode dalam fuzzy logic. Seperti yang kami bahas sebelumnya, yaitu mengenai pelayanan dan juga kualitas makanan dan minuman pada sebuah restoran. Didapat bahwasanya 10 restoran terbaik pada kota A memiliki ID : 50, 27, 17, 67, 55, 65, 42, 46, 14, dan 18.

B. Daftar Pustaka

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

<https://drive.google.com/file/d/1z3v3V8uDffzOh3RhpP0wYtMJpPNIItKp0/view> Accessed 30 May 2021

Ibrahim, Ahmad M. (1997). Introduction to Applied Fuzzy Electronics. Englewood Cliffs, NJ: Prentice Hall. ISBN 978-0-13-206400-2.

C. Tautan Google Drive

Video Kelompok:

https://drive.google.com/file/d/1FRgqzAYmh5OVgP0W_COQ0B4NfOeUIAk3/view?usp=sharing

Tautan dataset :

https://drive.google.com/file/d/11aRsqOlcon_MwuCfs6Wtf8x4vqpeuQI8/view?usp=sharing