

**LAPORAN TUGAS PENGANTAR KECERDASAN  
BUATAN**



Disusun oleh :

Bhagas Ade Pramono 1304201018

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**UNIVERSITAS TELKOM BANDUNG**

**BANDUNG**

**2021**

## **Daftar Isi**

<b>Strategi Penyelesaian Masalah</b>	<b>3</b>
Jumlah dan Nama Linguistik Setiap Input	3
Bentuk dan Batas Fungsi Keanggotaan Input	3
Logika pengambilan keputusan (fuzzy inference)	4
Metode Defuzzifikasi	4
Bentuk dan Batas Fungsi Keanggotaan Output	4
<b>Proses yang dibangun</b>	<b>5</b>
Membaca File	5
Fuzzifikasi	5
Logika pengambilan keputusan (fuzzy inference)	7
Defuzzifikasi	8
Hasil	9
<b>Parameter Fuzzy Paling Optimum</b>	<b>10</b>
<b>Kesimpulan</b>	<b>10</b>
<b>Hasil Percobaan</b>	<b>11</b>
<b>Referensi</b>	<b>12</b>

## A. Strategi Penyelesaian Masalah

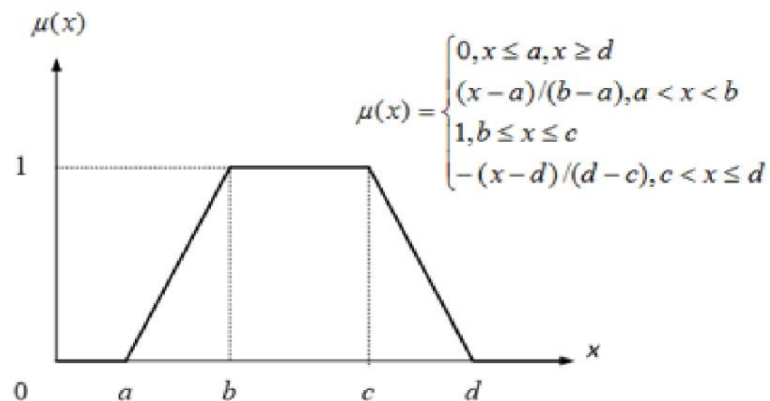
### 1. Jumlah dan Nama Linguistik Setiap Input

Variabel linguistik adalah suatu interval numerik dan mempunyai nilai-nilai linguistik, yang semantiknya didefinisikan oleh fungsi keanggotaannya. Pada kasus ini, terdapat dua variabel linguistik yaitu pelayanan dan harga kami membagi variabel linguistik menjadi empat nilai.

Makanan		Pelayanan	
Nilai	Interval	Nilai	Interval
Tidak Enak (TE)	1 - 3	Sangat Buruk (SR)	1 - 35
Cukup Enak (CE)	4 - 6	Buruk (R)	36 - 60
Enak (E)	7 - 8	Baik (B)	61 - 75
Sangat Enak (SE)	9 - 10	Sangat Baik (SB)	76 - 100

### 2. Bentuk dan Batas Fungsi Keanggotaan Input

Dengan menggunakan array dengan length sebanyak 4 untuk fungsi keanggotaan pada sistem fuzzy ini yang dimana setiap indexnya merepresentasikan bentuk kurva linear yang akan membentuk sebuah trapesium. Bentuk dari kurva menyerupai gambar dibawah ini :

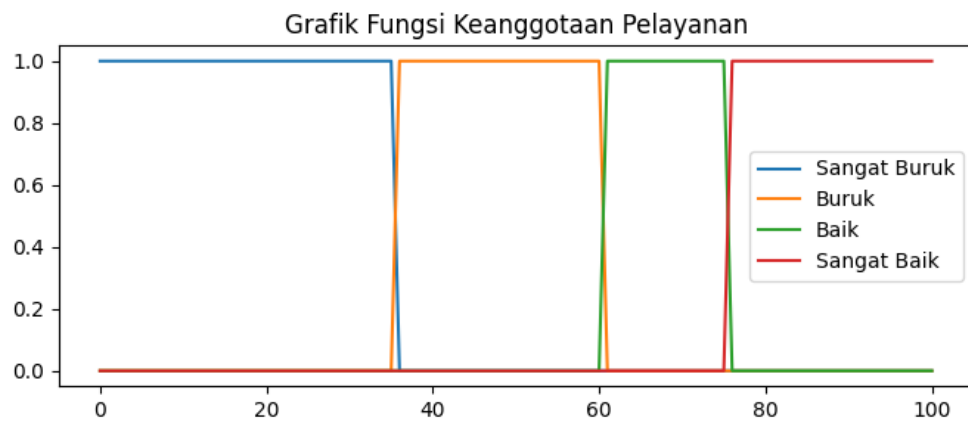


Pada pembangunan sistem fuzzy ini, batas keanggotaan yang digunakan pada input pertama yaitu atribut servis/pelayanan dengan ketentuan :

fkpelayanan(x):

pelayanan = {'SR': 0, 'R' : 0, 'B' : 0, 'SB' : 0}

titik = [35, 36, 60, 61, 75, 76]

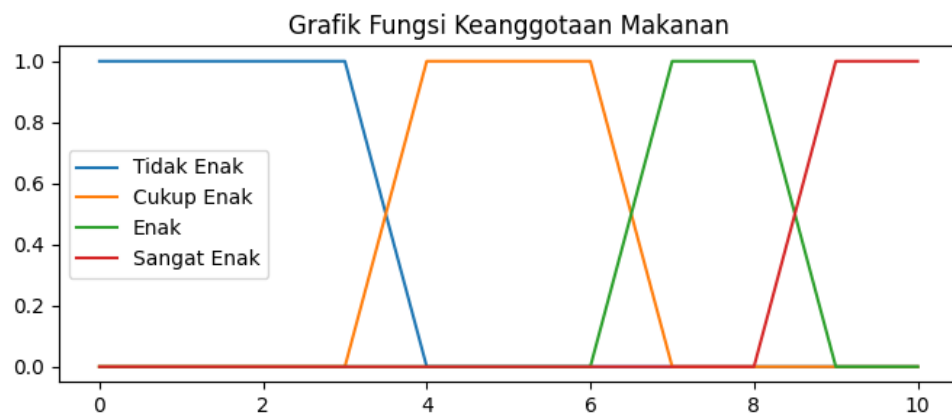


Lalu batas ketentuan keanggotaan yang digunakan pada input kedua yaitu atribut makanan dengan ketentuan :

fkmakanan(x):

makanan = {'TE': 0, 'CE' : 0, 'E' : 0, 'SE' : 0}

titik = [3, 4, 6, 7, 8, 9]



### 3. Logika pengambilan keputusan (fuzzy inference)

Inferensi dalam KBBI merupakan simpulan

**inferensi** /in-fe-ren-si/ /inférénsi/ *n* simpulan; yang disimpulkan

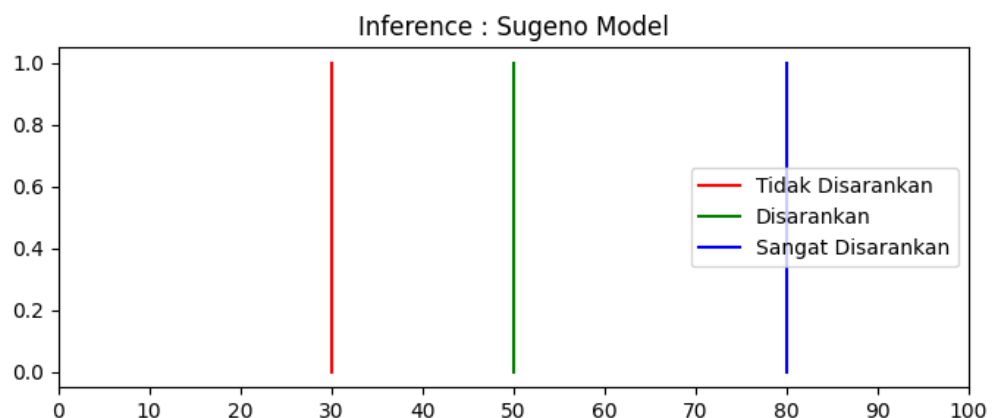
Pada konteks fuzzy logic, inferensi digunakan untuk menyimpulkan korelasi antara nilai linguistik. Pada kasus ini saya membuat tiga jenis kesimpulan untuk menyatakan inferensi antara dua nilai linguistik yaitu **Tidak Disarankan, Disarankan, Sangat Disarankan** korelasi data yang dibuat menjadi seperti dibawah ini:

	Pelayanan (SR)	Pelayanan (R)	Pelayanan (B)	Pelayanan (SB)
Makanan (TE)	Tidak Disarankan	Tidak Disarankan	Tidak Disarankan	Tidak Disarankan
Makanan (CE)	Tidak Disarankan	Tidak Disarankan	Disarankan	Disarankan
Makanan (E)	Tidak Disarankan	Disarankan	Disarankan	Sangat Disarankan
Makanan (SE)	Tidak Disarankan	Disarankan	Sangat Disarankan	Sangat Disarankan

### 4. Metode Defuzzifikasi

Metode constant defuzzification (Takagi-Sugeno-Style) yang saya gunakan sebagai metode perhitungan sistem fuzzy dimana metode ini memilih nilai konstan sebagai acuan untuk menentukan output dari setiap linguistik, kemudian menghitung crisp nya dengan rumus :

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



Contoh grafik yang dihasilkan dari (Takagi-Sugeno-Style)

## 5. Bentuk dan Batas Fungsi Keanggotaan Output

Pada penelitian ini, fungsi keanggotaan output yang digunakan berbentuk segitiga, linier naik, dan linier turun. Terdapat lima jenis Nilai Kelayakan, yaitu disappointing, meh, soso, okay, dan lit.

Batas-batas nilai kelayakan berbeda-beda. Suatu restoran akan dikatakan disappointing apabila nilai kelayakannya berada di 0 hingga 30, dikatakan meh apabila nilai kelayakannya berada di 0 hingga 50, dikatakan soso apabila nilai kelayakannya berada di 30 hingga 70, dikatakan okay apabila nilai kelayakannya berada di 50 hingga 100, dan dikatakan disappointing apabila nilai kelayakannya berada di 70 hingga 100.

## B. Proses yang dibangun

### 1. Membaca File

Metode yang digunakan untuk membaca file restoran.xlsx adalah menggunakan pandas sebagai library agar struktur data dan analisis data mudah digunakan dan untuk membuat tabel, mengubah dimensi data, mengecek data, dan lain sebagainya. Sebelumnya file restoran.xlsx telah dimasukkan ke github terlebih dahulu untuk memudahkan proses pembacaan file. Implementasi pada program sebagai berikut :

```
[2] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

### Membaca Data Set

```
[3] url = 'https://github.com/bhagasee/Fuzzy-logic/blob/main/restoran.xlsx?raw=true'
df = pd.read_excel(url)
df.head()
```

Terdapat variabel baru yang kami buat, seperti df, pelayanan dan makanan. Variabel df digunakan untuk menyimpan keseluruhan data pada file restoran.xlsx, variabel pelayanan untuk menyimpan data file khususnya kolompelayanan, dan variabel makanan untuk menyimpan data file khususnya kolommakanan.

### 2. Fuzzifikasi

Pada proses fuzzifikasi digunakan fungsi keanggotaan input dengan bentuk segitiga, linier naik dan linier turun untuk variabel linguistik pelayanan dan makanan. Implementasi code fungsi anggota pada program menggunakan bahasa python 3 sebagai berikut :

```
~ Fuzzifikasi

[9] def fuzzifikasi(data):
    hasilFuzzi = []

    for i in range(len(data)):
        hasil = [0,0]
        hasil[0] = fkpelayanan(data['pelayanan'][i])
        hasil[1] = fkmakanan(data['makanan'][i])

        hasilFuzzi.append(hasil)
    return hasilFuzzi

[10] print('Example of Fuzzification Result :\n')
print("Restoran ID : ", df['id'][0])
print("Pelayanan : ", fuzzifikasi(df)[1][0])
print("Makanan : ", fuzzifikasi(df)[0][1])

Example of Fuzzification Result :

Restoran ID : 1
Pelayanan : {'SR': 0, 'R': 1, 'B': 0, 'SB': 0}
Makanan : {'TE': 0, 'CE': 0.0, 'E': 1, 'SE': 0}
```

## Variabel linguistik Makanan

```
Fungsi Keanggotaan Makanan

[5] def fMakanan(x):
    makanan = {'TE': 0, 'CE': 0, 'E': 0, 'SE': 0}
    titik = [3, 4, 6, 7, 8, 9]

    #Kategori makanan tidak enak (TE)
    if x <= titik[0]:
        makanan['TE'] = 1
    if titik[0] < x <= titik[1]:
        makanan['TE'] = -(x - titik[1]) / (titik[1] - titik[0])

    #Kategori makanan cukup enak (CE)
    if titik[0] < x < titik[1]:
        makanan['CE'] = (x - titik[0]) / (titik[1] - titik[0])
    if titik[1] <= x <= titik[2]:
        makanan['CE'] = 1
    if titik[2] < x <= titik[3]:
        makanan['CE'] = -(x - titik[3]) / (titik[3] - titik[2])

    #Kategori makanan enak (E)
    if titik[2] < x < titik[3]:
        makanan['E'] = (x - titik[2]) / (titik[3] - titik[2])
    if titik[3] <= x <= titik[4]:
        makanan['E'] = 1
    if titik[4] < x <= titik[5]:
        makanan['E'] = -(x - titik[5]) / (titik[5] - titik[4])

    #Kategori makanan sangat enak (SE)
    if x >= titik[5]:
        makanan['SE'] = 1
    if titik[4] < x <= titik[5]:
        makanan['SE'] = (x - titik[4]) / (titik[5] - titik[4])

    return makanan
```

## Variable linguistik Pelayanan

```
[7] def fKelayanan(x):
    pelayanan = {'SR': 0, 'R': 0, 'B': 0, 'SB': 0}
    titik = [35, 36, 60, 61, 75, 76]

    #Kategori pelayanan sangat buruk (SR)
    if x <= titik[0]:
        pelayanan['SR'] = 1
    if titik[0] < x <= titik[1]:
        pelayanan['SR'] = -(x - titik[1]) / (titik[1] - titik[0])

    #Kategori pelayanan buruk (R)
    if titik[0] < x < titik[1]:
        pelayanan['R'] = (x - titik[0]) / (titik[1] - titik[0])
    if titik[1] <= x <= titik[2]:
        pelayanan['R'] = 1
    if titik[2] < x <= titik[3]:
        pelayanan['R'] = -(x - titik[3]) / (titik[3] - titik[2])

    #Kategori pelayanan baik (B)
    if titik[2] < x < titik[3]:
        pelayanan['B'] = (x - titik[2]) / (titik[3] - titik[2])
    if titik[3] <= x <= titik[4]:
        pelayanan['B'] = 1
    if titik[4] < x <= titik[5]:
        pelayanan['B'] = -(x - titik[5]) / (titik[5] - titik[4])

    #Kategori pelayanan sangat baik (SB)
    if x >= titik[5]:
        pelayanan['SB'] = 1
    if titik[4] < x <= titik[5]:
        pelayanan['SB'] = (x - titik[4]) / (titik[5] - titik[4])

    return pelayanan
```

### 3. Logika pengambilan keputusan (*fuzzy inference*)

Proses *inference* dilakukan dengan pemodelan mamdani yang perhitungannya lebih intuitif mengikuti perasaan manusia. Pada sisi input, digunakan fungsi keanggotaan Segitiga, Linier naik dan Linier turun untuk Pelayanan dan Makanan. Pada sisi output, digunakan fungsi keanggotaan yang sama yaitu Segitiga, Linier naik dan Linier turun untuk kelayakan yang mendapatkan nilai restoran. Terakhir, digunakan tabel berisi dua belas aturan inferensi sebagai berikut:

	Pelayanan (SR)	Pelayanan (R)	Pelayanan (B)	Pelayanan (SB)
Makanan (TE)	Tidak Disarankan	Tidak Disarankan	Tidak Disarankan	Tidak Disarankan
Makanan (CE)	Tidak Disarankan	Tidak Disarankan	Disarankan	Disarankan
Makanan (E)	Tidak Disarankan	Disarankan	Disarankan	Sangat Disarankan
Makanan (SE)	Tidak Disarankan	Disarankan	Sangat Disarankan	Sangat Disarankan

Aturan pada tabel menggunakan koneksi disjuncton untuk mendapatkan derajat kebenaran dari setiap aturan, maka dipilih nilai paling minimum dari suatu nilai linguistik.

Implementasi code pada program menggunakan bahasa python 3 sebagai berikut :

```
def fuzzyRules(x, y) :
    rules = ''
    if x == 'TE':
        if y == 'SR' or y == 'R' or y == 'B' or y == 'SB':
            rules = 'Tidak Disarankan'
    elif x == 'CE':
        if y == 'SR' or y == 'R':
            rules = 'Tidak Disarankan'
        elif y == 'B' or y == 'SB':
            rules = 'Disarankan'
    elif x == 'E':
        if y == 'SR':
            rules = 'Tidak Disarankan'
        elif y == 'R' or y == 'B':
            rules = 'Disarankan'
        elif y == 'SB':
            rules = 'Sangat Disarankan'
    elif x == 'SE':
        if y == 'SR':
            rules = 'Tidak Disarankan'
        elif y == 'R':
            rules = 'Disarankan'
        elif y == 'B' or y == 'SB':
            rules = 'Sangat Disarankan'

    return rules
```

```
def inferensi(nilaiFuzzi):
    hasil = {'Tidak Disarankan': 0, 'Disarankan': 0, 'Sangat Disarankan': 0}
    for keyPelayanan in nilaiFuzzi[0].keys():
        for keyMakanan in nilaiFuzzi[1].keys():
            nilaiMin = min(nilaiFuzzi[0][keyPelayanan], nilaiFuzzi[1][keyMakanan])
            keyHasil = fuzzyRules(keyMakanan, keyPelayanan)
            hasil[keyHasil] = max(nilaiMin, hasil[keyHasil])
    return hasil
```

```
#Contoh hasil inferensi
hasilInferensi = []
hasilFuzzi = fuzzifikasi(df)
for i in hasilFuzzi:
    hasilInferensi.append(inferensi(i))
print('Example of Inference Result :\n')
print('Restoran ID: ', df['id'][0])
print('Status: ', hasilInferensi[0], end="\n\n")
```

Example of Inference Result :

```
Restoran ID: 1
Status: {'Tidak Disarankan': 0, 'Disarankan': 1, 'Sangat Disarankan': 0}
```



#### 4. Defuzzifikasi

Proses defuzzifikasi dilakukan dengan membandingkan nilai kelayakan dengan fungsi keanggotaan masing-masing nilai linguistik yang ada pada variabel linguistik kelayakan untuk mendapatkan daerah arsiran pada grafik.

Implementasi code pada program menggunakan bahasa python 3 sebagai berikut :

```
defuzz = {'Tidak Disarankan' : 30, 'Disarankan' : 50, 'Sangat Disarankan' : 80}
def defuzzification(inferensi, deffuz):
    w, z = 0, 0
    for result in deffuz.keys():
        w += inferensi[result] * deffuz[result]
        z += inferensi[result]
    return w/z
```

Tapi sebelum melakukan defuzzifikasi, kita dapat memetakan nilai kelayakan dengan Sugeno Singleton Model dengan implementasi code sebagai berikut :

```
plt.figure(figsize=(8,3))
plt.plot([30, 30],[0, 1], 'r', linewidth=1.5, label= 'Tidak Disarankan')
plt.plot([50, 50],[0, 1], 'g', linewidth=1.5, label= 'Disarankan')
plt.plot([80, 80],[0, 1], 'b', linewidth=1.5, label= 'Sangat Disarankan')
plt.title('Inference : Sugeno Model')
plt.xticks(np.arange(0, 110, 10.0))
plt.legend()
plt.show()
```

#### 5. Hasil

Proses output hasil pada program dilakukan dengan membuat data frame dari hasil klasifikasi menggunakan fuzzy logic kemudian membuat kolom dengan nama hasil yang menampilkan nilai kelayakan dari setiap restoran. Karena data hasil akan digabungkan dengan data awal maka diperlukan nilai unik untuk menggabungkan data yaitu id dari restoran yang ditampilkan dalam kolom id. Kemudian dilakukan inner join terhadap data awal dan data hasil dengan id restoran sebagai nilai unik lalu dilakukan pengurutan (*sorting*) secara menurun (*descending*) berdasarkan nilai hasil. Terakhir hasil penggabungan data awal dan data hasil di export dalam bentuk excel.

Implementasi code pada program menggunakan bahasa python 3 sebagai berikut :

```
hasilFuzzi = fuzzifikasi(df)
hasilInferensi = []
for i in hasilFuzzi:
    hasilInferensi.append(inferensi(i))
hasilDefuzz = []
for j in hasilInferensi:
    hasilDefuzz.append(defuzzification(j, defuzz))

print('Example of Defuzzification Result :\n')
print('Restoran ID: ', df['id'][10])
print('Nilai: ', hasilDefuzz[10])

Example of Defuzzification Result :

Restoran ID: 11
Nilai: 30.0

df['score'] = hasilDefuzz
bestResult = df.sort_values(by='score', ascending=False)[:10]
notBest = df.drop(bestResult.index)

bestResult['id'].to_excel('peringkat.xlsx', index=False, header=False)
bestResult.head(10)
```

### C. Parameter Fuzzy paling Optimum

- |  |  |
|--|--|
| 1. Jumlah variabel linguistik              | : 3 (Pelayanan (4), Makanan (4), Kelayakan (3)). |
| 2. Bentuk fungsi keanggotaan input Turun.  | : Segitiga, Linier Naik, dan Linier Turun.       |
| 3. Aturan inferensi                        | : pemodelan Mamdani.                             |
| 4. Defuzzifikasi                           | : Sugeno Singleton Model.                        |
| 5. Bentuk fungsi keanggotaan output Turun. | : Segitiga, Linier Naik, dan Linier Turun.       |
| 6. Batas fungsi keanggotaan output         | : [0, 100] 0 sampai 100.                         |

### D. Tautan video presentasi :

<https://github.com/bhagasee/Fuzzy-logic>

## E. Kesimpulan

Kesimpulan yang didapatkan dari penelitian yang kami lakukan adalah fuzzy logic merupakan metode penalaran yang menyerupai penalaran manusia dengan melibatkan semua kemungkinan dalam pengambilan keputusan, untuk itu fuzzy logic dapat digunakan untuk masalah yang mengandung ketidakpastian.

Dalam merancang sistem kontrol menggunakan *fuzzy logic* terdapat tiga proses yaitu fuzzifikasi, *fuzzy inference* dan defuzzifikasi. Masing-masing proses tersebut akan mempengaruhi hasil dari program. Proses fuzzifikasi merubah *crisp input* menjadi *fuzzy input* yang digunakan pada proses *inference* menggunakan serangkaian *fuzzy rules* kemudian menghasilkan *fuzzy output*, melalui proses defuzzifikasi, *fuzzy output* diubah menjadi *crisp value* yang menjadi hasil akhir.

Penelitian ini dilakukan dengan membangun sebuah sistem menggunakan *fuzzy logic* untuk memilih sepuluh restoran pada file restoran.xls berdasarkan kualitas pelayanan dan makanannya kemudian menghasilkan output berupa sebuah file peringkat.xls.

## F. Hasil percobaan

id		pelayanan	makanan	score
23	24	100	9	80.0
41	42	94	10	80.0
24	25	61	10	80.0
30	31	74	9	80.0
78	79	87	9	80.0
69	70	78	8	80.0
68	69	86	10	80.0
62	63	78	7	80.0
21	22	79	9	80.0
53	54	64	10	80.0

## ***Referensi***