

LAPORAN TUGAS PEMROGRAMAN II
PENGANTAR KECERDASAN BUATAN



Disusun Oleh:

Muhammad Daffa' Ibrahim (1301204051)

Raditya Pradhyatama Nugroho (1301201597)

IF-44-06

FAKULTAS INFORMATIKA
PRODI S1 INFORMATIKA
UNIVERSITAS TELKOM
TAHUN 2022

BAB I

PENDAHULUAN

1.1. Latar Belakang

Logika Fuzzy adalah suatu logika yang memiliki nilai kekaburan atau kesamaran antara benar dan salah sederhananya, logika fuzzy adalah logika yang mengandung ketidakpastian. Algoritma Fuzzy adalah suatu penyelesaian masalah berupa data yang memiliki nilai samar antara benar dan salah. Dalam tugas pemrograman ke-2 ini, metode algoritma fuzzy digunakan untuk mengolah data bengkel mobil dalam menentukan bengkel dengan servis terbaik dan harga yang terjangkau.

1.2. Tujuan

Tujuan dari penyusunan laporan ini adalah:

- a. Untuk memenuhi tugas mata kuliah Pengantar Kecerdasan Buatan
- b. Sebagai bukti tertulis dari program komputer algoritma fuzzy yang telah disusun
- c. Mengetahui peranan fuzzy logic dalam menyelesaikan masalah.

1.3. Deskripsi Tugas/Masalah

Terdapat file **bengkel.xlsx** berupa himpunan data 100 bengkel mobil yang ada di kota Bandung dengan dua atribut: Kualitas Servis (bilangan real 1-100; semakin tinggi semakin baik) dan Harga (bilangan real 1-10, semakin tinggi semakin mahal). Bangunlah sebuah sistem berbasis Fuzzy Logic untuk memilih 10 bengkel terbaik di kota Bandung. Sistem membaca masukan file **bengkel.xlsx** dan mengeluarkan output berupa sebuah file **peringkat.xlsx** yang berisi 10 nomor/ID bengkel terbaik beserta skor-nya (output Defuzzification).

BAB II

ANALISIS

2.1. Jumlah dan Nama Linguistik Setiap Atribut Input

Terdapat dua atribut, yakni **Kualitas Servis** dan **Harga** yang masing-masing memiliki 4 dan 3 variabel linguistik:

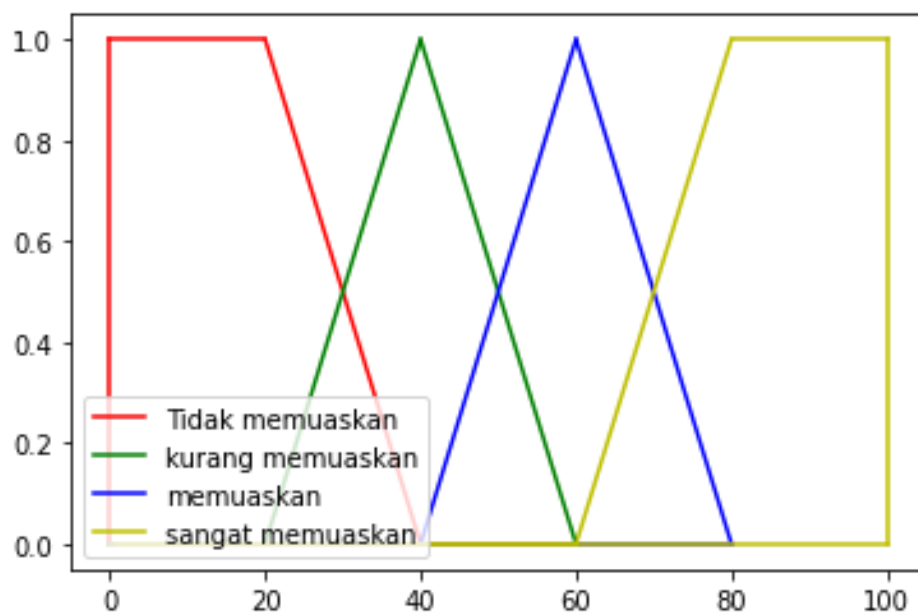
- a. Kualitas Servis : Sangat Memuaskan, Memuaskan, Kurang Memuaskan, Tidak Memuaskan
- b. Harga : Murah, Normal, Mahal.

2.2. Bentuk dan Batas Fungsi Keanggotaan Input

Bentuk grafik fungsi yang digunakan adalah *Linear Function* yang diibaratkan berbentuk trapesium atau segitiga.

a. Batas Fungsi **Kualitas Servis**

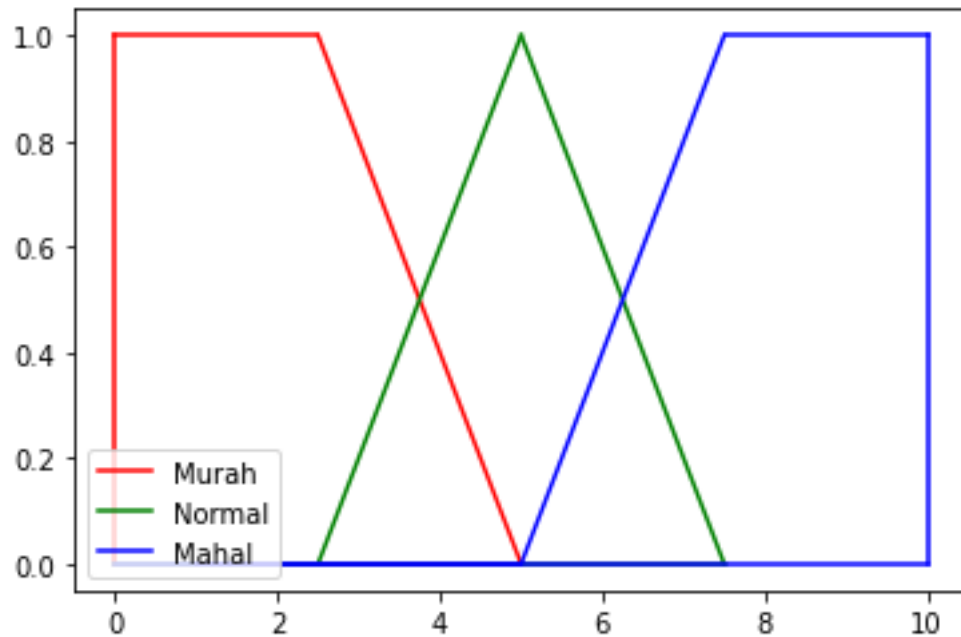
- i. Sangat Memuaskan : $x > 80$ (sangat memuaskan) dan $x \leq 60$ (tidak sangat memuaskan), dengan fuzzy area $(60,80]$
- ii. Memuaskan : $x = 60$, dengan fuzzy area $[40,60)$ dan $[60,80)$
- iii. Kurang Memuaskan : $x = 40$, dengan fuzzy area $[20,40)$ dan $[40,60)$
- iv. Tidak Memuaskan : $x \leq 20$ (tidak memuaskan) dan $x \geq 0$ (tidak memuaskan), dengan fuzzy area $(20,30)$.



Gambar 2.2.1 Grafik fungsi keanggotaan kualitas servis

b. Batas Fungsi **Harga**

- 1.1. Murah : $x \leq 2,5$ (Murah) dan $x \geq 5$ (Tidak Murah), dengan fuzzy area $(2.5, 5]$
- 2.1. Normal : $x = 5$ dengan fuzzy area $[2.5, 5)$ dan $[5, 7.5)$
- 3.1. Mahal : $x \geq 7$ (Mahal) dan $x \leq 5$ (Tidak Mahal), dengan fuzzy area $(5, 7.5]$.



Gambar 2.2.2 Grafik fungsi keanggotaan harga

2.3. Aturan Inferensi

Berdasarkan kualitas servis dan harga yang menjadi input algoritma. menghasilkan tiga kategori nilai, yaitu Baik, Standar, dan Buruk. Kemudian dari ketiga kategori nilai ini, menghasilkan 12 aturan inferensi seperti di bawah ini:

Kualitas/Harga	Murah	Normal	Mahal
Sangat Memuaskan	Baik	Standar	Buruk
Memuaskan	Baik	Standar	Buruk
Kurang Memuaskan	Standar	Standar	Buruk
Tidak Memuaskan	Buruk	Buruk	Buruk

Gambar 2.3.1 Tabel aturan inferensi

2.4. Metode Defuzzification

Metode defuzzifikasi yang digunakan dalam penyusunan algoritma ini adalah Metode Takagi-Sugeno. Metode ini merupakan metode yang sesuai dengan permasalahan ini karena dengan input dan output algoritma berupa string, metode sugeno ini lebih sederhana dalam penyusunannya serta tidak membutuhkan model matematika yang rumit didalamnya jika dibandingkan dengan Metode Mamdani. Berikut rumus dari Metode Sugeno:

$$z^* = \frac{\sum_{i=1}^l \mu B_i \cdot c_i}{\sum_{i=1}^l \mu B_i}$$

$c_i = \text{constant for } i^{\text{th}} \text{ linguistic}$
 $\mu B_i = \text{membership for } i^{\text{th}} \text{ linguistic}$

Gambar 2.4.1. Rumus metode defuzzifikasi

2.5. Bentuk dan Batas Fungsi Keanggotaan Output

Batas fungsi keanggotaan pada nilai kelayakan

Nilai Kelayakan : Baik = 75, Standar = 50, Buruk = 25.

BAB III

HASIL EKSEKUSI PROGRAM

3.1. Mengimpor Data dan Membaca File *bengkel.xlsx*

```
#Mengimport data dari file bengkel.xlsx
from google.colab import files
TheFile = files.upload()
```

```
#Membaca file bengkel.xlsx
import pandas as pd
data = pd.read_excel('bengkel.xlsx')
print(data)
```

	id	servis	harga
0	1	58	7
1	2	54	1
2	3	98	2
3	4	52	4
4	5	11	4
..
95	96	30	1
96	97	25	3
97	98	27	10
98	99	8	6
99	100	11	8

[100 rows x 3 columns]

3.2. Membuat Grafik Fungsi Keanggotaan

```
#Membuat grafik
import matplotlib.pyplot as plt

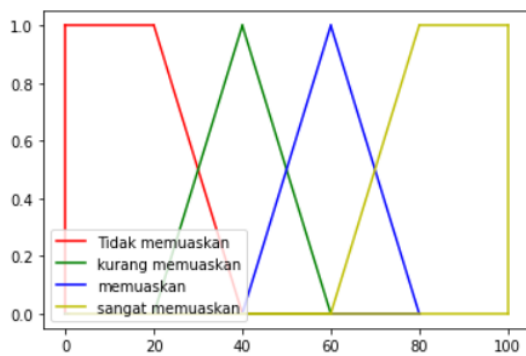
servis = {
    'Tidak memuaskan' : [0.00, 0.00, 20.00,40.00 ],
    'kurang memuaskan' : [20.00, 40.00 , 40.00, 60.00],
    'memuaskan' : [40.00, 60.00, 60.00, 80.00],
    'sangat memuaskan' : [60.00, 80.00, 100.00, 100.00],
}

Harga = {
    'Murah' : [0.00, 0.00, 2.50, 5.00],
    'Normal' : [2.50, 5.00, 5.00, 7.50],
    'Mahal' : [5.00, 7.50, 10.00, 10.00]
}

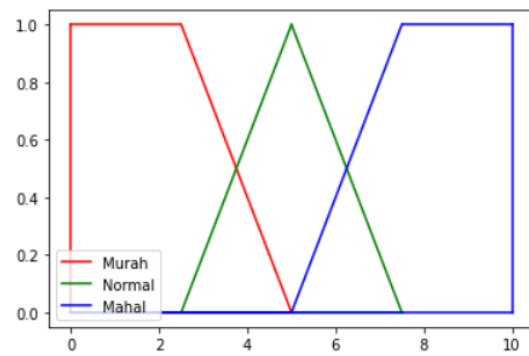
def nilai_Linguistik(batas, warna, label=' ', min=0, max=1):
    y = [0,1,1,0]
    plt.plot((min,batas[0]), (0,0), warna)
    for i in range(len(batas)-1):
        plt.plot((batas[i], batas[i+1]), (y[i], y[i+1]), warna)
    plt.plot((batas[-1], max), (0,0), warna, label=label)

def grafik(membership, min = 0, max=1):
    warna = ['r', 'g', 'b', 'y', 'm', 'c', 'k']
    i = 0
    for j in membership:
        nilai_Linguistik(membership[j], warna[i], j, min=min, max=max)
        i += 1
    plt.legend(loc=3)
    plt.show
```

grafik(servis)



grafik(Harga)



3.3. Fuzzifikasi

```
#Fuzzifikasi kualitas servis
#Nilai linguistik = Sangat Memuaskan, Memuaskan, Kurang memuaskan, Tidak Memuaskan

def fuzzifikasiKualitasServis(servis):
    hasil = [["Tidak Memuaskan",0], ["Kurang Memuaskan",0], ["Memuaskan",0], ["Sangat Memuaskan",0]]

    if servis <= 20:
        hasil[0][1] = 1
    elif servis > 20 and servis < 40:
        x = (40 - servis)/(40 - 20)
        y = (servis - 20)/(40 - 20)
        hasil[0][1] = x
        hasil[1][1] = y
    elif servis == 40:
        hasil[1][1] = 1
    elif servis > 40 and servis < 60:
        x = (60 - servis)/(60 - 40)
        y = (servis - 40)/(60 - 40)
        hasil[1][1] = x
        hasil[2][1] = y
    elif servis == 60:
        hasil[2][1] = 1
    elif servis > 60 and servis < 80:
        x = (80 - servis)/(80 - 60)
        y = (servis - 60)/(80 - 60)
        hasil[2][1] = x
        hasil[3][1] = y
    elif servis >= 80:
        hasil[3][1] = 1

    return hasil
```

```

#Fuzifikasi Harga
#Nilai linguistik = Mahal, Normal, Murah
def fuzifikasiHarga(harga):
    hasil = [["Murah",0], ["Normal",0], ["Mahal",0],]
    if harga <= 2.5:
        hasil[0][1] = 1
    elif harga > 2.5 and harga < 5:
        x = (5 - harga)/(5 - 2.5)
        y = (harga - 2.5)/(5 - 2.5)
        hasil[0][1] = x
        hasil[1][1] = y
    elif harga == 5:
        hasil[1][1] = 1
    elif harga > 5 and harga < 7.5:
        x = (7.5 - harga)/(7.5 - 5)
        y = (harga - 5)/(7.5 - 5)
        hasil[1][1] = x
        hasil[2][1] = y
    elif harga >= 7.5:
        hasil[2][1] = 1

    return hasil

```

3.4. Inferensi

```

#Konjungsi Nilai kelayakan = Baik
def conjunctionbaik(fuzharga, fuzkualitas):
    baik = []
    if (fuzharga[0][0] == "Murah" and fuzkualitas[2][0] == "Memuaskan"):
        if fuzharga[0][1] < fuzkualitas[2][1]:
            nilai = fuzharga[0][1]
        else:
            nilai = fuzkualitas[2][1]
        baik.append(nilai)
    if (fuzharga[0][0] == "Murah" and fuzkualitas[3][0] == "Sangat Memuaskan"):
        if fuzharga[0][1] < fuzkualitas[3][1]:
            nilai = fuzharga[0][1]
        else:
            nilai = fuzkualitas[3][1]
        baik.append(nilai)

    return baik

```



```

#Konjungsi Nilai kelayakan = Standar
def conjunctionstandar(fuzharga, fuzkualitas):
    standar = []

    if (fuzharga[0][0] == "Murah" and fuzkualitas[1][0] == "Kurang Memuaskan"):
        if fuzharga[0][1] < fuzkualitas[1][1]:
            nilai = fuzharga[0][1]
        else:
            nilai = fuzkualitas[1][1]
        standar.append(nilai)
    if (fuzharga[1][0] == "Normal" and fuzkualitas[1][0] == "Kurang Memuaskan"):
        if fuzharga[1][1] < fuzkualitas[1][1]:
            nilai = fuzharga[1][1]
        else:
            nilai = fuzkualitas[1][1]
        standar.append(nilai)
    if (fuzharga[1][0] == "Normal" and fuzkualitas[2][0] == "Memuaskan"):
        if fuzharga[1][1] < fuzkualitas[2][1]:
            nilai = fuzharga[1][1]
        else:
            nilai = fuzkualitas[2][1]
        standar.append(nilai)
    if (fuzharga[1][0] == "Normal" and fuzkualitas[3][0] == "Sangat Memuaskan"):
        if fuzharga[1][1] < fuzkualitas[3][1]:
            nilai = fuzharga[1][1]
        else:
            nilai = fuzkualitas[3][1]
        standar.append(nilai)

    return standar

```

```

#Konjungsi Nilai kelayakan = Buruk
def conjunctionburuk(fuzharga, fuzkualitas):
    buruk = []

    if (fuzharga[0][0] == "Murah" and fuzkualitas[0][0] == "Tidak Memuaskan"):
        if fuzharga[0][1] < fuzkualitas[0][1]:
            nilai = fuzharga[0][1]
        else:
            nilai = fuzkualitas[0][1]
        buruk.append(nilai)
    if (fuzharga[1][0] == "Normal" and fuzkualitas[0][0] == "Tidak Memuaskan"):
        if fuzharga[1][1] < fuzkualitas[0][1]:
            nilai = fuzharga[1][1]
        else:
            nilai = fuzkualitas[0][1]
        buruk.append(nilai)
    if (fuzharga[2][0] == "Mahal" and fuzkualitas[0][0] == "Tidak Memuaskan"):
        if fuzharga[2][1] < fuzkualitas[0][1]:
            nilai = fuzharga[2][1]
        else:
            nilai = fuzkualitas[0][1]
        buruk.append(nilai)
    if (fuzharga[2][0] == "Mahal" and fuzkualitas[1][0] == "Kurang Memuaskan"):
        if fuzharga[2][1] < fuzkualitas[1][1]:
            nilai = fuzharga[2][1]
        else:
            nilai = fuzkualitas[1][1]
        buruk.append(nilai)
    if (fuzharga[2][0] == "Mahal" and fuzkualitas[2][0] == "Memuaskan"):
        if fuzharga[2][1] < fuzkualitas[2][1]:
            nilai = fuzharga[2][1]
        else:
            nilai = fuzkualitas[2][1]

```

```

        buruk.append(nilai)
    if (fuzharga[2][0] == "Mahal" and fuzkualitas[3][0] == "Sangat Memuaskan"):
        if fuzharga[2][1] < fuzkualitas[3][1]:
            nilai = fuzharga[2][1]
        else:
            nilai = fuzkualitas[3][1]
        buruk.append(nilai)
    return buruk

```

3.5. Defuzzifikasi

```

#Defuzzification
#Nilai kelayakan:
#
#           Baik = 75
#           Standar = 50
#           Buruk = 25
def Z(baik, standar, buruk):
    z = ((baik * 75) + (standar * 50) + (buruk * 25)) / (baik + standar + buruk)
    return z

```

3.6. Menentukan Nilai Maksimum

```

#Fungsi Menentukan Nilai Maksimal
def maks(array):
    maks = array[0]
    i = 1
    while i < len(array):
        if array[i] > maks:
            maks = array[i]
        i = i + 1
    return maks

```

3.7. Mengurutkan Array Secara Descending

```

#Fungsi Mengurutkan Array secara Descending
def bubbleSort(arr):
    n = len(arr)

    for i in range(n-1):
        for j in range(0, n-i-1):
            if arr[j][1] < arr[j + 1][1] :
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

```

3.8. Program Utama

```
#MAIN PROGRAM

id = data["id"]
servis = data["servis"]
harga = data["harga"]
dataServis = []

i = 0

#Fuzzy Rules sebanyak data (100 kali)
while i < 100:
    #Fuzzifikasi
    fuzservis = fuzzifikasiKualitasServis(servis[i])
    fuzharga = fuzzifikasiHarga(harga[i])

    #Inferensi
    baik = conjunctionbaik(fuzharga, fuzservis)
    standar = conjunctionstandar(fuzharga, fuzservis)
    buruk = conjunctionburuk(fuzharga, fuzservis)

    maksbaik = maks(baik)
    maksstandar = maks(standar)
    maksburuk = maks(buruk)

    #Defuzzifikasi
    nilaiZ = Z(maksbaik, maksstandar, maksburuk)
    dataServis.append([id[i], nilaiZ])

    i = i + 1

bubbleSort(dataServis)
```

3.9. Menampilkan 10 Bengkel Terbaik

```
#Menampilkan 10 Bengkel Terbaik
i = 0
while i < 10:
    print(i+1, "ID: ",dataServis[i][0], "NILAI: ",dataServis[i][1])
    i = i + 1
```

```
1 ID: 3 NILAI: 75.0
2 ID: 44 NILAI: 75.0
3 ID: 75 NILAI: 75.0
4 ID: 13 NILAI: 70.0
5 ID: 52 NILAI: 70.0
6 ID: 91 NILAI: 70.0
7 ID: 92 NILAI: 70.0
8 ID: 17 NILAI: 67.85714285714286
9 ID: 2 NILAI: 67.5
10 ID: 56 NILAI: 61.25
```

3.10. Mengkonversi Hasil Output ke File Peringkat.xlsx

```
array_data = dataServis[:10]

id = []
skor = []

for i in range(10):
    id.append(array_data[i][0])
    skor.append(array_data[i][1])

#Mengkonversi hasil output ke file Peringkat.xlsx
data = pd.DataFrame(id, columns = ['ID'])
data['Skor'] = skor
data.to_excel('Peringkat.xlsx')
print(data)
```

	ID	Skor
0	3	75.000000
1	44	75.000000
2	75	75.000000
3	13	70.000000
4	52	70.000000
5	91	70.000000
6	92	70.000000
7	17	67.857143
8	2	67.500000
9	56	61.250000

BAB IV

PENUTUP

4.1. Kesimpulan

Berdasarkan algoritma penilaian bengkel mobil yang telah disusun dapat disimpulkan bahwa:

- a. Dalam tahap fuzzifikasi, penentuan nilai batas fungsi keanggotaan akan mempengaruhi pengelompokan dalam nilai linguistik dari kualitas servis dan harga pelayanan suatu bengkel mobil.
- b. Pada tahap inferensi, penentuan nilai batas kelayakan akan mempengaruhi pengelompokan inferensi kelayakan dalam pelayanan suatu bengkel mobil.

Peran Anggota

Muhammad Daffa' Ibrahim :

- Membuat program komputer menggunakan algoritma fuzzy
- Membuat bentuk dan batas fungsi keanggotaan input pada laporan.

Raditya Pradhyatama Nugroho :

- Membuat isi laporan tugas pemrograman
- Melakukan *finishing* pada laporan dan program.

Link Source Code

<https://colab.research.google.com/drive/1d51U9S-gLawpshIidSgkI3fd12-BqZ?usp=sharing>

Link Video Presentasi

<https://drive.google.com/file/d/1P94Vpn5b9t3v2EuSmrIjsXNICmLKJ0na/view?usp=sharing>