

LAPORAN OBSERVASI
TUGAS PEMROGRAMAN 02 – FUZZY LOGIC

diajukan untuk memenuhi tugas mata kuliah (CII – 2M3) Pengantar Kecerdasan Buatan

oleh Kelompok 18:

Otniel Abiezer (NIM 1301180469)

Muhammad Haidir Ali (NIM 1301180205)



PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2021

DAFTAR ISI

1. Hal yang Perlu Diobservasi.....	1
a. Jumlah dan Nama Linguistik Input	1
b. Batas dan Fungsi Keanggotaan Input	1
c. Aturan Inferensi.....	3
d. Metode Defuzzifikasi.....	3
e. Bentuk dan Batas Fungsi Keanggotaan Output.....	3
2. Prosedur/Fungsi yang Harus Dibangun	4
a. Membaca File	4
b. Fuzzifikasi	4
c. Inferensi	5
d. Defuzzifikasi	6
e. Function Lain (Mendapatkan top 10)	6
f. Export ke Excel	7
3. Kesimpulan	7
4. Video Presentasi	7

1. Hal yang Perlu Diobservasi

a. Jumlah dan Nama Linguistik Input

Pada Pelayanan, memiliki batasan input dari 1-100 dan memiliki 3 nama linguistik input yang terdiri dari :

- 1) Buruk
- 2) Sedang
- 3) Baik

Sedangkan, pada Makanan, memiliki batasan input dari 1-10 dan memiliki 3 nama linguistik input yang terdiri dari :

- 1) Tidak Enak
- 2) Biasa
- 3) Enak

```
58 def inBuruk(pelayanan):
59     buruk = []
60     for i in range(0,100):
61         nilai = pelayanan[i]
62         buruk.append(pelayananBuruk(nilai))
63     return buruk
64
65 def inSedang(pelayanan):
66     sedang = []
67     for i in range(0,100):
68         nilai = pelayanan[i]
69         sedang.append(pelayananSedang(nilai))
70     return sedang
71
72 def inBaik(pelayanan):
73     baik = []
74     for i in range(0,100):
75         nilai = pelayanan[i]
76         baik.append(pelayananBaik(nilai))
77     return baik
78
```

```
79 def inTidakEnak(makanan):
80     tidak_enak = []
81     for i in range(0,100):
82         nilai = makanan[i]
83         tidak_enak.append(makananTidakEnak(nilai))
84     return tidak_enak
85
86 def inBiasa(makanan):
87     biasa = []
88     for i in range(0,100):
89         nilai = makanan[i]
90         biasa.append(makananBiasa(nilai))
91     return biasa
92
93 def inEnak(makanan):
94     enak = []
95     for i in range(0,100):
96         nilai = makanan[i]
97         enak.append(makananEnak(nilai))
98     return enak
99
```

b. Batas dan Fungsi Keanggotaan Input

Batas dari keanggotaan di atas adalah :

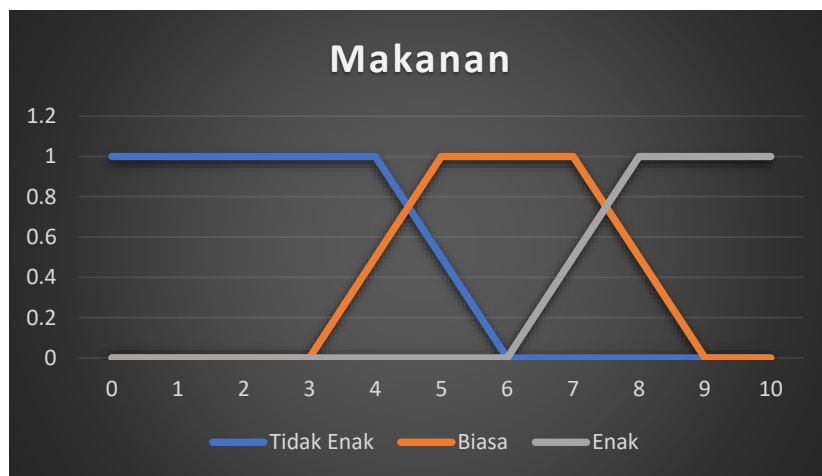
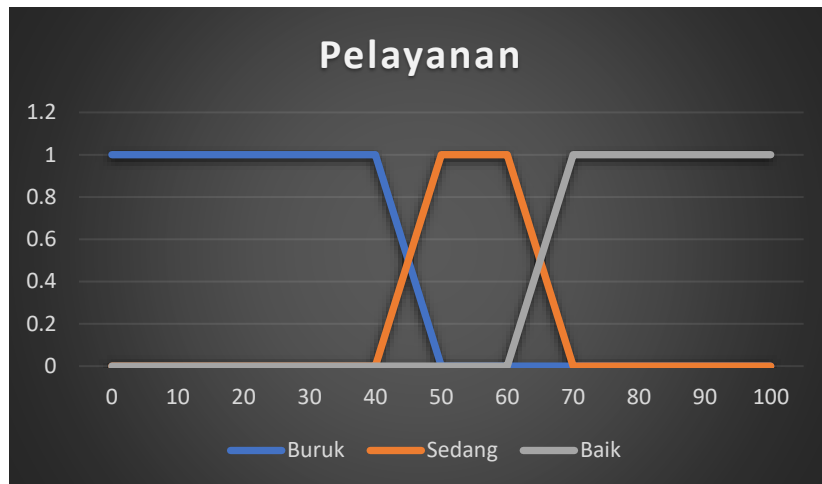
Pelayanan :

- 1) Buruk : $x \leq 40$
- 2) Sedang : $40 < x < 70$
- 3) Baik : $x \geq 70$

Makanan

- 1) Tidak Enak : $x \leq 4$
- 2) Biasa : $5 \leq x \leq 7$
- 3) Enak : $x \geq 8$

Dengan bentuk fungsi keanggotaan adalah Linear yang bisa dilihat seperti di bawah



```

5  def pelayananBuruk(nilai_pelayanan):
6      if (nilai_pelayanan <=40):
7          return 1
8      elif (41 <= nilai_pelayanan <= 50):
9          return (50 - nilai_pelayanan)/10
10     else:
11         return 0
12
13  def pelayananSedang(nilai_pelayanan):
14      if (41 <= nilai_pelayanan <= 50):
15          return (nilai_pelayanan - 40)/10
16      elif (51 <= nilai_pelayanan <= 60):
17          return 1
18      elif (61 <= nilai_pelayanan <= 70):
19          return (70 - nilai_pelayanan)/10
20      else:
21          return 0
22
23  def pelayananBaik(nilai_pelayanan):
24      if (nilai_pelayanan >= 71):
25          return 1
26      elif (61 <= nilai_pelayanan <= 70):
27          return (nilai_pelayanan - 60)/10
28      else:
29          return 0
30

```

```

31  def makananTidakEnak(nilai_makanan):
32      if (nilai_makanan <=4):
33          return 1
34      elif (3 <= nilai_makanan <= 5):
35          return (5 - nilai_makanan)/2
36      else:
37          return 0
38
39  def makananBiasa(nilai_makanan):
40      if (3 <= nilai_makanan <= 5):
41          return (nilai_makanan - 3)/2
42      elif (5 <= nilai_makanan <= 7):
43          return 1
44      elif (7 <= nilai_makanan <= 9):
45          return (9 - nilai_makanan)/2
46      else:
47          return 0
48
49  def makananEnak(nilai_makanan):
50      if (nilai_makanan >= 8):
51          return 1
52      elif (6 <= nilai_makanan <= 8):
53          return (nilai_makanan - 6)
54      else:
55          return 0
56

```

c. Aturan Inferensi

Aturan Inferensi memiliki sebanyak 9 kemungkinan yang didapat dari 3 x 3 (banyak keanggotaan Pelayanan dan Makanan) dengan nilai output ada 3, yaitu Mengecewakan, OK, dan Puas. Dapat dilihat di tabel di bawah:

Pelayanan	Makanan	Output
Buruk	Tidak Enak	Mengecewakan
Buruk	Biasa	Mengecewakan
Buruk	Enak	Mengecewakan
Sedang	Tidak Enak	Mengecewakan
Sedang	Biasa	OK
Sedang	Enak	Puas
Baik	Tidak Enak	OK
Baik	Biasa	Puas
Baik	Enak	Puas

```
#Inference
mengecewakan = tblMengecewakan(buruk, sedang, tidak_enak, biasa, enak)
ok = tblOK(sedang, baik, tidak_enak, biasa)
puas = tblPuas(sedang, baik, biasa, enak)
```

d. Metode Defuzzifikasi

Dengan menggunakan metode Sugeno, karena mudah untuk diimplementasikan

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

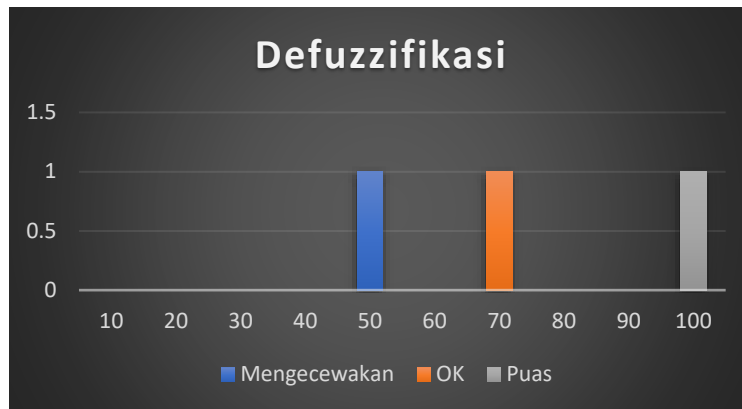
```
#Defuzzification
defuzz = defuzzSugeno(mengecewakan, ok, puas)
```

e. Bentuk dan Batas Fungsi Keanggotaan Output

Dengan menggunakan defuzzifikasi Sugeno, maka memiliki batas berupa nilai konstan sebagai berikut :

- 1) Mengecewakan = 50
- 2) OK = 70
- 3) Puas = 100

Sehingga hasil dari defuzzifikasi memiliki rentang dari 50-100.
Atau dapat digambarkan grafik seperti berikut :



```

123 def defuzzSugeno(mengecewakan, ok, puas):
124     defuzz = []
125
126     for i in range(0,100):
127         defuzz.append((mengecewakan[i]*50 + ok[i]*70 + puas[i]*100)/(mengecewakan[i] + ok[i] + puas[i]))
128     return defuzz
129

```

2. Prosedur/Fungsi yang Harus Dibangun

a. Membaca File

Pada Main Program (Driver)

```

156 #Import xlsx
157 restoran = pd.read_excel('restoran.xlsx')
158 indeks = restoran['id'].tolist()
159 pelayanan = restoran['pelayanan'].tolist()
160 makanan = restoran['makanan'].tolist()

```

b. Fuzzifikasi

Function/Prosedur menghitung nilai Fuzzifikasi (sama dengan 1b)

```

5 def pelayananBuruk(nilai_pelayanan):
6     if (nilai_pelayanan <= 40):
7         return 1
8     elif (41 <= nilai_pelayanan <= 50):
9         return (50 - nilai_pelayanan)/10
10    else:
11        return 0
12
13 def pelayananSedang(nilai_pelayanan):
14     if (41 <= nilai_pelayanan <= 50):
15         return (nilai_pelayanan - 40)/10
16     elif (51 <= nilai_pelayanan <= 60):
17         return 1
18     elif (61 <= nilai_pelayanan <= 70):
19         return (70 - nilai_pelayanan)/10
20    else:
21        return 0
22
23 def pelayananBaik(nilai_pelayanan):
24     if (nilai_pelayanan >= 71):
25         return 1
26     elif (61 <= nilai_pelayanan <= 70):
27         return (nilai_pelayanan - 60)/10
28    else:
29        return 0
30
31 def makananTidakEnak(nilai_makanan):
32     if (nilai_makanan <= 4):
33         return 1
34     elif (3 <= nilai_makanan <= 5):
35         return (5 - nilai_makanan)/2
36    else:
37        return 0
38
39 def makananBiasa(nilai_makanan):
40     if (3 <= nilai_makanan <= 5):
41         return (nilai_makanan - 3)/2
42     elif (5 <= nilai_makanan <= 7):
43         return 1
44     elif (7 <= nilai_makanan <= 9):
45         return (9 - nilai_makanan)/2
46    else:
47        return 0
48
49 def makananEnak(nilai_makanan):
50     if (nilai_makanan >= 8):
51         return 1
52     elif (6 <= nilai_makanan <= 8):
53         return (nilai_makanan - 6)
54    else:
55        return 0
56

```

Function/Prosedur untuk menampung nilai Fuzzifikasi (sama dengan 1a)

```

58 def inBuruk(pelayanan):
59     buruk = []
60     for i in range(0,100):
61         nilai = pelayanan[i]
62         buruk.append(pelayananBuruk(nilai))
63     return buruk
64
65 def inSedang(pelayanan):
66     sedang = []
67     for i in range(0,100):
68         nilai = pelayanan[i]
69         sedang.append(pelayananSedang(nilai))
70     return sedang
71
72 def inBaik(pelayanan):
73     baik = []
74     for i in range(0,100):
75         nilai = pelayanan[i]
76         baik.append(pelayananBaik(nilai))
77     return baik
78
79 def inTidakEnak(makanan):
80     tidak_enak = []
81     for i in range(0,100):
82         nilai = makanan[i]
83         tidak_enak.append(makananTidakEnak(nilai))
84     return tidak_enak
85
86 def inBiasa(makanan):
87     biasa = []
88     for i in range(0,100):
89         nilai = makanan[i]
90         biasa.append(makananBiasa(nilai))
91     return biasa
92
93 def inEnak(makanan):
94     enak = []
95     for i in range(0,100):
96         nilai = makanan[i]
97         enak.append(makananEnak(nilai))
98     return enak
99

```

Penerapan di Main Program (Driver)

```

162 #Fuzzification
163 buruk = inBuruk(pelayanan)
164 sedang = inSedang(pelayanan)
165 baik = inBaik(pelayanan)
166 tidak_enak = inTidakEnak(makanan)
167 biasa = inBiasa(makanan)
168 enak = inEnak(makanan)

```

c. Inferensi

Function/Prosedur

```

100 #=====INFERENCE=====
101 def tblMengecewakan(buruk, sedang, tidak_enak, biasa, enak):
102     mengecewakan = []
103
104     for i in range(0,100):
105         mengecewakan.append(max(min(buruk[i], tidak_enak[i]), min(buruk[i], biasa[i]), min(buruk[i], enak[i]), min(sedang[i], biasa[i]), min(sedang[i], enak[i]))))
106     return mengecewakan
107
108 def tblOK(sedang, baik, tidak_enak, biasa):
109     ok = []
110
111     for i in range(0,100):
112         ok.append(max(min(sedang[i], biasa[i]), min(baik[i], tidak_enak[i]))))
113     return ok
114
115 def tblPuas(sedang, baik, biasa, enak):
116     puas = []
117
118     for i in range(0,100):
119         puas.append(max(min(sedang[i], enak[i]), min(baik[i], biasa[i]), min(baik[i], enak[i]))))
120     return puas
121

```

Penerapan di Main Program (Driver) (sama dengan 1c)

```
#Inference
mengecewakan = tblMengecewakan(buruk, sedang, tidak_enak, biasa, enak)
ok = tblOK(sedang, baik, tidak_enak, biasa)
puas = tblPuas(sedang, baik, biasa, enak)
```

d. Defuzzifikasi

Function/Prosedur (Sama dengan 1e)

```
123 def defuzzSugeno(mengecewakan, ok, puas):
124     defuzz = []
125
126     for i in range(0,100):
127         defuzz.append((mengecewakan[i]*50 + ok[i]*70 + puas[i]*100)/(mengecewakan[i] + ok[i] + puas[i]))
128     return defuzz
129
```

Di Main Program (Driver) (sama dengan 1d)

```
#Defuzzification
defuzz = defuzzSugeno(mengecewakan, ok, puas)
```

e. Function Lain (Mendapatkan top 10)

Function/Prosedur

```
131 def isiTblAkhir(defuzz):
132     tblAkhir = []
133
134     for i in range(0,100):
135         tampung = []
136         tampung.append(i+1)
137         tampung.append(defuzz[i])
138         tblAkhir.append(tampung)
139     return tblAkhir
140
141 def selectionSort(tblAkhir):
142     for i in range(0,100):
143         maks = i
144         for j in range(i+1,100):
145             if (tblAkhir[j][1] > tblAkhir[maks][1]):
146                 maks = j
147             tblAkhir[maks], tblAkhir[i] = tblAkhir[i], tblAkhir[maks]
148
149 def top10(tblAkhir):
150     tertinggi_10 = []
151     for i in range(0,10):
152         tertinggi_10.append(tblAkhir[i][0])
153     return tertinggi_10
154
```


Di Main Program (Driver)

```
179 tblAkhir = isiTblAkhir(defuzz)
180 selectionSort(tblAkhir)
181 tertinggi_10 = top10(tblAkhir)
```

f. Export ke Excel

Di Main Program (Driver)

```
183 #Export to xlsx
184 df = pd.DataFrame(tertinggi_10)
185 df.to_excel(r'peringkat.xlsx', index = False)
```

3. Kesimpulan

Jadi, dapat disimpulkan bahwa fuzzy logic dapat menyelesaikan permasalahan yang memiliki nilai *fuzzy* (tidak pasti) dengan menggunakan Reasoning yang memiliki pendekatan yang mirip dengan perasaan manusia sehingga menghasilkan nilai luaran yang lebih manusiawi.

Hasil Output :

```
PS D:\Kuliah Online\Semester 6\Pengantar Ai\Tugas Pemrograman 2> python 18_if4305_1301180469.py
PS D:\Kuliah Online\Semester 6\Pengantar Ai\Tugas Pemrograman 2> █
```

File peringkat.xlsx

	A
1	0
2	6
3	16
4	15
5	22
6	24
7	42
8	25
9	31
10	54
11	60

4. Video Presentasi

<https://drive.google.com/file/d/1PG9sHg4BqUaPysBnjusfmOiAvgfuseic/view?usp=sharing>