

TUGAS PEMROGRAMAN 2
PENGANTAR KECERDASAN BUATAN
FUZZY LOGIC



Disusun oleh:

1301218548 – Irgi Ahmad Maulana

1301218586 - Muhamad Azmi Rizkifar

FAKULTAS INFORMATIKA
PROGRAM STUDI S1 INFORMATIKA
UNIVERSITAS TELKOM

2021/2022

Pada tugas pemrograman 2 Pengantar Kecerdasan Buatan ini kami membangun sebuah sistem berbasis fuzzy untuk memilih 10 bengkel terbaik yang ada di kota Bandung berdasarkan data yang tersedia di LMS dengan nama “bengkel.xlsx” yang menghimpun 100 data bengkel dengan isian atribut **servis/pelayanan (bilangan real 1-100)** dan **harga (bilangan real 1-10)**.

Berikut adalah beberapa hal yang kami observasi dari sistem fuzzy yang telah dibangun :

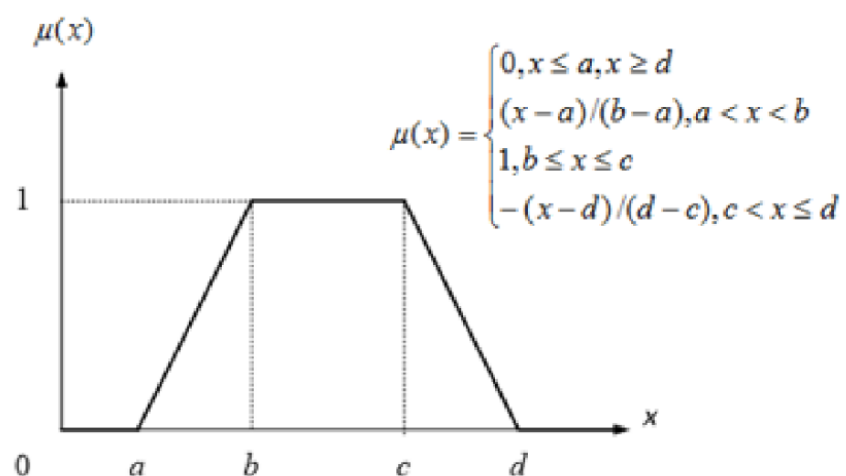
1. Jumlah dan Nama Linguistik setiap atribut input

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

Pelayanan		Harga	
Nilai	Interval	Nilai	Interval
buruk	1-40	murah	1-4
standar	35-65	standar	3-8
bagus	60-80	mahal	6-9
sangat bagus	75-100	sangat mahal	8-10

2. Bentuk dan Batas Fungsi Keanggotaan Input

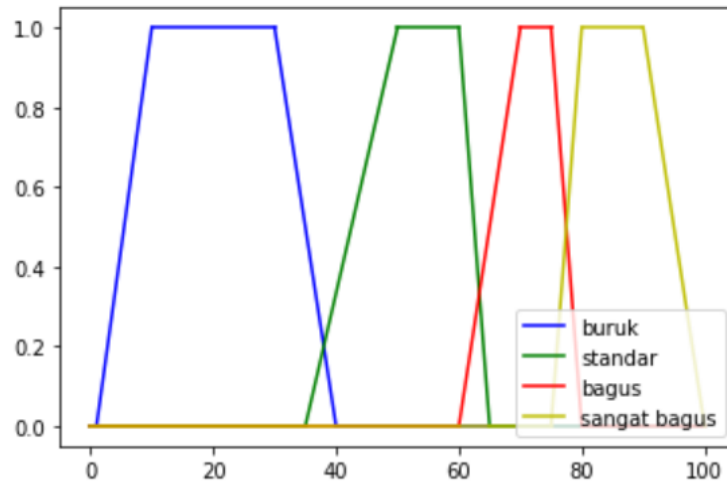
Kami menggunakan array dengan length sebanyak 4 untuk fungsi keanggotaan pada sistem fuzzy ini yang dimana setiap indexnya merepresentasikan bentuk kurva linear yang akan membentuk sebuah trapesium. Bentuk dari kurva menyerupai gambar dibawah ini :



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

'buruk' : [1, 10, 30, 40]
 'standar' : [35, 50, 60, 65]
 'bagus' : [60, 70, 75, 80]
 'sangat bagus' : [75, 80, 90, 100]

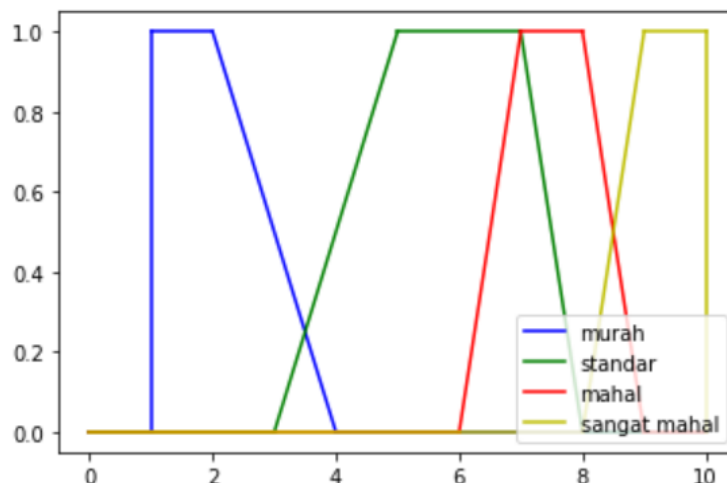
Hasil kurva yang terbentuk :



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

'murah' : [1, 1, 2, 4]
 'standar' : [3, 5, 7, 8]
 'mahal' : [6, 7, 8, 9]
 'sangat mahal' : [8, 9, 10, 10]

Hasil kurva yang terbentuk :



3. Aturan Inferensi

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 kami membuat tiga jenis kesimpulan untuk menyatakan inferensi antara dua nilai linguistik yaitu **accept**, **considered**, **reject** korelasi data yang dibuat menjadi seperti dibawah ini:

	murah	standar	mahal	sangat mahal
buruk	reject	reject	reject	reject
standar	accept	consider	reject	reject
bagus	accept	accept	consider	consider
sangat bagus	accept	accept	accept	consider

4. Metode Defuzzifikasi

Metode constant defuzzification (Takagi-Sugeno-Style) kami 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 :

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

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

Batas fungsi keanggotaan yang digunakan pada kasus ini yaitu **accept : 100**, **consider: 60**, **reject : 10**.

Berikut proses membangun sistem fuzzy :

1. Pembacaan data

Kami melakukan proses pembacaan data dari file yang disediakan di LMS ke dalam directory google collab yang nantinya akan dijadikan sebagai data untuk menghitung nilai fuzzy dan mendapatkan 10 data terbaik berdasarkan perhitungannya.

```
dataFrame = pd.read_excel("bengkel.xlsx")
dataFrame.head(10)
```

	id	servis	harga
0	1	58	7
1	2	54	1
2	3	98	2
3	4	52	4
4	5	11	4
5	6	59	10
6	7	61	8
7	8	30	10
8	9	45	1
9	10	36	9

2. Fuzzifikasi

Fuzzifikasi merupakan proses yang dilakukan untuk mengubah suatu makna dari bentuk crisp menjadi variabel-variabel linguistik yang membentuk himpunan fuzzy dengan fungsi keanggotaannya masing-masing.

Hal yang dilakukan dalam membangun proses fuzzifikasi diantaranya adalah:

- Membuat fungsi linguistik yang menerima inputan berupa nilai yang akan di fuzzifikasi (x) dan list berisi 4 nilai dari representasi linguistik yang memberikan output nilai fuzzifikasi dalam bilangan real.

```
def fuzzyLinguistik(x, point):
    # jika point berada didepan dan dibelakang grafik
    if x < point[0] or x > point[3]:
        return 0

    # jika point berada ditengah grafik
    elif x ≥ point[1] and x ≤ point[2]:
        return 1

    # jika point berada di 1/4 grafik kiri
    elif x ≥ point[0] and x < point[1]:
        return (x - point[0]) / (point[1] - point[0])

    # jika point berada di 1/4 grafik kanan
    elif x ≥ point[2] and x < point[3]:
        return (point[3] - x) / (point[3] - point[2])

    return 0
```

- b. Membuat fungsi keanggotaan yang menerima inputan berupa nilai input yang akan di fuzzifikasi (x) dan dictionary yang berisi kumpulan linguistik dan nilai linguistiknya yang memberikan output dictionary berupa semua linguistik anggota dan nilainya.

```
def fuzzyMembership(x, keanggotaan):
    result = {}

    # Mencari nilai fuzzy untuk masing-masing linguistik dalam Membership
    for ling in keanggotaan:
        result[ling] = fuzzyLinguistik(x, keanggotaan[ling])
        print(result)

    return result
```

- c. Membuat fungsi fuzzifikasi yang menghitung seluruh nilai fuzzifikasi dari x terhadap seluruh linguistik yang ada dan memberikan output dictionary dan hasil fuzzifikasi seluruh anggota.

```
def fuzzyfication(x, list_keanggotaan):
    fuzzyfication = []

    # menghitung nilai fuzzy dari member yang dibuat
    for i in range(len(list_keanggotaan)):
        fuzzyfication.append(fuzzyMembership(x[i], list_keanggotaan[i]))

    return fuzzyfication
```

3. Inferensi

Pada proses ini dilakukan inferensi pada variabel linguistik yang ada sehingga mendapatkan nilai kesimpulan dari variabel linguistik yang digabungkan

```
def inferensi(nilai_fuzzy, rule):
    inferenceData = {}
    # loop nilai linguistik untuk membuat inference
    for l1, v1 in nilai_fuzzy[0].items():
        for l2, v2 in nilai_fuzzy[1].items():

            # r dan min_value merupakan kombinasi dua linguistik
            # dan value minimal dari kedua linguistik
            r, min_value = (l1, l2), min(v1, v2)

            # lihat nilai yang sudah didapat pada hasil
            current_value = inferenceData.get(rule[r], 0)

            # replace nilai hasil apabila current value lebih kecil
            inferenceData[rule[r]] = max(min_value, current_value)

    return inferenceData
```

4. Defuzzifikasi

Pada proses ini dilakukan defuzzifikasi dengan menggunakan metode constant defuzzification (Takagi-Sugeno-Style) yang akan memberikan output nilai fuzzy yang nantinya dapat digunakan untuk menentukan 10 bengkel terbaik yang ada di kota Bandung.

```
def sugenoDefuzzification(x_infer, membership):
    num, den = 0, 0

    # rumus defuzzification model sugeno
    for ling in membership:
        num += x_infer[ling] * membership[ling]
        den += x_infer[ling]

    if den > 0 :
        return num / den
```

Kesimpulan

Dari 100 data yang tersedia, didapatkan 10 data dengan korelasi terbaik yang diurutkan menurun (descending) berdasarkan skor dari hasil proses fuzzy logic, berikut terlampir 10 data bengkel terbaik :

	id	servis	harga	Skor	Infer
55	56	49	3	100.0	{'reject': 0, 'accept': 0.5, 'consider': 0}
20	21	48	2	100.0	{'reject': 0, 'accept': 0.8666666666666667, 'c...
18	19	42	3	100.0	{'reject': 0, 'accept': 0.4666666666666667, 'c...
27	28	44	3	100.0	{'reject': 0, 'accept': 0.5, 'consider': 0}
16	17	70	3	100.0	{'reject': 0, 'accept': 0.5, 'consider': 0}
15	16	82	6	100.0	{'reject': 0, 'accept': 1, 'consider': 0}
14	15	78	5	100.0	{'reject': 0, 'accept': 0.6, 'consider': 0}
31	32	42	3	100.0	{'reject': 0, 'accept': 0.4666666666666667, 'c...
12	13	80	3	100.0	{'reject': 0, 'accept': 0.5, 'consider': 0}
51	52	94	3	100.0	{'reject': 0, 'accept': 0.5, 'consider': 0}

Plot data pelayanan dan harga antara 10 bengkel terbaik dan 90 bengkel lainnya.

