

**Laporan Tugas Pemrograman 01 – Searching02 –
Reasoning CII-2M3 Pengantar Kecerdasan Buatan
Semester Genap 2021/2022**

Disusun Oleh:

Akmal Muhammad Firdaus-1301204188

Andre Eka Putra Simanjuntak-1301204173



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2022**

1. Pendahuluan

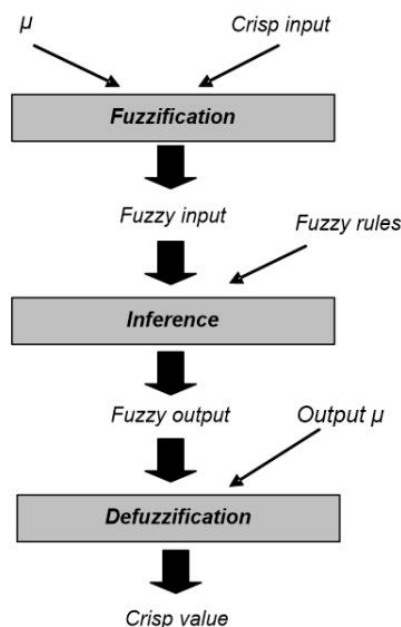
Fuzzy Logic merupakan salah satu algoritma yang ada di dalam kecerdasan buatan. Algoritma ini merupakan salah satu konsep dari algoritma *Reasoning Logic* dimana akan mengembalikan nilai derajat kebenaran dari input-input yang tidak pasti. Hal ini biasa diterapkan untuk menyelesaikan masalah-masalah yang sifatnya kualitatif. Dalam penyelesaian algoritma ini ada beberapa hal yang perlu diobservasi seperti:

- Jumlah dan Nama Linguistik setiap input.
- Bentuk dan Batas Fungsi Keanggotaan Input.
- Metode *Fuzzification*.
- Metode *Inference*.
- Metode *Defuzzification*.
- Bentuk dan Batas Fungsi Keanggotaan Output.

Hal-hal tersebut yang menjadi faktor sehingga mempengaruhi hasil akhir dari pemeringkatan yang akan dilakukan oleh algoritma Fuzzy Logic ini.

2. Alur dan Rancangan Program

Fuzzy logic merupakan suatu cara untuk memetakan suatu ruang input ke dalam ruang output. Secara umum, *fuzzy logic* cocok untuk menyelesaikan masalah-masalah yang sulit didefinisikan dengan model matematis atau masalah yang mengandung ketidakpastian. Keuntungan dari *fuzzy logic* yaitu sistem ini mempunyai kemampuan dalam penalaran yang sama dengan penalaran manusia karena sistem ini punya kemampuan untuk memberikan respon berdasarkan informasi yang kualitatif, tidak akurat, dan ambigu.



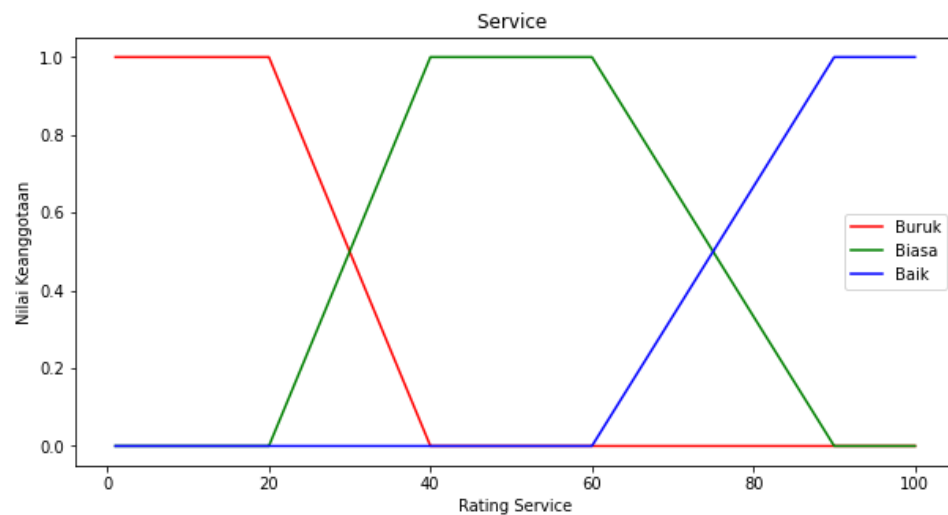
Gambar 1.1 (Flow Fuzzy Logic)

Dapat dilihat pada Gambar 1.1, tahapan pada *fuzzy logic* ada tiga, yaitu *fuzzification*, *inference*, dan *defuzzification*. Pada *fuzzification* terdapat inputan berupa μ dan *crisp input*, kemudian *crisp input* diubah menjadi *fuzzy input* yang berupa nilai

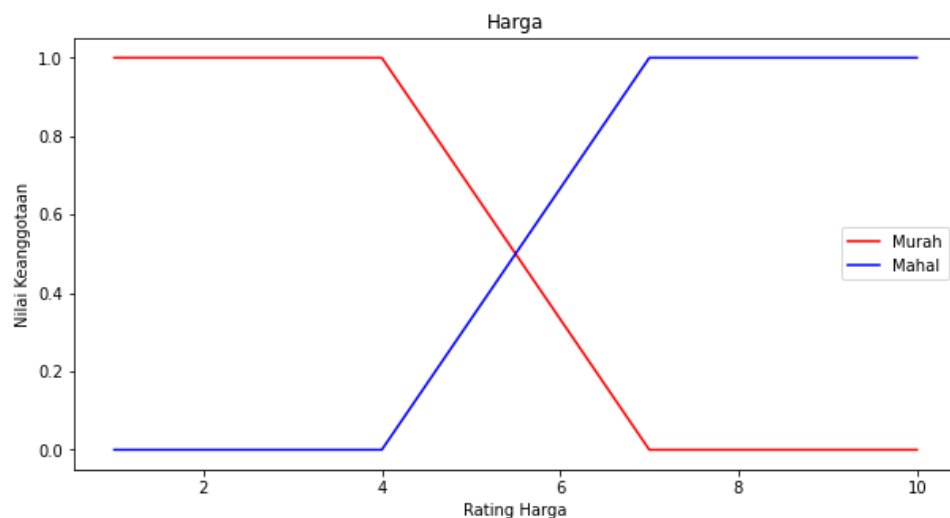
linguistik. Pada *inference* kita melakukan penalaran menggunakan *fuzzy input* dan *fuzzy rules* yang telah ditentukan sehingga menghasilkan *fuzzy output*. Langkah yang terakhir adalah *defuzzification* yaitu perubahan *fuzzy output* menjadi *crisp value* berupa nilai kelayakan yang berdasar pada fungsi keanggotaan yang telah ditentukan.

2.1. Fuzzification

Fuzzification merupakan tahap pertama dari algoritma *Fuzzy Logic*. Pada tahap ini, kami merubah nilai crisp input dengan menggunakan μ menjadi fuzzy input yang berupa nilai linguistik. Pada persoalan yang telah ditentukan, terdapat dua atribut yaitu servis bengkel dengan skala rating bilangan real 1-100 dan harga bengkel dengan skala rating bilangan real 1-10. Pada tahap ini, kami menentukan sendiri bentuk dan batasan pada fungsi kualitas makanan dan kualitas pelayanan.



Gambar 2.1.1 (Grafik Fungsi Keanggotaan Servis Bengkel)



Gambar 2.1.2 (Grafik Fungsi Keanggotaan Harga Bengkel)

Pada Gambar 2.1 merupakan grafik fungsi keanggotaan dari servis bengkel. Kami menentukan untuk menggunakan 3 fungsi trapesium dengan batasan servis "Buruk" pada skala bilangan real 1 sampai 20, servis "Biasa"

dengan skala bilangan real 40 sampai 60, dan servis "Baik" dengan skala bilangan real 90 sampai 100. Pada Gambar 2.1 merupakan grafik fungsi keanggotaan dari harga servis bengkel. Kami menentukan untuk menggunakan 2 fungsi trapesium dengan batasan harga "Murah" pada skala bilangan real 1 sampai 4 dan servis "Mahal" dengan skala bilangan real 7 sampai 10. Sehingga dari grafik tersebut dapat kami tentukan rumus untuk mencari nilai μ dari masing masing data sebagai berikut :

$$\mu_{Buruk} = \begin{cases} -1, & x \geq 40 \\ 1, & x \leq 20 \\ \frac{40 - x}{40 - 20}, & 20 < x < 40 \end{cases}$$

$$\mu_{Biasa} = \begin{cases} -1, & x \leq 20, x \geq 90 \\ 1, & 40 \leq x \leq 60 \\ \frac{x - 20}{40 - 20}, & 20 < x < 40 \\ \frac{90 - x}{90 - 60}, & 60 < x < 90 \end{cases}$$

$$\mu_{Baik} = \begin{cases} -1, & x \leq 60 \\ 1, & x \geq 90 \\ \frac{x - 60}{90 - 60}, & 60 < x < 90 \end{cases}$$

Gambar 2.1.3 (Rumus mencari μ servis)

$$\mu_{Murah} = \begin{cases} -1, & x \geq 7 \\ 1, & x \leq 4 \\ \frac{7 - x}{7 - 4}, & 4 < x < 7 \end{cases}$$

$$\mu_{Mahal} = \begin{cases} -1, & x \leq 4 \\ 1, & x \geq 7 \\ \frac{4 - x}{7 - 4}, & 4 < x < 7 \end{cases}$$

Gambar 2.1.4 (Rumus mencari μ harga)

```

def fuzzification(data_bengkel):
    for i in range(len(data_bengkel)):
        #Fuzzification terhadap data bengkel bagian servis
        if data_bengkel[i].servis >= 1 and data_bengkel[i].servis <= 20:
            data_bengkel[i].servis_buruk = 1
        elif data_bengkel[i].servis > 20 and data_bengkel[i].servis < 40:
            data_bengkel[i].servis_buruk = (40 - data_bengkel[i].servis)/(40 - 20)
            data_bengkel[i].servis_biasa = (data_bengkel[i].servis - 20)/(40 - 20)
        elif data_bengkel[i].servis >= 40 and data_bengkel[i].servis <= 60:
            data_bengkel[i].servis_biasa = 1
        elif data_bengkel[i].servis > 60 and data_bengkel[i].servis < 90:
            data_bengkel[i].servis_biasa = (90 - data_bengkel[i].servis)/(90 - 60)
            data_bengkel[i].servis_baik = (data_bengkel[i].servis - 60)/(90 - 60)
        elif data_bengkel[i].servis >= 90 and data_bengkel[i].servis <= 100:
            data_bengkel[i].servis_baik = 1

        #Fuzzification terhadap data bengkel bagian harga
        if data_bengkel[i].harga >= 1 and data_bengkel[i].harga <= 4:
            data_bengkel[i].harga_murah = 1
        elif data_bengkel[i].harga > 4 and data_bengkel[i].harga < 7:
            data_bengkel[i].harga_murah = (7 - data_bengkel[i].harga)/(7 - 4)
            data_bengkel[i].harga_mahal = (data_bengkel[i].harga - 4)/(7 - 4)
        elif data_bengkel[i].harga >= 7 and data_bengkel[i].harga <= 10:
            data_bengkel[i].harga_mahal = 1

    return data_bengkel

```

Gambar 2.1.5 (Fungsi yang kami gunakan untuk melakukan fuzzification)

2.2. Inference

Inference merupakan proses lebih lanjut untuk menentukan fuzzy input berdasarkan aturan-aturan *fuzzy* yang telah ditentukan. Pada permasalahan ini, kami menetapkan enam aturan yang digunakan untuk menentukan *fuzzy output* dari biaya dan kualitas servis suatu bengkel.

Harga \ Service	Buruk	Biasa	Baik
Mahal	Buruk	Buruk	Buruk
Murah	Buruk	Bagus	Bagus

Gambar 2.2.1 (Tabel inference)

1. Jika biaya servis mahal, namun kualitas buruk. Maka bengkel tersebut buruk.
2. Jika biaya servis mahal, namun kualitas biasa. Maka bengkel tersebut buruk.
3. Jika biaya servis mahal, namun kualitas baik. Maka bengkel tersebut buruk.
4. Jika biaya servis murah, namun kualitas buruk. Maka bengkel tersebut buruk.
5. Jika biaya servis murah, namun kualitas biasa. Maka bengkel tersebut bagus.
6. Jika biaya servis murah, namun kualitas baik. Maka bengkel tersebut bagus.

```

def inference(data_bengkel):
    for i in range(len(data_bengkel)):
        #Aturan untuk harga mahal
        if data_bengkel[i].harga_mahal != -1:
            if data_bengkel[i].servis_buruk != -1:
                data_bengkel[i].bagus.append(min(data_bengkel[i].harga_mahal, data_bengkel[i].servis_buruk))

            data_bengkel[i].buruk.append(min(data_bengkel[i].harga_mahal, data_bengkel[i].servis_baik))

        #Aturan untuk harga murah
        if data_bengkel[i].harga_murah != -1:
            if data_bengkel[i].servis_buruk != -1:
                data_bengkel[i].buruk.append(min(data_bengkel[i].harga_murah, data_bengkel[i].servis_buruk))

            data_bengkel[i].bagus.append(min(data_bengkel[i].harga_murah, data_bengkel[i].servis_baik))

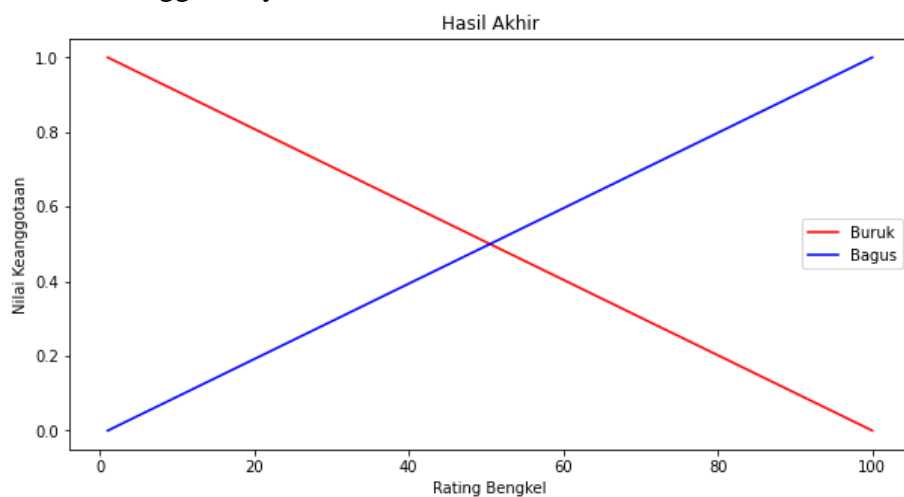
    return data_bengkel

```

Gambar 2.2.2 (Fungsi untuk melakukan inference)

2.3. Defuzzification

Setelah mendapatkan fuzzy output dari tahap inference, tahap selanjutnya merupakan *defuzzification*. *Defuzzification* adalah proses untuk memproduksi sebuah fuzzy output menjadi nilai crisp yang dimana nilai tersebut berdasarkan dari himpunan fuzzy dan derajat keanggotannya.



Gambar 2.3.1 (Grafik Fungsi Keanggotaan Hasil Akhir)

Pada permasalahan ini, kami menggunakan metode Tsukamoto dalam melakukan *defuzzification*. Metode ini kami pilih karena lebih cepat dalam melakukan komputasi. Pada metode ini, kami menetapkan bahwa terdapat dua fungsi derajat keanggotaan, yaitu “bagus” dan “buruk”.

```
def defuzzification(data_bengkel):
    #Menggunakan metode Tsukamoto
    param_buruk = {
        "batas_atas":100,
        "batas_bawah":1
    }

    param_bagus = {
        "batas_atas":100,
        "batas_bawah":1
    }

    for i in range(len(data_bengkel)):
        mu_buruk = 0
        mu_bagus = 0
        if len(data_bengkel[i].buruk) > 0:
            mu_buruk = max(data_bengkel[i].buruk)
        if len(data_bengkel[i].bagus) > 0:
            mu_bagus = max(data_bengkel[i].bagus)

        if mu_buruk == 1 or mu_bagus == 1:
            if mu_buruk == 1:
                data_bengkel[i].rating = param_buruk["batas_bawah"]
            elif mu_bagus == 1:
                data_bengkel[i].rating = param_bagus["batas_atas"]
            else:
                data_bengkel[i].rating = -1
        else:
            x1 = param_buruk["batas_atas"] - (mu_buruk * (param_buruk["batas_atas"] - param_buruk["batas_bawah"]))
            x2 = (mu_bagus * (param_bagus["batas_atas"] - param_bagus["batas_bawah"])) - param_bagus["batas_bawah"]
            rating = ((mu_bagus * x2) + (mu_buruk * x1)) / (mu_buruk + mu_bagus)
            data_bengkel[i].rating = rating

    return data_bengkel
```

Gambar 2.3.2 (Fungsi defuzzification)

3. Output Program

Output dari program kami berbentuk file excel, yang dimana file excel tersebut berisi 10 bengkel teratas dengan skor *defuzzification* tertinggi.

	A	B	C	D
1	ID	Harga Bengkel	Servis Bengkel	Skor Defuz
2	3	2	98	100
3	34	4	93	100
4	52	3	94	100
5	91	3	98	100
6	92	3	83	75,36666667
7	13	3	80	65,66666667
8	15	5	78	59,2
9	8	10	30	50,5
10	86	5	30	50,5
11	96	1	30	50,5

Gambar 3.1 (Ouput program)

4. Kesimpulan

Dengan menggunakan sistem *fuzzy logic*, dapat ditentukan bengkel terbaik dengan melihat 2 atribut setiap bengkel yaitu harga bengkel dan servis bengkel. Program dapat membuat crisp input menjadi fuzzy input, lalu diproses melalui inferensi menjadi fuzzy output dan dikembalikan lagi menjadi crisp value agar dapat dimengerti manusia. Berdasarkan sistem yang telah kami buat, dapat ditentukan 10 bengkel terbaik seperti yang telah disinggung pada bab 3. Jadi dengan *fuzzy logic*, kita dapat mengetahui nilai kelayakan untuk suatu masalah dengan mempertimbangkan beberapa atribut dan dengan penalaran yang lebih manusiawi. *Fuzzy logic* bisa digunakan dalam permasalahan praktis dan bisa menangani ketidakpastian.

Peran anggota :

- Akmal Muhamad Firdaus (1301204188) :
 - o Desain Class Fuzzy Logic
 - o Membuat fungsi fuzzification
 - o Membuat fungsi inference
 - o Membuat laporan
- Andre Eka Putra Simanjuntak(1301204173):
 - o Membuat fungsi Deffuzification
 - o Membuat fungsi output excel
 - o Membuat laporan
 - o Membuat PPT

Link Video Presentasi (Akmal):

<https://youtu.be/yFgoAO-Gwc8>

Link Google Collab:

<https://colab.research.google.com/drive/1keVioAwzcgDI3oCOSOArUddNIExYrNgq?usp=sharing>