

CASE BASED - REASONING

MEMILIH 5 RESTORAN TERBAIK MENGUNAKAN FUZZY LOGIC INFERENCE

Oleh:

Maria Nathasya Desfera Pangestu – 2211104008

Lintang Suminar Tyas Wening – 2211104009

SE 06 01

Kecerdasan Buatan – Fakultas Informatika





DESKRIPSI MASALAH

Dalam tugas Case Based ini, diberikan sebuah file restoran.xlsx berisi 100 data review restoran di kota Bandung. Dengan setiap review memiliki dua atribut yaitu:

- Kualitas Servis (1-100): Semakin tinggi nilai maka semakin baik pelayanan.
- Harga (Rp25.000 - Rp55.000): Semakin tinggi nilai maka semakin mahal.

Dengan permasalahan utama yang harus diselesaikan yaitu membangun sebuah sistem berbasis Fuzzy Logic yang memiliki beberapa poin penting diantaranya:

- Dapat membaca data dari restoran.xlsx.
- Melakukan fuzzification pada atribut kualitas servis dan harga.
- Melakukan inferensi fuzzy untuk menghitung skor kelayakan restoran.
- Melakukan defuzzification untuk menentukan skor akhir.
- Menentukan 5 restoran terbaik berdasarkan skor kelayakan.
- Menyimpan hasil akhir dalam peringkat.xlsx dengan informasi ID restoran, kualitas servis, harga, dan skor kelayakan.



DESAIN DAN ANALISIS SISTEM

Sistem ini dirancang untuk menentukan 5 restoran terbaik di kota Bandung berdasarkan kualitas servis dan harga, menggunakan pendekatan Fuzzy Logic. Data berasal dari restoran.xlsx, dan hasil akhir disimpan dalam peringkat.xlsx. Poin-poin yang harus ada berupa:

1. Kategori Linguistik Atribut Input

Sistem ini menggunakan dua variabel input, masing-masing dengan tiga kategori linguistik yaitu:

- Kualitas Servis: Buruk, Sedang, Bagus
- Harga: Murah, Sedang, Mahal

Kategori ini digunakan dalam proses fuzzifikasi untuk menangkap variasi data lebih fleksibel.



DESAIN DAN ANALISIS SISTEM

2. Bentuk dan Batas Fungsi Keanggotaan Input

Fungsi keanggotaan menggunakan segitiga (triangular membership function) karena sederhana dan efektif:

- Kualitas Servis (Skala 0 - 100)
 - Buruk: 0 - 50
 - Sedang: 40 - 70
 - Bagus: 60 - 100
- Harga (Skala Rp30.000 - Rp70.000)
 - Murah: 0 - 30.000
 - Sedang: 25.000 - 45.000
 - Mahal: 40.000 - 70.000

Batas ini dibuat agar terjadi overlap antar kategori, memungkinkan satu nilai input memiliki derajat keanggotaan ganda, sesuai prinsip fuzzy logic.



Penerapan untuk bagian kualitas servis pada program adalah pada bagian berikut:

```
# Fungsi Membership

def membership_servis(servis):
    """
    Menghitung derajat keanggotaan untuk kategori servis (buruk, sedang, bagus).
    Menggunakan bentuk segitiga untuk keanggotaan fuzzy.
    """

    # Buruk: 0-50 (semakin kecil semakin buruk)
    if servis <= 25:
        buruk = 1
    elif 25 < servis <= 50:
        buruk = (50 - servis) / 25
    else:
        buruk = 0

    # Sedang: 40-70 (naik dari 40 ke 55, turun dari 55 ke 70)
    if 40 < servis <= 55:
        sedang = (servis - 40) / 15
    elif 55 < servis <= 70:
        sedang = (70 - servis) / 15
    else:
        sedang = 0

    # Bagus: 60-100 (naik dari 60 ke 80, lalu penuh)
    if 60 < servis <= 80:
        bagus = (servis - 60) / 20
    elif 80 < servis <= 100:
        bagus = 1
    else:
        bagus = 0

    return {'buruk': max(min(buruk, 1), 0),
            'sedang': max(min(sedang, 1), 0),
            'bagus': max(min(bagus, 1), 0)}
```



Penerapan untuk bagian harga pada program adalah pada bagian berikut:

```
def membership_harga(harga):  
    """  
    Menghitung derajat keanggotaan untuk kategori harga (murah, sedang, mahal).  
    Menggunakan fungsi segitiga.  
  
    Batas harga:  
    - Murah: 0 - 30,000  
    - Sedang: 25,000 - 45,000  
    - Mahal: 40,000 - 70,000  
    """  
    if harga <= 20000:  
        murah = 1  
    elif 20000 < harga <= 30000:  
        murah = (30000 - harga) / 10000  
    else:  
        murah = 0  
  
    if 25000 < harga <= 35000:  
        sedang = (harga - 25000) / 10000  
    elif 35000 < harga <= 45000:  
        sedang = (45000 - harga) / 10000  
    else:  
        sedang = 0  
  
    if 40000 < harga <= 55000:  
        mahal = (harga - 40000) / 15000  
    elif 55000 < harga <= 70000:  
        mahal = 1  
    else:  
        mahal = 0  
  
    return {'murah': max(min(murah, 1), 0),  
            'sedang': max(min(sedang, 1), 0),  
            'mahal': max(min(mahal, 1), 0)}
```



DESAIN DAN ANALISIS SISTEM

3. Aturan Inferensi

Proses inferensi dilakukan menggunakan Fuzzy Inference System (FIS) berbasis pendekatan Mamdani, dengan aturan IF-THEN untuk menghubungkan input fuzzy (Kualitas Servis dan Harga) dengan kategori output kualitas.

Inferensi dilakukan dengan operator fuzzy AND, yang diimplementasikan menggunakan fungsi $\min()$, sehingga nilai keanggotaan output ditentukan berdasarkan nilai minimum dari input yang relevan.

Sistem ini menekankan kualitas servis sementara harga tetap diperhitungkan membentuk logika inferensi yang menjadi jembatan antara input fuzzy dan output fuzzy, sebelum tahap defuzzifikasi untuk menentukan skor akhir kelayakan restoran.



DESAIN DAN ANALISIS SISTEM

3. Aturan Inferensi

Sistem inferensi dibangun dari kombinasi kondisi fuzzy dari dua atribut input dan menghasilkan 9 aturan fuzzy, yaitu:

- IF Kualitas Servis Bagus AND Harga Murah THEN Sangat Baik
- IF Kualitas Servis Bagus AND Harga Sedang THEN Biasa Saja
- IF Kualitas Servis Bagus AND Harga Mahal THEN Biasa Saja
- IF Kualitas Servis Sedang AND Harga Murah THEN Biasa Saja
- IF Kualitas Servis Sedang AND Harga Sedang THEN Biasa Saja
- IF Kualitas Servis Sedang AND Harga Mahal THEN Buruk
- IF Kualitas Servis Buruk AND Harga Murah THEN Buruk
- IF Kualitas Servis Buruk AND Harga Sedang THEN Buruk
- IF Kualitas Servis Buruk AND Harga Mahal THEN Buruk



Penerapannya pada program adalah pada bagian berikut:

```
# Fungsi Inferensi

def inferensi(mb_servis, mb_harga):
    rules = []

    rules.append(('sangat_baik', min(mb_servis['bagus'], mb_harga['murah'])))
    rules.append(('biasa_saja', min(mb_servis['bagus'], mb_harga['sedang'])))
    rules.append(('biasa_saja', min(mb_servis['bagus'], mb_harga['mahal'])))

    rules.append(('biasa_saja', min(mb_servis['sedang'], mb_harga['murah'])))
    rules.append(('biasa_saja', min(mb_servis['sedang'], mb_harga['sedang'])))
    rules.append(('buruk', min(mb_servis['sedang'], mb_harga['mahal'])))

    rules.append(('buruk', min(mb_servis['buruk'], mb_harga['murah'])))
    rules.append(('buruk', min(mb_servis['buruk'], mb_harga['sedang'])))
    rules.append(('buruk', min(mb_servis['buruk'], mb_harga['mahal'])))

    hasil = {'buruk': 0, 'biasa_saja': 0, 'sangat_baik': 0}
    for kategori, nilai in rules:
        if nilai > hasil[kategori]:
            hasil[kategori] = nilai
    return hasil
```



DESAIN DAN ANALISIS SISTEM

4. Metode Defuzzification

Pada bagian ini dalam sistem logika fuzzy, metode yang diterapkan adalah Metode Rata-Rata Tertimbang (Weighted Average Method), atau disebut juga sebagai Centroid Method dalam konteks defuzzifikasi Mamdani.

Tujuan dari defuzzifikasi adalah untuk mengubah hasil akhir dari proses inferensi fuzzy yang berupa nilai keanggotaan fuzzy pada beberapa kategori output, menjadi nilai numerik yang bersifat crisp (tegas), agar dapat digunakan untuk pengurutan atau pengambilan keputusan, seperti menentukan peringkat restoran.

Setiap kategori output diberikan bobot:

- Buruk \rightarrow 25
- Biasa Saja \rightarrow 50
- Sangat Baik \rightarrow 85

Skor akhir dihitung sebagai rata-rata terbobot berdasarkan derajat keanggotaan setiap restoran. Metode ini dipilih karena sederhana, sesuai dengan konsep fuzzy, dan memberikan hasil yang konsisten untuk proses seleksi dan perankingan restoran.



Penerapannya pada program adalah pada bagian berikut:

```
# Fungsi Defuzzifikasi

def defuzzifikasi(hasil_inferensi):
    """
    Melakukan proses defuzzifikasi menggunakan metode rata-rata terbobot (weighted average).

    Mapping nilai (mengikuti label pada inferensi):
    - 'buruk' : 25
    - 'biasa_saja' : 50
    - 'sangat_baik' : 85
    """
    bobot = {
        'buruk': 25,
        'biasa_saja': 50,
        'sangat_baik': 85
    }

    nilai_total = (hasil_inferensi['buruk'] * bobot['buruk'] +
                   hasil_inferensi['biasa_saja'] * bobot['biasa_saja'] +
                   hasil_inferensi['sangat_baik'] * bobot['sangat_baik'])

    total_keanggotaan = (hasil_inferensi['buruk'] +
                         hasil_inferensi['biasa_saja'] +
                         hasil_inferensi['sangat_baik'])

    if total_keanggotaan == 0:
        return 0

    return nilai_total / total_keanggotaan
```



DESAIN DAN ANALISIS SISTEM

5. Bentuk dan Batas Fungsi Keanggotaan Output

Output dari sistem ini adalah Skor Kelayakan restoran, yang memiliki rentang nilai dari 0 hingga 100, dan dikategorikan ke dalam tiga tingkat kualitas yaitu:

- Buruk \rightarrow Skor < 35
- Biasa Saja \rightarrow Skor $35 - 64$
- Sangat Baik \rightarrow Skor ≥ 65

Pembagian ini dilakukan agar interpretasi hasil lebih jelas dan intuitif, memudahkan proses seleksi restoran terbaik.



Penerapannya pada program adalah pada bagian berikut:

```
# Fungsi Menentukan Keterangan Kualitas

def keterangan_kualitas(skor):
    if skor < 35:
        return "Buruk"
    elif skor < 65:
        return "Biasa Saja"
    else:
        return "Sangat Baik"
```



Penerapan kode untuk membaca data dari file restoran.xlsx serta penerapan kode untuk menyimpan hasil akhir ke dalam file peringkat.xlsx, yaitu sebagai berikut:

```
# Membaca data dari restoran.xlsx

def baca_data(filename):
    wb = openpyxl.load_workbook(filename)
    ws = wb.active
    data = []
    for row in ws.iter_rows(min_row=2, values_only=True):
        id_restoran, servis, harga = row
        data.append({'id': id_restoran, 'pelayanan': servis, 'harga': harga})
    return data

# Menyimpan output ke peringkat.xlsx

def simpan_output(filename, hasil):
    wb = openpyxl.Workbook()
    ws = wb.active
    ws.append(['ID Restoran', 'Pelayanan', 'Harga', 'Skor Kelayakan', 'Keterangan Kualitas'])
    for item in hasil:
        ws.append([item['id'], item['pelayanan'], item['harga'], item['skor'], item['keterangan']])
    wb.save(filename)
```



Penerapan kode main program (inti program) dimana semua proses dijalankan mulai dari membaca data hingga menghasilkan daftar restoran terbaik.

```
# Main Program

def main():
    data = baca_data('restoran.xlsx')
    hasil = []

    for restoran in data:
        mb_servis = membership_servis(restoran['pelayanan'])
        mb_harga = membership_harga(restoran['harga'])
        hasil_inferensi = inferensi(mb_servis, mb_harga)
        skor = defuzzifikasi(hasil_inferensi)
        ket = keterangan_kualitas(skor)
        hasil.append({'id': restoran['id'], 'pelayanan': restoran['pelayanan'],
                    'harga': restoran['harga'], 'skor': skor, 'keterangan': ket})

    hasil = sorted(hasil, key=lambda x: x['skor'], reverse=True)
    top5 = hasil[:5]

    simpan_output('peringkat.xlsx', top5)

    print("5 Restoran Terbaik:")
    for item in top5:
        print(f"ID: {item['id']}, Pelayanan: {item['pelayanan']}, Harga: {item['harga']}, Skor: {item['skor']:.2f}, Kualitas: {item['keterangan']}")

if __name__ == "__main__":
    main()
```

HASIL OUTPUT

```
PS D:\SEMESTER 6\Kecerdasan Buatan\Fuzzy> python fuzzy_restoran.py
5 Restoran Terbaik:
ID: 23, Pelayanan: 77, Harga: 22825, Skor: 85.00, Kualitas: Sangat Baik
ID: 33, Pelayanan: 73, Harga: 24704, Skor: 85.00, Kualitas: Sangat Baik
ID: 66, Pelayanan: 80, Harga: 20052, Skor: 85.00, Kualitas: Sangat Baik
ID: 69, Pelayanan: 85, Harga: 24551, Skor: 85.00, Kualitas: Sangat Baik
ID: 79, Pelayanan: 92, Harga: 22360, Skor: 85.00, Kualitas: Sangat Baik
PS D:\SEMESTER 6\Kecerdasan Buatan\Fuzzy> 
```

Setelah proses perhitungan sistem fuzzy dan defuzzifikasi, diperoleh daftar 5 restoran terbaik dengan skor kelayakan tertinggi yaitu 85 yang disimpan dalam file peringkat.xlsx. Tabel output mencakup ID restoran, kualitas servis(pelayanan), harga, skor kelayakan, dan kategori kualitas. Hasil ini ditampilkan sesuai akurasi dan konsistensi perhitungan fuzzy melalui terminal dengan menjalankan fuzzy_restoran.py.

TERIMA
KASIH

