

## Tugas Individu 2 - Logika Fuzzy

Nama: Fadlan Dwi Febrio

NPM: G1A022051

Dosen: Dr. Endina Putri Purwandari, S.T, M.Kom.

Deadline: 30 September 2024

Petunjuk:

1. Mahasiswa dengan NPM Ganjil → kerjakan dengan fungsi keanggotaan SEGITIGA
2. Mahasiswa dengan NPM Genap → kerjakan dengan fungsi keanggotaan TRAPESIUM

### FUNGSI KEANGGOTAAN SEGITIGA

#### Soal:

Suatu penelitian dilakukan untuk mencari jumlah produksi berdasarkan pengaruh faktor suhu, kebisingan, dan pencahayaan. Dalam penelitian ini ada 30 pekerja, yang masing-masing melakukan 27 kali percobaan dengan kombinasi suhu ( $^{\circ}\text{C}$ ), kebisingan (dB), dan pencahayaan (lux) yang berbeda untuk menghasilkan sejumlah produk. Banyaknya data diperoleh sejumlah 810 data. Dari ketigapuluh data untuk setiap kombinasi diambil nilai rata-ratanya, sehingga data yang akan diolah tinggal 27 data sebagai berikut :

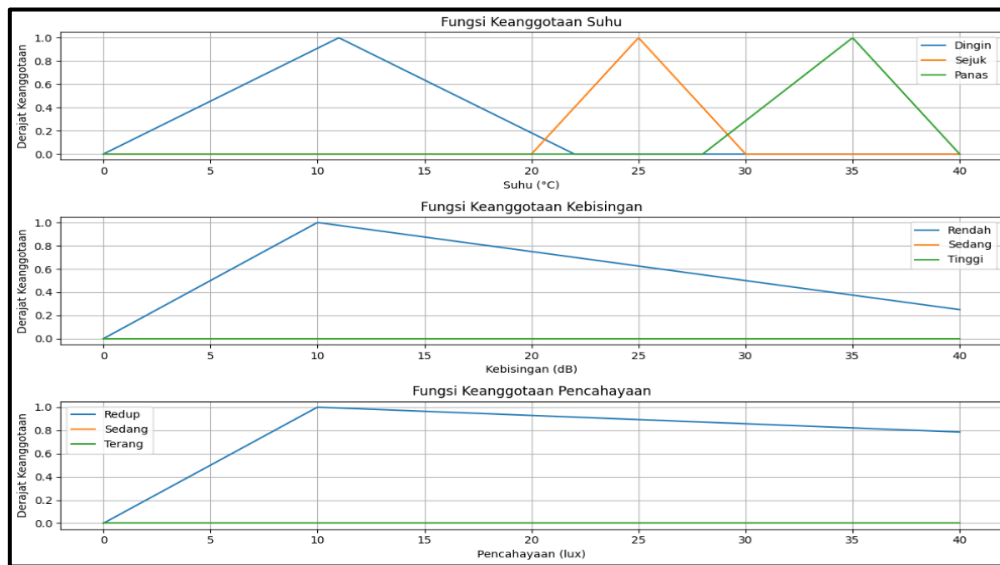
No	Suhu ( $^{\circ}\text{C}$ )	Kebisingan (dB)	Pencahayaan (lux)	Rata-rata jumlah produk	Standar deviasi
1	22	55	150	148,00	4,71
2	22	55	300	150,90	4,78
3	22	55	500	146,50	4,90
4	22	75	150	143,10	4,90
5	22	75	300	146,53	4,58
6	22	75	500	142,73	5,42
7	22	90	150	136,73	4,49
8	22	90	300	140,77	4,49
9	22	90	500	135,97	4,75
10	26	55	150	149,73	4,43
11	26	55	300	153,27	5,59
12	26	55	500	152,13	5,04
13	26	75	150	148,00	5,15
14	26	75	300	150,63	5,06
15	26	75	500	147,63	4,84
16	26	90	150	141,47	5,69
17	26	90	300	145,67	4,81
18	26	90	500	140,20	4,76
19	32	55	150	142,10	4,28
20	32	55	300	146,53	5,38
21	32	55	500	142,17	4,53
22	32	75	150	138,70	4,84
23	32	75	300	141,40	4,95
24	32	75	500	138,30	5,12
25	32	90	150	133,33	4,71
26	32	90	300	138,53	4,51
27	32	90	500	137,77	4,83

Tentukan :

- a. Fungsi Keanggotaan beserta gambarnya
- b. 27 aturan Fuzzy
- c. Derajat keanggotaan nilai tiap variable dalam setiap himpunan
- d.  $\alpha$ -predikat untuk setiap aturan
- e. Rata-rata jumlah produk (gunakan metode defuzzy weighted average)

## PEMBAHASAN

### a. Fungsi keanggotaan beserta gambarnya!



Gambar 1.1 Fungsi Keanggotaan

Penjelasan:

Gambar diatas merupakan gambar dari fungsi keanggotaan segitiga mulai dari gambar fungsi keanggotaan suhu, kebisingan, dan juga pencahayaan, sehingga menghasilkan gambar diatas.

### b. 27 Aturan fuzzy

```
# Aturan Fuzzy: Outputnya (0 = rendah, 1 = sedang, 2 = tinggi)
def aturan_fuzzy(suhu, kebisingan, pencahayaan):
    if suhu == 'dingin' and kebisingan == 'rendah' and pencahayaan == 'redup':
        return 0 # Rendah
    elif suhu == 'dingin' and kebisingan == 'rendah' and pencahayaan == 'sedang':
        return 0
    elif suhu == 'dingin' and kebisingan == 'rendah' and pencahayaan == 'terang':
        return 0
    elif suhu == 'dingin' and kebisingan == 'sedang' and pencahayaan == 'redup':
        return 0
    elif suhu == 'dingin' and kebisingan == 'sedang' and pencahayaan == 'sedang':
        return 1
    elif suhu == 'dingin' and kebisingan == 'sedang' and pencahayaan == 'terang':
        return 1
    elif suhu == 'dingin' and kebisingan == 'tinggi' and pencahayaan == 'redup':
        return 0
    elif suhu == 'dingin' and kebisingan == 'tinggi' and pencahayaan == 'sedang':
        return 1
    elif suhu == 'dingin' and kebisingan == 'tinggi' and pencahayaan == 'terang':
        return 2
    elif suhu == 'sejuk' and kebisingan == 'rendah' and pencahayaan == 'redup':
        return 0
    elif suhu == 'sejuk' and kebisingan == 'rendah' and pencahayaan == 'sedang':
        return 1
    elif suhu == 'sejuk' and kebisingan == 'rendah' and pencahayaan == 'terang':
        return 1
    elif suhu == 'sejuk' and kebisingan == 'sedang' and pencahayaan == 'redup':
        return 1
    elif suhu == 'sejuk' and kebisingan == 'sedang' and pencahayaan == 'sedang':
        return 1
    elif suhu == 'sejuk' and kebisingan == 'sedang' and pencahayaan == 'terang':
        return 2
    elif suhu == 'sejuk' and kebisingan == 'tinggi' and pencahayaan == 'redup':
        return 1
    elif suhu == 'sejuk' and kebisingan == 'tinggi' and pencahayaan == 'sedang':
        return 2
    elif suhu == 'sejuk' and kebisingan == 'tinggi' and pencahayaan == 'terang':
        return 2
    elif suhu == 'panas' and kebisingan == 'rendah' and pencahayaan == 'redup':
        return 1
    elif suhu == 'panas' and kebisingan == 'rendah' and pencahayaan == 'sedang':
        return 1
    elif suhu == 'panas' and kebisingan == 'rendah' and pencahayaan == 'terang':
        return 2
    elif suhu == 'panas' and kebisingan == 'sedang' and pencahayaan == 'redup':
        return 1
    elif suhu == 'panas' and kebisingan == 'sedang' and pencahayaan == 'sedang':
        return 2
    elif suhu == 'panas' and kebisingan == 'sedang' and pencahayaan == 'terang':
        return 2
    elif suhu == 'panas' and kebisingan == 'tinggi' and pencahayaan == 'redup':
        return 2
    elif suhu == 'panas' and kebisingan == 'tinggi' and pencahayaan == 'sedang':
        return 2
    elif suhu == 'panas' and kebisingan == 'tinggi' and pencahayaan == 'terang':
        return 2
    return None # Default jika tidak ada aturan yang sesuai
```

Gambar 1.2 27 aturan fuzzy

Penjelasan:

Pada gambar diatas merupakan gambar dari 27 aturan dari soal diatas, yang memungkinkan sebanyak 27 kemungkinan yang dapat terjadi jika kita mengikuti aturan yang berlalu diatas.

c. Derajat keanggotaan nilai tiap variabel dalam tiap himpunan

```
Derajat Keanggotaan:  
Suhu: {'sejuk': 0.8}  
Kebisingan: {'sedang': 0.6666666666666666}  
Pencahayaannya: {'sedang': 1.0}
```

Gambar 1.3 Derajat Keanggotaan

Penjelasan:

Pada gambar diatas bisa kita lihat bahwa derajat keanggotaan mulai dari suhu yaitu sejuk= 0,8, kebisingan sedang= 0,6, dan pencahayaan terang sedang= 1.0, sehingga menghasilkan penyelesaian dari soal diatas.

d. A-predikat untuk setiap aturan

```
 $\alpha$ -Predikat untuk setiap output:  
Rendah (0): 0  
Sedang (1): 1  
Tinggi (2): 0
```

Gambar 1.4 A-predikat

Penjelasan:

Pada gambar diatas terlihat a-predikat untuk setiap aturan yaitu rendah=0 sedang =1, tinggi= 0 sehingga a predikat ini berhubungan dengan setiap aturan yang ada dan menghasilkan fungsi keanggotaan yang ada.

e. Rata Rata jumlah produk (gunakan metode defuzzy weighted average)

```
Hasil Defuzzifikasi: 1.0
```

Gambar 1.5 Rata rata jumlah produk

Penjelasan:

Pada gambar diatas ditampilkan hasil perhitungan rata rata dari jumlah produk yang ada sehingga output yang dihasilkan adalah 1.0.

### Source Code:

```
import numpy as np
import matplotlib.pyplot as plt
# Fungsi keanggotaan segitiga
def fungsi_segitiga(x, a, b, c):
    if a == b or b == c or a == c: # Pastikan tidak ada pembagian dengan nol
        return np.zeros_like(x) # Mengembalikan derajat keanggotaan nol
    return np.maximum(np.minimum((x - a) / (b - a), (c - x) / (c - b)), 0)
# Fungsi a: Plot fungsi keanggotaan
def plot_fungsi_keanggotaan():
    x = np.linspace(0, 40, 500)
    # Fungsi keanggotaan untuk variabel suhu
    suhu_dingin = fungsi_segitiga(x, 0, 11, 22)
    suhu_sejuk = fungsi_segitiga(x, 20, 25, 30)
    suhu_panas = fungsi_segitiga(x, 28, 35, 40)
    # Fungsi keanggotaan untuk variabel kebisingan
    kebisingan_rendah = fungsi_segitiga(x, 0, 10, 50)
    kebisingan_sedang = fungsi_segitiga(x, 40, 55, 70)
    kebisingan_tinggi = fungsi_segitiga(x, 60, 80, 100)
    # Fungsi keanggotaan untuk variabel pencahayaan
    pencahayaan_redup = fungsi_segitiga(x, 0, 10, 150)
    pencahayaan_sedang = fungsi_segitiga(x, 100, 300, 500)
    pencahayaan_terang = fungsi_segitiga(x, 400, 600, 800)
    # Plot semua fungsi keanggotaan
    plt.figure(figsize=(12, 8))
    # Suhu
    plt.subplot(3, 1, 1)
    plt.plot(x, suhu_dingin, label='Dingin')
    plt.plot(x, suhu_sejuk, label='Sejuk')
    plt.plot(x, suhu_panas, label='Panas')
    plt.title('Fungsi Keanggotaan Suhu')
    plt.xlabel('Suhu (°C)')
    plt.ylabel('Derajat Keanggotaan')
    plt.legend()
```

```

plt.grid(True)
# Kebisingan
plt.subplot(3, 1, 2)
plt.plot(x, kebisingan_rendah, label='Rendah')
plt.plot(x, kebisingan_sedang, label='Sedang')
plt.plot(x, kebisingan_tinggi, label='Tinggi')
plt.title('Fungsi Keanggotaan Kebisingan')
plt.xlabel('Kebisingan (dB)')
plt.ylabel('Derajat Keanggotaan')
plt.legend()
plt.grid(True)
# Pencahayaan
plt.subplot(3, 1, 3)
plt.plot(x, pencahayaan_redup, label='Redup')
plt.plot(x, pencahayaan_sedang, label='Sedang')
plt.plot(x, pencahayaan_terang, label='Terang')
plt.title('Fungsi Keanggotaan Pencahayaan')
plt.xlabel('Pencahayaan (lux)')
plt.ylabel('Derajat Keanggotaan')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# Definisikan variabel fuzzy
def suhu_fuzzy(x):
    dingin = fungsi_segitiga(x, 0, 11, 22)
    sejuk = fungsi_segitiga(x, 20, 25, 30)
    panas = fungsi_segitiga(x, 28, 35, 40)
    return {'dingin': dingin, 'sejuk': sejuk, 'panas': panas}

def kebisingan_fuzzy(x):
    rendah = fungsi_segitiga(x, 0, 10, 50)
    sedang = fungsi_segitiga(x, 40, 55, 70)
    tinggi = fungsi_segitiga(x, 60, 80, 100)
    return {'rendah': rendah, 'sedang': sedang, 'tinggi': tinggi}

```

```

def pencahayaan_fuzzy(x):
    redup = fungsi_segitiga(x, 0, 10, 150)
    sedang = fungsi_segitiga(x, 100, 300, 500)
    terang = fungsi_segitiga(x, 400, 600, 800)
    return {'redup': redup, 'sedang': sedang, 'terang': terang}

# Aturan Fuzzy: Outputnya (0 = rendah, 1 = sedang, 2 = tinggi)
def aturan_fuzzy(suhu, kebisingan, pencahayaan):
    if suhu == 'dingin' and kebisingan == 'rendah' and pencahayaan == 'redup':
        return 0 # Rendah
    elif suhu == 'dingin' and kebisingan == 'rendah' and pencahayaan == 'sedang':
        return 0
    elif suhu == 'dingin' and kebisingan == 'rendah' and pencahayaan == 'terang':
        return 0
    elif suhu == 'dingin' and kebisingan == 'sedang' and pencahayaan == 'redup':
        return 0
    elif suhu == 'dingin' and kebisingan == 'sedang' and pencahayaan == 'sedang':
        return 0
    elif suhu == 'dingin' and kebisingan == 'sedang' and pencahayaan == 'terang':
        return 1 # Sedang
    elif suhu == 'dingin' and kebisingan == 'tinggi' and pencahayaan == 'redup':
        return 1
    elif suhu == 'dingin' and kebisingan == 'tinggi' and pencahayaan == 'sedang':
        return 1
    elif suhu == 'dingin' and kebisingan == 'tinggi' and pencahayaan == 'terang':
        return 2 # Tinggi
    elif suhu == 'sejuk' and kebisingan == 'rendah' and pencahayaan == 'redup':
        return 0
    elif suhu == 'sejuk' and kebisingan == 'rendah' and pencahayaan == 'sedang':
        return 1
    elif suhu == 'sejuk' and kebisingan == 'rendah' and pencahayaan == 'terang':
        return 1
    elif suhu == 'sejuk' and kebisingan == 'sedang' and pencahayaan == 'redup':
        return 1
    elif suhu == 'sejuk' and kebisingan == 'sedang' and pencahayaan == 'sedang':

```

```

        return 1
    elif suhu == 'sejuk' and kebisingan == 'sedang' and pencahayaan == 'terang':
        return 2 # Tinggi
    elif suhu == 'sejuk' and kebisingan == 'tinggi' and pencahayaan == 'redup':
        return 1
    elif suhu == 'sejuk' and kebisingan == 'tinggi' and pencahayaan == 'sedang':
        return 2 # Tinggi
    elif suhu == 'sejuk' and kebisingan == 'tinggi' and pencahayaan == 'terang':
        return 2 # Tinggi
    elif suhu == 'panas' and kebisingan == 'rendah' and pencahayaan == 'redup':
        return 1
    elif suhu == 'panas' and kebisingan == 'rendah' and pencahayaan == 'sedang':
        return 1
    elif suhu == 'panas' and kebisingan == 'rendah' and pencahayaan == 'terang':
        return 2 # Tinggi
    elif suhu == 'panas' and kebisingan == 'sedang' and pencahayaan == 'redup':
        return 1
    elif suhu == 'panas' and kebisingan == 'sedang' and pencahayaan == 'sedang':
        return 2 # Tinggi
    elif suhu == 'panas' and kebisingan == 'sedang' and pencahayaan == 'terang':
        return 2 # Tinggi
    elif suhu == 'panas' and kebisingan == 'tinggi' and pencahayaan == 'redup':
        return 2 # Tinggi
    elif suhu == 'panas' and kebisingan == 'tinggi' and pencahayaan == 'sedang':
        return 2 # Tinggi
    elif suhu == 'panas' and kebisingan == 'tinggi' and pencahayaan == 'terang':
        return 2 # Tinggi
    return None # Default jika tidak ada aturan yang sesuai

# Hitung hasil fuzzy
def hitung_fuzzy(suhu_input, kebisingan_input, pencahayaan_input):
    hasil_suhu = suhu_fuzzy(suhu_input)
    hasil_kebisingan = kebisingan_fuzzy(kebisingan_input)
    hasil_pencahayaan = pencahayaan_fuzzy(pencahayaan_input)
    # Ambil derajat keanggotaan

```

```

derajat_suhu = {k: v for k, v in hasil_suhu.items() if v > 0}
derajat_kebisingan = {k: v for k, v in hasil_kebisingan.items() if v > 0}
derajat_pencahayaan = {k: v for k, v in hasil_pencahayaan.items() if v > 0}
# Menampilkan derajat keanggotaan (Bagian c)
print("Derajat Keanggotaan:")
print(f"Suhu: {derajat_suhu}")
print(f"Kebisingan: {derajat_kebisingan}")
print(f"Pencahayaan: {derajat_pencahayaan}")
# Ambil aturan fuzzy
aturan = []
for suhu in derajat_suhu:
    for kebisingan in derajat_kebisingan:
        for pencahayaan in derajat_pencahayaan:
            output = aturan_fuzzy(suhu, kebisingan, pencahayaan)
            if output is not None:
                aturan.append(output)
# Hitung  $\alpha$ -predikat (Bagian d)
alpha_predikat = {0: 0, 1: 0, 2: 0}
for predikat in aturan:
    alpha_predikat[predikat] += 1
# Menampilkan  $\alpha$ -predikat
print("\n $\alpha$ -Predikat untuk setiap output:")
print(f"Rendah (0): {alpha_predikat[0]}")
print(f"Sedang (1): {alpha_predikat[1]}")
print(f"Tinggi (2): {alpha_predikat[2]}")
# Defuzzifikasi dengan metode rata-rata berbobot
total = sum(alpha_predikat.values())
if total == 0:
    return 0
rata_rata = sum(k * v for k, v in alpha_predikat.items()) / total
return rata_rata
# Fungsi utama
def main():
    plot_fungsi_keanggotaan() # Tampilkan grafik fungsi keanggotaan

```



```

# Input dari pengguna
suhu_input = 24 # Misalnya suhu 24°C
kebisingan_input = 60 # Misalnya kebisingan 60 dB
pencahayaan_input = 300 # Misalnya pencahayaan 300 lux
# Langkah c-e: Hitung hasil fuzzy
hasil = hitung_fuzzy(suhu_input, kebisingan_input, pencahayaan_input)
print(f"\nHasil Defuzzifikasi: {hasil}")
if __name__ == "__main__":
    main()

```

Penjelasan:

Kode di atas mengimplementasikan sistem logika fuzzy untuk menentukan tingkat keanggotaan dari variabel suhu, kebisingan, dan pencahayaan menggunakan fungsi keanggotaan segitiga. Pertama, kode mendefinisikan fungsi untuk menghitung dan memplot fungsi keanggotaan untuk masing-masing variabel. Kemudian, kode mendefinisikan aturan fuzzy yang mengaitkan kombinasi variabel input dengan keluaran tertentu (rendah, sedang, tinggi). Setelah itu, fungsi `hitung_fuzzy` digunakan untuk menghitung derajat keanggotaan dari masing-masing variabel dan menentukan  $\alpha$ -predikat untuk setiap aturan yang memenuhi syarat. Hasil akhir berupa defuzzifikasi yang memberikan nilai keluaran berdasarkan perhitungan rata-rata berbobot dari  $\alpha$ -predikat. Seluruh proses diintegrasikan dalam fungsi utama (`main`) yang memplot grafik dan mencetak hasil keanggotaan dan defuzzifikasi.

Kode ini berfungsi sebagai sistem inferensi fuzzy yang memungkinkan pengguna untuk mendapatkan informasi berdasarkan input yang diberikan, seperti suhu, kebisingan, dan pencahayaan. Setiap variabel input dihitung derajat keanggotaannya menggunakan fungsi keanggotaan segitiga yang disesuaikan dengan nilai batas yang telah ditentukan. Kemudian, melalui serangkaian aturan yang telah didefinisikan, sistem mengidentifikasi output yang sesuai untuk kombinasi input yang diberikan. Dengan mengumpulkan semua output yang relevan, kode ini menghitung  $\alpha$ -predikat yang mencerminkan seberapa banyak setiap kategori (rendah, sedang, tinggi) berkontribusi terhadap hasil akhir. Proses defuzzifikasi kemudian dilakukan untuk menghasilkan nilai tunggal yang merepresentasikan keputusan akhir, menjadikan sistem ini efektif dalam situasi di mana ketidakpastian dan variasi input perlu dikelola dengan cara yang logis dan terstruktur.