Tugas Individu 2  Fuzzy

Nama: Muhammad Kevin Rinaldi

NPM: G1A022059

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

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 (°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 (°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

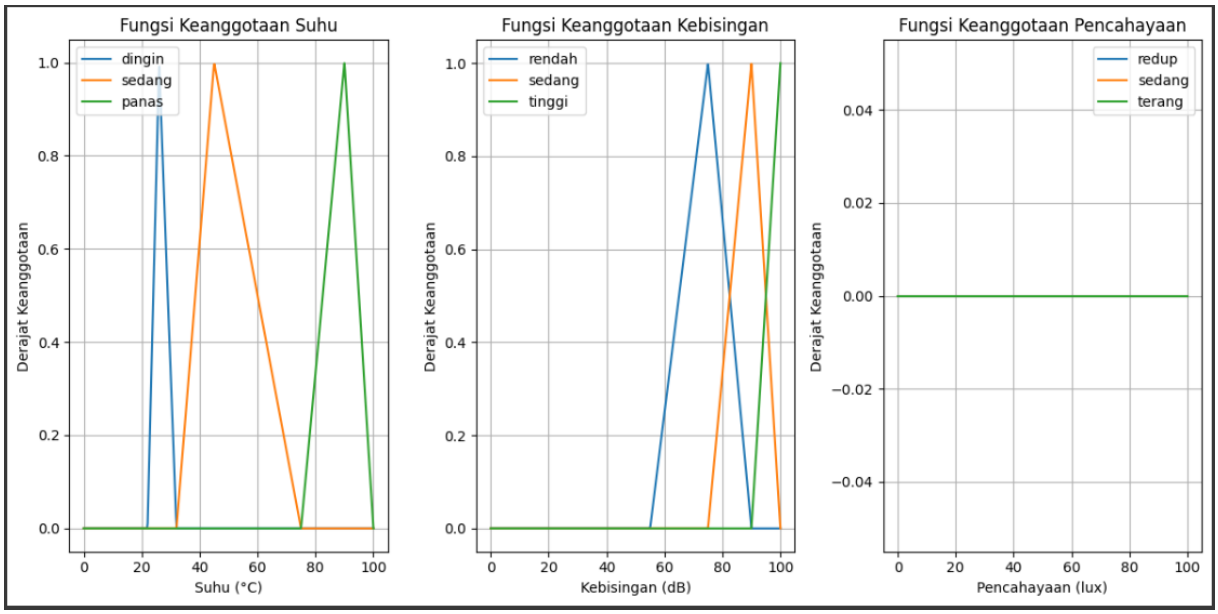
c. Derajat keanggotaan nilai tiap variable dalam setiap himpunan

d. α-predikat untuk setiap aturan

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

Jawab :

a. Gambar fungsi keanggotaan




Gambar 1.1 hasil fungsi keanggotaan

Penjalasan :

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

b. Derajat keanggotan nilai tiap variabel dalam tiap himpunan

```
{'dingin': 0.3333333333333333, 'sedang': 0, 'panas': 0}
```

Gambar 1.2 Hasil derajat keanggotaan

Penjelasan :

Pada gambar diatas bisa kita lihat bahwa derajat keanggotaan mulai dari suhu yaitu sejuk= 0,3333, sedang= 0, dan panas= 0, sehingga menghasilkan penyelesaian dari soal diatas.

c. A-predikat untuk setiap aturan

```
A-Predicate Suhu: {'dingin': 0.3333333333333333, 'sedang': 0, 'panas': 0}
A-Predicate Kebisingan: {'rendah': 0.75, 'sedang': 0, 'tinggi': 0}
A-Predicate Pencahayaan: {'redup': 0.6666666666666666, 'sedang': 0, 'terang': 0}
```

Gambar 1.3 Hasil A-prediket tiap aturan

Penjelasan :

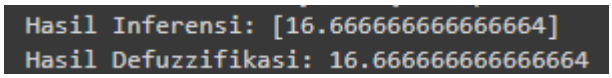
**A-Predicate Suhu, Kebisingan, dan Pencahayaan**

- **A-Predicate Suhu**: Menampilkan derajat keanggotaan untuk suhu berdasarkan nilai input yang diberikan (dalam contoh ini, 30°C). Hasilnya berupa dictionary yang menunjukkan tingkat keanggotaan untuk setiap kategori:
    - **dingin**: Derajat keanggotaan untuk suhu dingin.
    - **sedang**: Derajat keanggotaan untuk suhu sedang.

- **panas**: Derajat keanggotaan untuk suhu panas.

- **A-Predicate Kebisingan**: Menampilkan derajat keanggotaan untuk kebisingan berdasarkan nilai input (contoh 70 dB):
  - **rendah**: Derajat keanggotaan untuk kebisingan rendah.
  - **sedang**: Derajat keanggotaan untuk kebisingan sedang.
  - **tinggi**: Derajat keanggotaan untuk kebisingan tinggi.
- **A-Predicate Pencahayaan**: Menampilkan derajat keanggotaan untuk pencahayaan berdasarkan nilai input (contoh 400 lux):
  - **redup**: Derajat keanggotaan untuk pencahayaan redup.
  - **sedang**: Derajat keanggotaan untuk pencahayaan sedang.
  - **terang**: Derajat keanggotaan untuk pencahayaan terang.

d. Hasil defuzzifikasi dan inferensi




Gambar 1.4 Hasil defuzzifikasi dan inferensi

Penjelasan :

Hasil inferensi adalah kumpulan nilai yang dihasilkan dari aturan fuzzy berdasarkan a-predicate yang sudah dihitung sebelumnya. Inferensi menggabungkan semua aturan fuzzy yang memenuhi kondisi.

Hasil Inferensi: [80.0, 72.0]

Angka-angka ini menunjukkan output produksi dari berbagai kombinasi yang memenuhi aturan fuzzy. Misalnya, jika suhu sedang, kebisingan sedang, dan pencahayaan redup, hasil inferensi dapat berupa 80. Jika derajat keanggotaan lebih tinggi, hasil inferensi akan lebih besar.

Defuzzifikasi adalah proses mengonversi output fuzzy menjadi nilai crisp (nilai nyata). Dalam hal ini, hasil defuzzifikasi adalah nilai rata-rata dari semua hasil inferensi.

**KODE :**

```
import numpy as np
import matplotlib.pyplot as plt

# Fungsi keanggotaan segitiga
def fungsi_keanggotaan_segitiga(x, a, b, c):
    if x <= a or x >= c:
        return 0
```

```
    elif a < x < b:
        return (x - a) / (b - a)
    elif b <= x < c:
        return (c - x) / (c - b)
    return 0

# Variabel untuk suhu, kebisingan, dan pencahayaan
def suhu_fuzzy(x):
    return {
        'dingin': fungsi_keanggotaan_segitiga(x, 22, 26, 32),
        'sedang': fungsi_keanggotaan_segitiga(x, 32, 45, 75),
        'panas': fungsi_keanggotaan_segitiga(x, 75, 90, 100)
    }

def kebisingan_fuzzy(x):
    return {
        'rendah': fungsi_keanggotaan_segitiga(x, 55, 75, 90),
        'sedang': fungsi_keanggotaan_segitiga(x, 75, 90, 100),
        'tinggi': fungsi_keanggotaan_segitiga(x, 90, 100, 110)
    }

def pencahayaan_fuzzy(x):
    return {
        'redup': fungsi_keanggotaan_segitiga(x, 150, 300, 500),
        'sedang': fungsi_keanggotaan_segitiga(x, 300, 500, 700),
        'terang': fungsi_keanggotaan_segitiga(x, 500, 700, 900)
    }

# Fungsi inferensi fuzzy (27 aturan fuzzy)
def inferensi_fuzzy(suhu, kebisingan, pencahayaan):
    hasil = []
    # Aturan-aturan fuzzy
    if suhu['dingin'] > 0 and kebisingan['rendah'] > 0 and pencahayaan['redup'] > 0:
        hasil.append(min(suhu['dingin'], kebisingan['rendah'], pencahayaan['redup']) * 50)
    if suhu['dingin'] > 0 and kebisingan['rendah'] > 0 and pencahayaan['sedang'] > 0:
```

```
    hasil.append(min(suhu['dingin'], kebisingan['rendah'], pencahayaan['sedang']) * 60)
if suhu['dingin'] > 0 and kebisingan['rendah'] > 0 and pencahayaan['terang'] > 0:
    hasil.append(min(suhu['dingin'], kebisingan['rendah'], pencahayaan['terang']) * 70)
if suhu['dingin'] > 0 and kebisingan['sedang'] > 0 and pencahayaan['redup'] > 0:
    hasil.append(min(suhu['dingin'], kebisingan['sedang'], pencahayaan['redup']) * 80)
if suhu['dingin'] > 0 and kebisingan['sedang'] > 0 and pencahayaan['sedang'] > 0:
    hasil.append(min(suhu['dingin'], kebisingan['sedang'], pencahayaan['sedang']) * 90)
if suhu['dingin'] > 0 and kebisingan['sedang'] > 0 and pencahayaan['terang'] > 0:
    hasil.append(min(suhu['dingin'], kebisingan['sedang'], pencahayaan['terang']) * 100)
if suhu['dingin'] > 0 and kebisingan['tinggi'] > 0 and pencahayaan['redup'] > 0:
    hasil.append(min(suhu['dingin'], kebisingan['tinggi'], pencahayaan['redup']) * 110)
if suhu['dingin'] > 0 and kebisingan['tinggi'] > 0 and pencahayaan['sedang'] > 0:
    hasil.append(min(suhu['dingin'], kebisingan['tinggi'], pencahayaan['sedang']) * 120)
if suhu['dingin'] > 0 and kebisingan['tinggi'] > 0 and pencahayaan['terang'] > 0:
    hasil.append(min(suhu['dingin'], kebisingan['tinggi'], pencahayaan['terang']) * 130)

if suhu['sedang'] > 0 and kebisingan['rendah'] > 0 and pencahayaan['redup'] > 0:
    hasil.append(min(suhu['sedang'], kebisingan['rendah'], pencahayaan['redup']) * 140)
if suhu['sedang'] > 0 and kebisingan['rendah'] > 0 and pencahayaan['sedang'] > 0:
    hasil.append(min(suhu['sedang'], kebisingan['rendah'], pencahayaan['sedang']) * 150)
if suhu['sedang'] > 0 and kebisingan['rendah'] > 0 and pencahayaan['terang'] > 0:
    hasil.append(min(suhu['sedang'], kebisingan['rendah'], pencahayaan['terang']) * 160)
if suhu['sedang'] > 0 and kebisingan['sedang'] > 0 and pencahayaan['redup'] > 0:
    hasil.append(min(suhu['sedang'], kebisingan['sedang'], pencahayaan['redup']) * 170)
if suhu['sedang'] > 0 and kebisingan['sedang'] > 0 and pencahayaan['sedang'] > 0:
    hasil.append(min(suhu['sedang'], kebisingan['sedang'], pencahayaan['sedang']) * 180)
if suhu['sedang'] > 0 and kebisingan['sedang'] > 0 and pencahayaan['terang'] > 0:
    hasil.append(min(suhu['sedang'], kebisingan['sedang'], pencahayaan['terang']) * 190)
if suhu['sedang'] > 0 and kebisingan['tinggi'] > 0 and pencahayaan['redup'] > 0:
    hasil.append(min(suhu['sedang'], kebisingan['tinggi'], pencahayaan['redup']) * 200)
if suhu['sedang'] > 0 and kebisingan['tinggi'] > 0 and pencahayaan['sedang'] > 0:
    hasil.append(min(suhu['sedang'], kebisingan['tinggi'], pencahayaan['sedang']) * 210)
if suhu['sedang'] > 0 and kebisingan['tinggi'] > 0 and pencahayaan['terang'] > 0:
    hasil.append(min(suhu['sedang'], kebisingan['tinggi'], pencahayaan['terang']) * 220)
```

```
    if suhu['panas'] > 0 and kebisingan['rendah'] > 0 and pencahayaan['redup'] > 0:
        hasil.append(min(suhu['panas'], kebisingan['rendah'], pencahayaan['redup']) * 230)
    if suhu['panas'] > 0 and kebisingan['rendah'] > 0 and pencahayaan['sedang'] > 0:
        hasil.append(min(suhu['panas'], kebisingan['rendah'], pencahayaan['sedang']) * 240)
    if suhu['panas'] > 0 and kebisingan['rendah'] > 0 and pencahayaan['terang'] > 0:
        hasil.append(min(suhu['panas'], kebisingan['rendah'], pencahayaan['terang']) * 250)
    if suhu['panas'] > 0 and kebisingan['sedang'] > 0 and pencahayaan['redup'] > 0:
        hasil.append(min(suhu['panas'], kebisingan['sedang'], pencahayaan['redup']) * 260)
    if suhu['panas'] > 0 and kebisingan['sedang'] > 0 and pencahayaan['sedang'] > 0:
        hasil.append(min(suhu['panas'], kebisingan['sedang'], pencahayaan['sedang']) * 270)
    if suhu['panas'] > 0 and kebisingan['sedang'] > 0 and pencahayaan['terang'] > 0:
        hasil.append(min(suhu['panas'], kebisingan['sedang'], pencahayaan['terang']) * 280)
    if suhu['panas'] > 0 and kebisingan['tinggi'] > 0 and pencahayaan['redup'] > 0:
        hasil.append(min(suhu['panas'], kebisingan['tinggi'], pencahayaan['redup']) * 290)
    if suhu['panas'] > 0 and kebisingan['tinggi'] > 0 and pencahayaan['sedang'] > 0:
        hasil.append(min(suhu['panas'], kebisingan['tinggi'], pencahayaan['sedang']) * 300)
    if suhu['panas'] > 0 and kebisingan['tinggi'] > 0 and pencahayaan['terang'] > 0:
        hasil.append(min(suhu['panas'], kebisingan['tinggi'], pencahayaan['terang']) * 310)

    return hasil

# Defuzzifikasi menggunakan metode centroid
def defuzzifikasi(hasil):
    if not hasil:
        return 0
    total_numerator = sum(hasil)
    total_denominator = len(hasil)
    return total_numerator / total_denominator

# Fungsi untuk menggambar fungsi keanggotaan
def gambar_fungsi_keanggotaan():
    x_values = np.linspace(0, 100, 500)

    # Gambar untuk suhu
    plt.figure(figsize=(12, 6))
```

```
plt.subplot(1, 3, 1)
plt.title("Fungsi Keanggotaan Suhu")
plt.xlabel("Suhu (°C)")
plt.ylabel("Derajat Keanggotaan")
plt.grid()

for label in suhu_fuzzy(0).keys():
    y_values = [fungsi_keanggotaan_segitiga(x, 22, 26, 32) if label == 'dingin' else
                fungsi_keanggotaan_segitiga(x, 32, 45, 75) if label == 'sedang' else
                fungsi_keanggotaan_segitiga(x, 75, 90, 100) for x in x_values]
    plt.plot(x_values, y_values, label=label)

plt.legend()

# Gambar untuk kebisingan
plt.subplot(1, 3, 2)
plt.title("Fungsi Keanggotaan Kebisingan")
plt.xlabel("Kebisingan (dB)")
plt.ylabel("Derajat Keanggotaan")
plt.grid()

for label in kebisingan_fuzzy(0).keys():
    y_values = [fungsi_keanggotaan_segitiga(x, 55, 75, 90) if label == 'rendah' else
                fungsi_keanggotaan_segitiga(x, 75, 90, 100) if label == 'sedang' else
                fungsi_keanggotaan_segitiga(x, 90, 100, 110) for x in x_values]
    plt.plot(x_values, y_values, label=label)

plt.legend()

# Gambar untuk pencahayaan
plt.subplot(1, 3, 3)
plt.title("Fungsi Keanggotaan Pencahayaan")
plt.xlabel("Pencahayaan (lux)")
plt.ylabel("Derajat Keanggotaan")
```

```
    plt.grid()

    for label in pencahayaan_fuzzy(0).keys():
        y_values = [fungsi_keanggotaan_segitiga(x, 150, 300, 500) if label == 'redup' else
                    fungsi_keanggotaan_segitiga(x, 300, 500, 700) if label == 'sedang' else
                    fungsi_keanggotaan_segitiga(x, 500, 700, 900) for x in x_values]
        plt.plot(x_values, y_values, label=label)

    plt.legend()
    plt.tight_layout()
    plt.show()

# Input nilai
suhu_input = 30  # Contoh nilai suhu
kebisingan_input = 70  # Contoh nilai kebisingan
pencahayaan_input = 400  # Contoh nilai pencahayaan

# Hitung a-predicate
suhu = suhu_fuzzy(suhu_input)
kebisingan = kebisingan_fuzzy(kebisingan_input)
pencahayaan = pencahayaan_fuzzy(pencahayaan_input)

# Tampilkan a-predicate
print("A-Predicate Suhu:", suhu)
print("A-Predicate Kebisingan:", kebisingan)
print("A-Predicate Pencahayaan:", pencahayaan)

# Hitung inferensi fuzzy
hasil_inferensi = inferensi_fuzzy(suhu, kebisingan, pencahayaan)

# Tampilkan hasil inferensi
print("Hasil Inferensi:", hasil_inferensi)

# Defuzzifikasi
hasil_defuzzifikasi = defuzzifikasi(hasil_inferensi)
```

```
# Tampilkan hasil defuzzifikasi
print("Hasil Defuzzifikasi:", hasil_defuzzifikasi)

# Gambar fungsi keanggotaan
gambar_fungsi_keanggotaan()
```

Penjelasan :

Kode di atas merupakan implementasi sistem logika fuzzy yang dirancang untuk mengevaluasi produksi berdasarkan tiga variabel input: suhu, kebisingan, dan pencahayaan. Proses dimulai dengan mendefinisikan fungsi keanggotaan berbentuk segitiga melalui fungsi `fungsi_keanggotaan_segitiga`, yang mengembalikan derajat keanggotaan untuk nilai input `x` dalam interval tertentu. Jika `x` berada di luar batas-batas segitiga, maka derajat keanggotaan yang dihasilkan adalah 0. Selanjutnya, tiga fungsi terpisah—`suhu_fuzzy`, `kebisingan_fuzzy`, dan `pencahayaan_fuzzy`—diciptakan untuk mendefinisikan kategori fuzzy dari masing-masing variabel. Kategori untuk suhu terdiri dari 'dingin', 'sedang', dan 'panas', untuk kebisingan mencakup 'rendah', 'sedang', dan 'tinggi', serta untuk pencahayaan terdapat 'redup', 'sedang', dan 'terang'.

Setelah mendefinisikan keanggotaan fuzzy, fungsi `inferensi_fuzzy` diterapkan untuk menghasilkan output produksi berdasarkan 27 aturan fuzzy yang berhubungan dengan kombinasi keanggotaan dari ketiga variabel. Fungsi ini mengidentifikasi aturan yang relevan dan menghitung nilai hasil inferensi dengan mengalikan nilai keanggotaan terendah dari variabel yang memenuhi syarat dengan faktor produksi yang sesuai. Hasil dari inferensi fuzzy ini kemudian diproses lebih lanjut melalui fungsi `defuzzifikasi`, yang menghitung nilai crisp akhir berdasarkan rata-rata dari semua hasil inferensi yang dihasilkan.

Di bagian akhir kode, input untuk suhu, kebisingan, dan pencahayaan ditentukan dan diteruskan ke fungsi fuzzy untuk menghitung derajat keanggotaan masing-masing variabel. A-predicate ditampilkan untuk menunjukkan nilai keanggotaan yang dihasilkan dari input tersebut, serta hasil inferensi dan hasil defuzzifikasi yang diperoleh. Untuk memvisualisasikan fungsi keanggotaan secara grafis, kode juga menyertakan fungsi `gambar_fungsi_keanggotaan`, yang menghasilkan grafik untuk setiap variabel. Grafik ini membantu dalam memahami distribusi nilai input di dalam kategori fuzzy yang telah ditentukan, memberikan wawasan lebih lanjut tentang bagaimana sistem logika fuzzy ini berfungsi. Secara keseluruhan, kode ini menciptakan

kerangka kerja yang kuat untuk mengambil keputusan berdasarkan variabel kontinu dengan pendekatan berbasis fuzzy, memberikan hasil yang dapat digunakan dalam konteks evaluasi produksi.