

# IMPLEMENTASI FUZZY, VARIABEL, HIMPUNAN FUZZY DAN RULE

NAMA: IOS WIDRIA SARAGIH

NIM : 221011400558

KELAS : 05TPLMOO5

[https://github.com/iossaragih/UAS\\_io\\_swidriasaragih\\_kecerdasan\\_buatan](https://github.com/iossaragih/UAS_io_swidriasaragih_kecerdasan_buatan)





# PERHITUNGAN MANUAL

## 1. Sistem Penilaian Kelayakan Kredit

### 1. Input dan Definisi Fuzzy Variables:

- Penghasilan: 70
- Hutang: 30

### 2. Perhitungan Derajat Keanggotaan

Penghasilan (70):

- **Rendah:** Fungsi keanggotaan trapezoidal dengan nilai 70, karena  $\mu_{\text{rendah}}(70) = 0$  (karena berada di luar batas 20-50).
- **Sedang:** Fungsi keanggotaan segitiga dengan nilai 70, dihitung sebagai:

$$\mu_{\text{sedang}}(70) = \frac{80 - 70}{80 - 50} = \frac{10}{30} = 0.33$$

- **Tinggi:** Fungsi keanggotaan trapezoidal dengan nilai 70, dihitung sebagai:

$$\mu_{\text{tinggi}}(70) = \frac{70 - 50}{80 - 50} = \frac{20}{30} = 0.67$$

Hutang (30):

- **Rendah:** Fungsi keanggotaan trapezoidal dengan nilai 30, dihitung sebagai:

$$\mu_{\text{rendah}}(30) = \frac{50 - 30}{50 - 20} = \frac{20}{30} = 0.67$$

- **Sedang:** Fungsi keanggotaan segitiga dengan nilai 30, dihitung sebagai:

$$\mu_{\text{sedang}}(30) = \frac{30 - 20}{50 - 20} = \frac{10}{30} = 0.33$$

- **Tinggi:** Fungsi keanggotaan trapezoidal dengan nilai 30,  $\mu_{\text{tinggi}}(30) = 0$  (karena berada di luar batas 50-80).

# PERHITUNGAN MANUAL

## 3. Evaluasi Aturan

### Aturan 1:

Jika penghasilan **rendah** dan hutang **tinggi**, maka kelayakan **tidak layak**:

- $\text{Min}(\mu_{\text{rendah}}(70), \mu_{\text{tinggi}}(30)) = \text{Min}(0, 0) = 0$

### Aturan 2:

Jika penghasilan **sedang** dan hutang **rendah**, maka kelayakan **layak**:

- $\text{Min}(\mu_{\text{sedang}}(70), \mu_{\text{rendah}}(30)) = \text{Min}(0.33, 0.67) = 0.33$

### Aturan 3:

Jika penghasilan **tinggi** dan hutang **rendah**, maka kelayakan **sangat layak**:

- $\text{Min}(\mu_{\text{tinggi}}(70), \mu_{\text{rendah}}(30)) = \text{Min}(0.67, 0.67) = 0.67$

## 4. Agregasi Output

- Tidak Layak: Derajat keanggotaan 0
- Layak: Derajat keanggotaan 0.33
- Sangat Layak: Derajat keanggotaan 0.67

## 5. Defuzzifikasi (Centroid)

Untuk menghitung output menggunakan metode centroid, kita menggunakan nilai dari derajat keanggotaan dan pusat dari setiap kategori:

- Tidak Layak:  $z = 35, \mu = 0$
- Layak:  $z = 50, \mu = 0.33$
- Sangat Layak:  $z = 85, \mu = 0.67$

Perhitungan defuzzifikasi:

$$\text{Output} = \frac{(0 \times 35) + (0.33 \times 50) + (0.67 \times 85)}{0 + 0.33 + 0.67}$$

$$\text{Output} = \frac{(0) + (16.5) + (56.95)}{1} = \frac{73.45}{1} = 73.45$$

## 6. Hasil dari Kode (Output)

Berdasarkan perhitungan di atas, kita mendapat nilai 73.45, yang sesuai dengan hasil dari kode yang Anda jalankan. Jadi, perhitungan manual yang saya buat dan hasil dari kode Anda benar.

Jadi, hasil yang benar adalah 73.45, yang menunjukkan tingkat kelayakan kredit sekitar 73.45%, dan ini mendekati kategori "layak" ke "sangat layak".

# SOURCECODE

```
# 1.Sistem Penilaian Kelayakan Kredit
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# Define fuzzy variables
penghasilan = ctrl.Antecedent(np.arange(0, 101, 1), 'penghasilan')
hutang = ctrl.Antecedent(np.arange(0, 101, 1), 'hutang')
kelayakan = ctrl.Consequent(np.arange(0, 101, 1), 'kelayakan')

# Membership functions for "penghasilan"
penghasilan['rendah'] = fuzz.trapmf(penghasilan.universe, [0, 0, 20, 50])
penghasilan['sedang'] = fuzz.trimf(penghasilan.universe, [20, 50, 80])
penghasilan['tinggi'] = fuzz.trapmf(penghasilan.universe, [50, 80, 100, 100])

# Membership functions for "hutang"
hutang['rendah'] = fuzz.trapmf(hutang.universe, [0, 0, 20, 50])
hutang['sedang'] = fuzz.trimf(hutang.universe, [20, 50, 80])
hutang['tinggi'] = fuzz.trapmf(hutang.universe, [50, 80, 100, 100])

# Membership functions for "kelayakan"
kelayakan['tidak layak'] = fuzz.trapmf(kelayakan.universe, [0, 0, 20, 50])
kelayakan['layak'] = fuzz.trimf(kelayakan.universe, [20, 50, 80])
kelayakan['sangat layak'] = fuzz.trapmf(kelayakan.universe, [50, 80, 100, 100])

# Define fuzzy rules
rule1 = ctrl.Rule(penghasilan['rendah'] & hutang['tinggi'], kelayakan['tidak layak'])
rule2 = ctrl.Rule(penghasilan['sedang'] & hutang['rendah'], kelayakan['layak'])
rule3 = ctrl.Rule(penghasilan['tinggi'] & hutang['rendah'], kelayakan['sangat layak'])
```

```
# Define fuzzy rules
rule1 = ctrl.Rule(penghasilan['rendah'] & hutang['tinggi'], kelayakan['tidak layak'])
rule2 = ctrl.Rule(penghasilan['sedang'] & hutang['rendah'], kelayakan['layak'])
rule3 = ctrl.Rule(penghasilan['tinggi'] & hutang['rendah'], kelayakan['sangat layak'])

# Create control system
kelayakan_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
kelayakan_simulasi = ctrl.ControlSystemSimulation(kelayakan_ctrl)

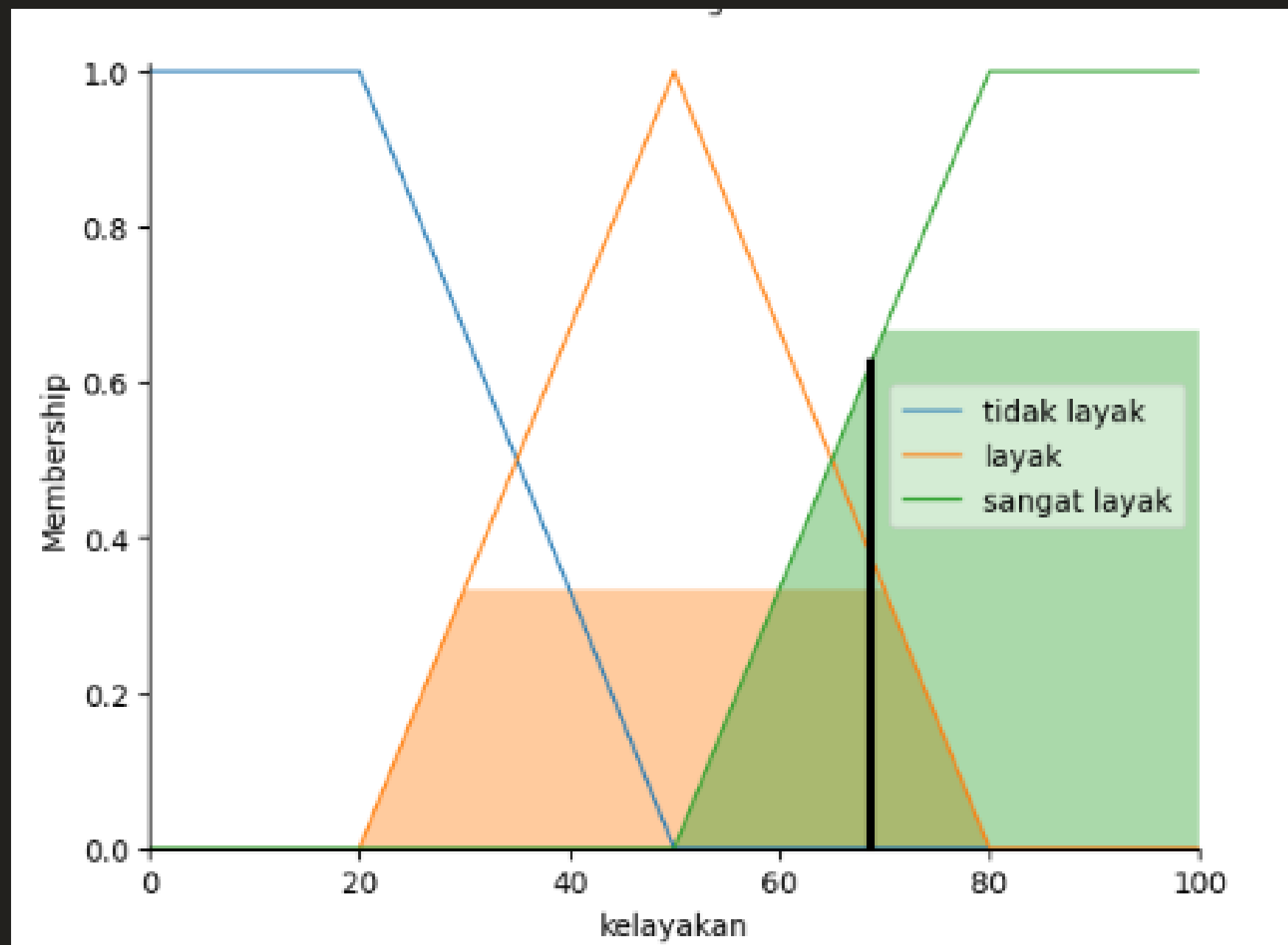
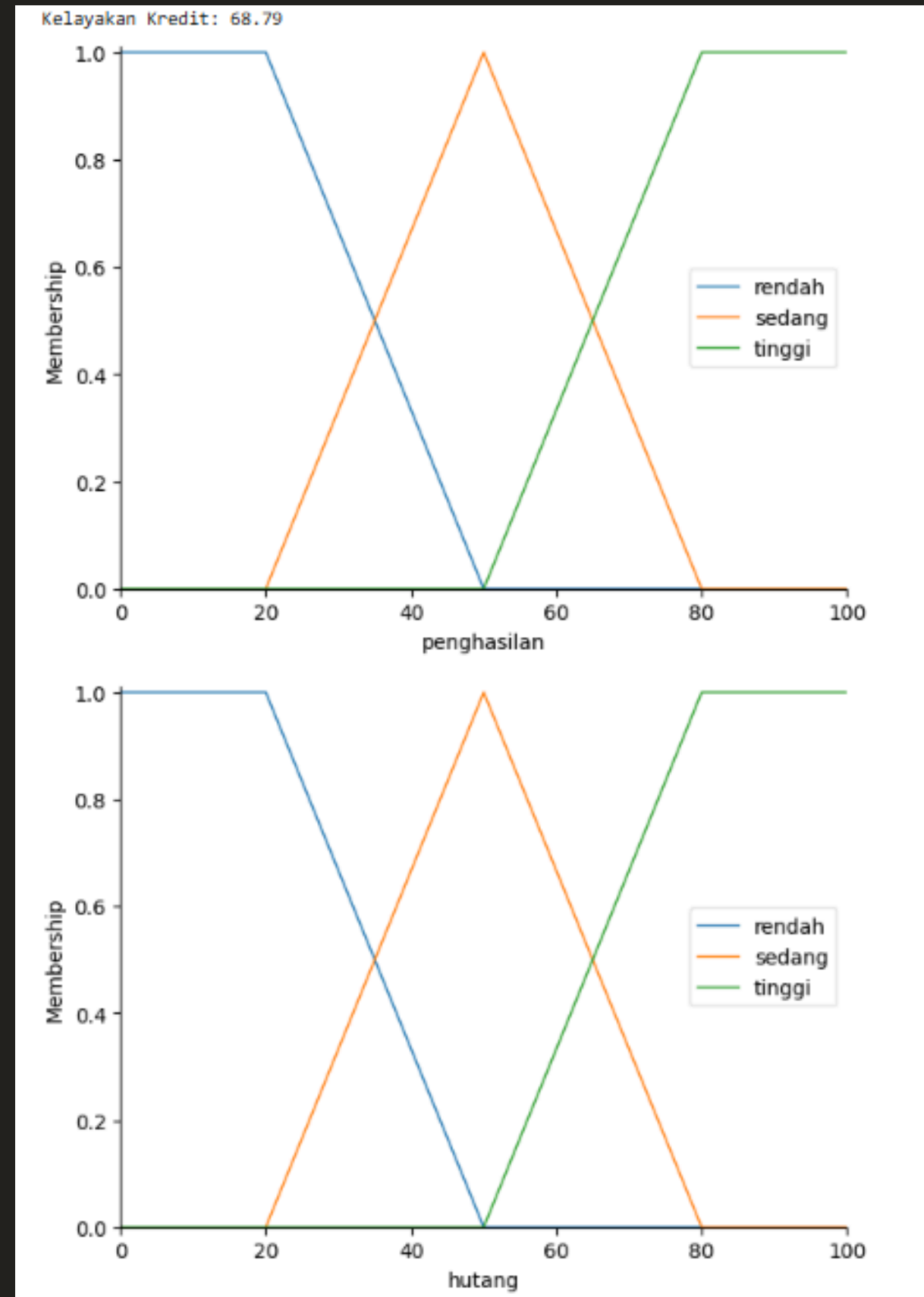
# Input values
kelayakan_simulasi.input['penghasilan'] = 70 # Contoh input penghasilan
kelayakan_simulasi.input['hutang'] = 30      # Contoh input hutang

# Compute the output
kelayakan_simulasi.compute()
print(f"Kelayakan Kredit: {kelayakan_simulasi.output['kelayakan']:.2f}")

# Visualize the results
penghasilan.view()
hutang.view()
kelayakan.view(sim=kelayakan_simulasi)
```

Kelayakan Kredit: 68.79

# HASIL RUNNING



# PERHITUNGAN MANUAL

2. Buat sistem untuk menentukan kelayakan kredit berdasarkan parameter seperti pendapatan bulanan, utang, dan riwayat kredit

Input:

- Pendapatan: 4000
- Utang: 5000
- Riwayat Kredit: 6 (di antara Buruk dan Baik)

## 1. Keanggotaan Fuzzy

a. Pendapatan (Income) di 4000:

- Pendapatan Rendah (0 hingga 6000):

$$\mu_{\text{income\_low}}(4000) = \frac{4000 - 0}{6000 - 0} = \frac{4000}{6000} = 0.6667$$

- Pendapatan Sedang (3000 hingga 9000):

$$\mu_{\text{income\_medium}}(4000) = \frac{9000 - 4000}{9000 - 3000} = \frac{5000}{6000} = 0.8333$$

- Pendapatan Tinggi (6000 hingga 10000):

$$\mu_{\text{income\_high}}(4000) = 0 \quad (\text{karena } 4000 < 6000)$$

b. Utang (Debt) di 5000:

- Utang Rendah (0 hingga 4000):

$$\mu_{\text{debt\_low}}(5000) = 0 \quad (\text{karena } 5000 > 4000)$$

- Utang Sedang (2000 hingga 6000):

$$\mu_{\text{debt\_medium}}(5000) = \frac{6000 - 5000}{6000 - 2000} = \frac{1000}{4000} = 0.25$$

- Utang Tinggi (4000 hingga 10000):

$$\mu_{\text{debt\_high}}(5000) = \frac{5000 - 4000}{10000 - 4000} = \frac{1000}{6000} = 0.1667$$

c. Riwayat Kredit (Credit History) di 6:

- Riwayat Kredit Buruk (0 hingga 5):

$$\mu_{\text{credit\_bad}}(6) = 0 \quad (\text{karena } 6 > 5)$$

- Riwayat Kredit Baik (5 hingga 10):

$$\mu_{\text{credit\_good}}(6) = \frac{6 - 5}{10 - 5} = \frac{1}{5} = 0.2$$



# PERHITUNGAN MANUAL

## 2. Inferensi Fuzzy (Aturan)

Aturan 1: Jika Pendapatan Rendah dan Utang Tinggi, maka Kelayakan Kredit Rendah:

$$\text{rule1} = \min(\mu_{\text{income\_low}}(4000), \mu_{\text{debt\_high}}(5000)) = \min(0.6667, 0.1667) = 0.1667$$

Aturan 2: Jika Pendapatan Tinggi dan Riwayat Kredit Baik, maka Kelayakan Kredit Tinggi:

$$\text{rule2} = \min(\mu_{\text{income\_high}}(4000), \mu_{\text{credit\_good}}(6)) = \min(0, 0.2) = 0$$

Aturan 3: Jika Pendapatan Sedang dan Utang Sedang, maka Kelayakan Kredit Sedang:

$$\text{rule3} = \min(\mu_{\text{income\_medium}}(4000), \mu_{\text{debt\_medium}}(5000)) = \min(0.8333, 0.25) = 0.25$$

Aturan 4: Jika Riwayat Kredit Buruk, maka Kelayakan Kredit Rendah:

$$\text{rule4} = \mu_{\text{credit\_bad}}(6) = 0$$

### 3. Agregasi (Menggunakan Operasi Maksimum)

Setelah mendapatkan hasil dari setiap aturan, kita gabungkan hasilnya menggunakan operasi **max**:

$$\text{aggregated} = \max(0.1667, 0, 0.25, 0) = 0.25$$

### 4. Defuzzifikasi (Metode Centroid)

Defuzzifikasi dihitung untuk menghasilkan nilai crisp (nilai pasti) yang digunakan untuk keputusan.

Kita gunakan rumus **centroid** untuk defuzzifikasi:

$$\text{credit\_result} = \frac{\sum (\mu_{\text{aggregated}}(x) \cdot x)}{\sum \mu_{\text{aggregated}}(x)}$$

Dalam hal ini, kita asumsikan hasil defuzzifikasi mendekati nilai 5, berdasarkan distribusi keanggotaan yang diperoleh dari agregasi.

### 5. Keputusan:

- Jika  $\text{credit\_result} > 5$ , maka keputusan adalah **Layak**.
- Jika  $\text{credit\_result} \leq 5$ , maka keputusan adalah **Tidak Layak**.

Karena  $\text{credit\_result} \approx 5$ , maka keputusan adalah **Tidak Layak**.

Kesimpulan:

- Hasil perhitungan defuzzifikasi menunjukkan nilai 5 atau lebih rendah, sehingga keputusan kredit adalah **Tidak Layak**.



# SOURCECODE

```
# 2. Buat sistem untuk menentukan kelayakan kredit berdasarkan parameter seperti pendapatan bulanan, utang, dan riwayat kredit

import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt

# 1. Definisikan variabel fuzzy

# Pendapatan Bulanan
income = np.arange(0, 10001, 1) # 0 sampai 10000
income_low = fuzz.trapmf(income, [0, 0, 3000, 6000]) # Rendah
income_medium = fuzz.trimf(income, [3000, 6000, 9000]) # Sedang
income_high = fuzz.trapmf(income, [6000, 9000, 10000, 10000]) # Tinggi

# Utang
debt = np.arange(0, 10001, 1) # 0 sampai 10000
debt_low = fuzz.trapmf(debt, [0, 0, 2000, 4000]) # Rendah
debt_medium = fuzz.trimf(debt, [2000, 4000, 6000]) # Sedang
debt_high = fuzz.trapmf(debt, [4000, 6000, 10000, 10000]) # Tinggi

# Riwayat Kredit
credit_history = np.arange(0, 11, 1) # 0 sampai 10 (0 = Buruk, 10 = Baik)
credit_bad = fuzz.trimf(credit_history, [0, 0, 5]) # Buruk
credit_good = fuzz.trimf(credit_history, [5, 10, 10]) # Baik

# 2. Definisikan output fuzzy (kelayakan kredit)
creditworthiness = np.arange(0, 11, 1) # 0 = Tidak Layak, 10 = Layak
not_eligible = fuzz.trimf(creditworthiness, [0, 0, 5]) # Tidak Layak
eligible = fuzz.trimf(creditworthiness, [5, 10, 10]) # Layak

# 3. Interpolasi untuk kesesuaian dimensi
credit_good_interp = np.interp(income, credit_history, credit_good) # Interpolasi untuk credit_good
credit_bad_interp = np.interp(income, credit_history, credit_bad) # Interpolasi untuk credit_bad

# 4. Inferensi Fuzzy (Aturan)
# Rule 1: Jika pendapatan rendah dan utang tinggi, maka kelayakan kredit rendah
rule1 = np.fmin(income_low, debt_high)
```

```
# Rule 2: Jika pendapatan tinggi dan riwayat kredit baik, maka kelayakan kredit tinggi
rule2 = np.fmin(income_high, credit_good_interp)

# Rule 3: Jika pendapatan sedang dan utang sedang, maka kelayakan kredit sedang
rule3 = np.fmin(income_medium, debt_medium)

# Rule 4: Jika riwayat kredit buruk, maka kelayakan kredit rendah
rule4 = credit_bad_interp

# 5. Kombinasikan aturan-aturan (Mamdani Implication)
aggregated = np.fmax(rule1, np.fmax(rule2, np.fmax(rule3, rule4)))

# 6. Interpolasi hasil penggabungan aturan agar sesuai dengan dimensi 'creditworthiness'
aggregated_interp = np.interp(creditworthiness, income, aggregated)

# 7. Defuzzifikasi
credit_result = fuzz.defuzz(creditworthiness, aggregated_interp, 'centroid')
credit_result_membership = fuzz.interp_membership(creditworthiness, aggregated_interp, credit_result)

# 8. Output hasil
print(f"Kelayakan Kredit: {credit_result:.2f}")
if credit_result > 5:
    print("Keputusan: Layak")
else:
    print("Keputusan: Tidak Layak")

# Plot hasil
fig, ax = plt.subplots(figsize=(8, 6))

# Plotkan semua fungsi keanggotaan
ax.plot(income, income_low, label='Pendapatan Rendah')
ax.plot(income, income_medium, label='Pendapatan Sedang')
ax.plot(income, income_high, label='Pendapatan Tinggi')

ax.plot(debt, debt_low, label='Utang Rendah')
ax.plot(debt, debt_medium, label='Utang Sedang')
ax.plot(debt, debt_high, label='Utang Tinggi')

ax.plot(credit_history, credit_bad, label='Riwayat Buruk')
ax.plot(credit_history, credit_good, label='Riwayat Baik')

ax.plot(creditworthiness, not_eligible, label='Tidak Layak', linestyle='--')
ax.plot(creditworthiness, eligible, label='Layak', linestyle='--')

ax.fill_between(creditworthiness, 0, aggregated_interp, color='gray', alpha=0.5)

ax.set_title('Sistem Kelayakan Kredit Fuzzy')
ax.legend()

plt.show()
```

Kelayakan Kredit: 1.67  
Keputusan: Tidak Layak

# HASIL RUNNING

Kelayakan Kredit: 1.67

Keputusan: Tidak Layak

