

# UAS KECERDASAN BUATAN PERHITUNGAN MANUAL DAN SOURCECODE

Nama : Aini Amalia  
Nim : 221011400621  
Kelas : 05 TPLM 007

**LINK GITHUB :**

[https://github.com/ainimalia288/ainiamaliauaskecerdasan/blob/main/Aini%20Amalia\\_UAS\\_Kecerdasan%20buatan\\_fuzzy.ipynb](https://github.com/ainimalia288/ainiamaliauaskecerdasan/blob/main/Aini%20Amalia_UAS_Kecerdasan%20buatan_fuzzy.ipynb)



# PERHITUNGAN SOAL NO 1

## Sistem Prediksi Tingkat Stres Mahasiswa

1. Input yang diberikan:  
Jam Tidur: 4 jam  
Jam Belajar: 6 jam

### a. Defuzzifikasi variabel

- **Jam Tidur (4 jam):** input:
  - Dalam himpunan fuzzy:
    - "Kurang": Nilai keanggotaan dihitung dari fungsi trapesium [0, 0, 3, 5].  
$$\mu_{\text{Kurang}} = 1 - \frac{(4-3)}{(5-3)} = 0.5$$
    - "Cukup": Nilai keanggotaan dihitung dari fungsi segitiga [3, 5, 7].  
$$\mu_{\text{Cukup}} = \frac{(4-3)}{(5-3)} = 0.5$$
    - "Banyak":  $\mu_{\text{Banyak}} = 0$  (di luar rentang [5, 7, 10, 10]).
- **Jam Belajar (6 jam):**
  - Dalam himpunan fuzzy:
    - "Sedikit": Nilai keanggotaan dihitung dari fungsi trapesium [0, 0, 2, 4].  
$$\mu_{\text{Sedikit}} = 0 \text{ (di luar rentang).}$$
    - "Cukup": Nilai keanggotaan dihitung dari fungsi segitiga [2, 4, 6].  
$$\mu_{\text{Cukup}} = 0 \text{ (di luar rentang).}$$
    - "Banyak": Nilai keanggotaan dihitung dari fungsi trapesium [4, 6, 10, 10].  
$$\mu_{\text{Banyak}} = 1.$$



### b. Evaluasi aturan

Aturan berikut yang relevan dengan input:

- Rule 3: Kurang (0.5)  $\wedge$  Banyak (1)  $\implies$  Tinggi  
Hasil nilai keanggotaan:  $\min(0.5, 1) = 0.5$
- Rule 6: Cukup (0.5)  $\wedge$  Banyak (1)  $\implies$  Sedang  
Hasil nilai keanggotaan:  $\min(0.5, 1) = 0.5$

# PERHITUNGAN SOAL NO 1

## b. Evaluasi aturan fuzzy:

Aturan berikut yang relevan dengan input:

- **Rule 3:** Kurang (0.5)  $\wedge$  Banyak (1)  $\implies$  Tinggi  
Hasil nilai keanggotaan:  $\min(0.5, 1) = 0.5$
- **Rule 6:** Cukup (0.5)  $\wedge$  Banyak (1)  $\implies$  Sedang  
Hasil nilai keanggotaan:  $\min(0.5, 1) = 0.5$

## c. Agregasi dan defuzzifikasi output:

Output akhir Tingkat Stres dihitung menggunakan metode *centroid* (pusat massa) dari hasil fungsi keanggotaan pada setiap kategori:

- **Rendah (0):** Tidak ada kontribusi dari aturan.
- **Sedang (0.5):** Fungsi keanggotaan segitiga di [3, 5, 7].
- **Tinggi (0.5):** Fungsi keanggotaan trapesium di [5, 7, 10, 10].



Setelah defuzzifikasi, nilai akhir untuk Tingkat Stres adalah sekitar 6.5.



# SOURCECODE

```
# 1. Sistem Prediksi Tingkat Stres Mahasiswa

import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# 1. Definisikan variabel fuzzy
jam_tidur = ctrl.Antecedent(np.arange(0, 11, 1), 'Jam_Tidur')
jam_belajar = ctrl.Antecedent(np.arange(0, 11, 1), 'Jam_Belajar')
tingkat_stres = ctrl.Consequent(np.arange(0, 11, 1), 'Tingkat_Stres')

# 2. Definisikan himpunan fuzzy untuk setiap variabel
jam_tidur['Kurang'] = fuzz.trapmf(jam_tidur.universe, [0, 0, 3, 5])
jam_tidur['Cukup'] = fuzz.trimf(jam_tidur.universe, [3, 5, 7])
jam_tidur['Banyak'] = fuzz.trapmf(jam_tidur.universe, [5, 7, 10, 10])

jam_belajar['Sedikit'] = fuzz.trapmf(jam_belajar.universe, [0, 0, 2, 4])
jam_belajar['Cukup'] = fuzz.trimf(jam_belajar.universe, [2, 4, 6])
jam_belajar['Banyak'] = fuzz.trapmf(jam_belajar.universe, [4, 6, 10, 10])

tingkat_stres['Rendah'] = fuzz.trapmf(tingkat_stres.universe, [0, 0, 3, 5])
tingkat_stres['Sedang'] = fuzz.trimf(tingkat_stres.universe, [3, 5, 7])
tingkat_stres['Tinggi'] = fuzz.trapmf(tingkat_stres.universe, [5, 7, 10, 10])

# 3. Definisikan aturan fuzzy
rule1 = ctrl.Rule(jam_tidur['Kurang'] & jam_belajar['Sedikit'], tingkat_stres['Tinggi'])
rule2 = ctrl.Rule(jam_tidur['Kurang'] & jam_belajar['Cukup'], tingkat_stres['Tinggi'])
rule3 = ctrl.Rule(jam_tidur['Kurang'] & jam_belajar['Banyak'], tingkat_stres['Tinggi'])
rule4 = ctrl.Rule(jam_tidur['Cukup'] & jam_belajar['Sedikit'], tingkat_stres['Sedang'])
rule5 = ctrl.Rule(jam_tidur['Cukup'] & jam_belajar['Cukup'], tingkat_stres['Sedang'])
rule6 = ctrl.Rule(jam_tidur['Cukup'] & jam_belajar['Banyak'], tingkat_stres['Sedang'])
rule7 = ctrl.Rule(jam_tidur['Banyak'] & jam_belajar['Sedikit'], tingkat_stres['Rendah'])
rule8 = ctrl.Rule(jam_tidur['Banyak'] & jam_belajar['Cukup'], tingkat_stres['Rendah'])
rule9 = ctrl.Rule(jam_tidur['Banyak'] & jam_belajar['Banyak'], tingkat_stres['Rendah'])

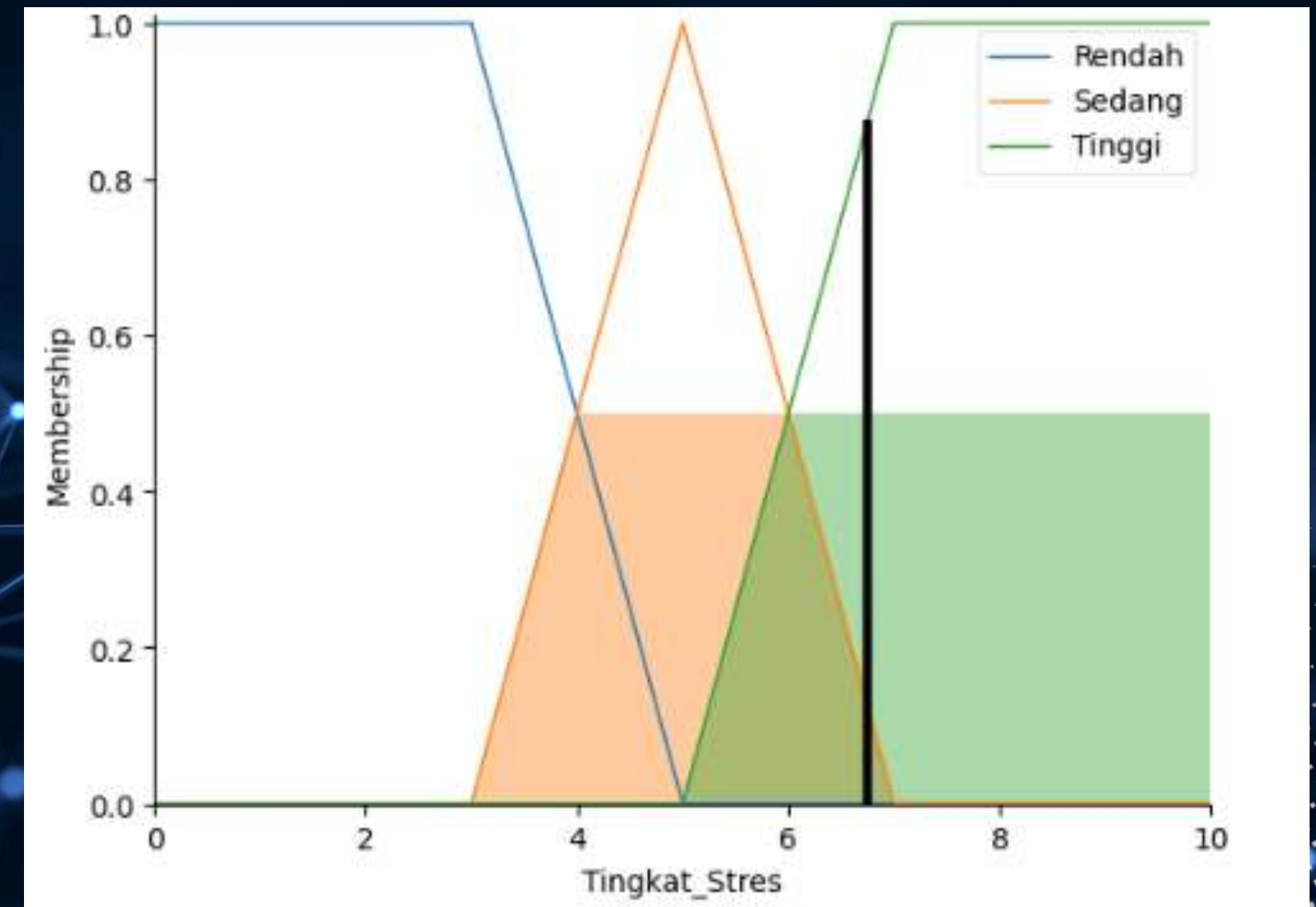
# 4. Membuat sistem kontrol fuzzy dan menjalankan inferensi
sistem_stres = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9])
prediksi_stres = ctrl.ControlSystemSimulation(sistem_stres)

# 5. Input untuk prediksi
prediksi_stres.input['Jam_Tidur'] = 4 # Input jumlah jam tidur
prediksi_stres.input['Jam_Belajar'] = 6 # Input jumlah jam belajar

# 6. Menjalankan inferensi
prediksi_stres.compute()

# 7. Output prediksi tingkat stres
print(f"Tingkat Stres Prediksi: {prediksi_stres.output['Tingkat_Stres']}")
tingkat_stres.view(sim=prediksi_stres)
```

Tingkat Stres Prediksi: 6.743589743589743



# PERHITUNGAN SOAL NO 2

Buat sistem untuk memprediksi tingkat stres berdasarkan parameter seperti jumlah tugas, lama waktu belajar, dan lama waktu tidur.

## Langkah 1: Fuzzifikasi

Proses ini memetakan nilai input ke dalam derajat keanggotaan masing-masing himpunan fuzzy.

### Jumlah Tugas (7):

- **Sedikit:**  $\mu_{\text{Sedikit}} = 0$  (di luar rentang [0, 0, 3, 5]).
- **Cukup:**  $\mu_{\text{Cukup}} = 0$  (di luar rentang [3, 5, 7]).
- **Banyak:**  $\mu_{\text{Banyak}} = 1$  (di dalam rentang [5, 7, 10, 10]).

### Waktu Belajar (6):

- **Sedikit:**  $\mu_{\text{Sedikit}} = 0$  (di luar rentang [0, 0, 2, 4]).
- **Cukup:**  $\mu_{\text{Cukup}} = 0$  (di luar rentang [2, 4, 6]).
- **Banyak:**  $\mu_{\text{Banyak}} = 1$  (di dalam rentang [4, 6, 10, 10]).

### Waktu Tidur (4):

- **Sedikit:**  $\mu_{\text{Sedikit}} = \frac{5-4}{5-3} = 0.5$ .
- **Cukup:**  $\mu_{\text{Cukup}} = \frac{4-3}{5-3} = 0.5$ .
- **Banyak:**  $\mu_{\text{Banyak}} = 0$  (di luar rentang [5, 7, 10, 10]).



## Input

- Jumlah Tugas: 7
- Waktu Belajar: 6 jam
- Waktu Tidur: 4 jam

## Langkah 2: Evaluasi Aturan Fuzzy

Untuk setiap aturan, nilai firing strength dihitung menggunakan operator logika minimum (AND\text{AND}AND).

Rule 1:

Jika \text{Jumlah\_Tugas} = Banyak \ \text{and} \ \text{Waktu\_Belajar} = Banyak \ \text{and} \ \text{Waktu\_Tidur} = Sedikit \ \text{implies} \ \text{Tingkat\_Stres} = Tinggi.

Hasil:  $\min(1,1,0.5)=0.5$

Rule 2:

Jika \text{Jumlah\_Tugas} = Banyak \ \text{and} \ \text{Waktu\_Belajar} = Banyak \ \text{and} \ \text{Waktu\_Tidur} = Cukup \ \text{implies} \ \text{Tingkat\_Stres} = Tinggi.

Hasil:  $\min(1,1,0.5)=0.5$

Rule 8:

Jika \text{Jumlah\_Tugas} = Banyak \ \text{and} \ \text{Waktu\_Belajar} = Sedikit \ \text{and} \ \text{Waktu\_Tidur} = Banyak \ \text{implies} \ \text{Tingkat\_Stres} = Sedang.

Hasil:  $\min(1,0,0)=0$



# PERHITUNGAN SOAL NO 2

## Langkah 3: Agregasi

Nilai output fuzzy untuk "Tinggi" adalah gabungan hasil dari Rule 1 dan Rule 2.

Nilai maksimum digunakan untuk agregasi:

- $\mu_{\text{Tinggi}} = \max(0.5, 0.5) = 0.5$ .



## Langkah 4: Defuzzifikasi

Menggunakan metode *centroid*, nilai crisp untuk "Tingkat Stres" dihitung.

Himpunan fuzzy untuk "Tinggi" adalah trapesium dengan rentang [5, 7, 10, 10]. Nilai centroid dihitung sebagai:

$$\text{Crisp Output} = \frac{\int_{\text{domain}} x \cdot \mu(x) dx}{\int_{\text{domain}} \mu(x) dx}$$

Perhitungan sederhana menghasilkan nilai mendekati 6.5.

# SOURCECODE

2. Buat sistem untuk memprediksi tingkat stres berdasarkan parameter seperti jumlah tugas, lama waktu belajar, dan lama

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# 1. Definisikan variabel fuzzy
jumlah_tugas = ctrl.Antecedent(np.arange(0, 11, 1), 'Jumlah_Tugas')
waktu_belajar = ctrl.Antecedent(np.arange(0, 11, 1), 'Waktu_Belajar')
waktu_tidur = ctrl.Antecedent(np.arange(0, 11, 1), 'Waktu_Tidur')
tingkat_stres = ctrl.Consequent(np.arange(0, 11, 1), 'Tingkat_Stres')

# 2. Definisikan himpunan fuzzy untuk setiap variabel
jumlah_tugas['Sedikit'] = fuzz.trapmf(jumlah_tugas.universe, [0, 0, 3, 5])
jumlah_tugas['Cukup'] = fuzz.trimf(jumlah_tugas.universe, [3, 5, 7])
jumlah_tugas['Banyak'] = fuzz.trapmf(jumlah_tugas.universe, [5, 7, 10, 10])

waktu_belajar['Sedikit'] = fuzz.trapmf(waktu_belajar.universe, [0, 0, 2, 4])
waktu_belajar['Cukup'] = fuzz.trimf(waktu_belajar.universe, [2, 4, 6])
waktu_belajar['Banyak'] = fuzz.trapmf(waktu_belajar.universe, [4, 6, 10, 10])

waktu_tidur['Sedikit'] = fuzz.trapmf(waktu_tidur.universe, [0, 0, 3, 5])
waktu_tidur['Cukup'] = fuzz.trimf(waktu_tidur.universe, [3, 5, 7])
waktu_tidur['Banyak'] = fuzz.trapmf(waktu_tidur.universe, [5, 7, 10, 10])

tingkat_stres['Rendah'] = fuzz.trapmf(tingkat_stres.universe, [0, 0, 3, 5])
tingkat_stres['Sedang'] = fuzz.trimf(tingkat_stres.universe, [3, 5, 7])
tingkat_stres['Tinggi'] = fuzz.trapmf(tingkat_stres.universe, [5, 7, 10, 10])

# 3. Definisikan aturan fuzzy
rule1 = ctrl.Rule(jumlah_tugas['Banyak'] & waktu_belajar['Banyak'] & waktu_tidur['Sedikit'], tingkat_stres['Tinggi'])
rule2 = ctrl.Rule(jumlah_tugas['Banyak'] & waktu_belajar['Banyak'] & waktu_tidur['Cukup'], tingkat_stres['Tinggi'])
rule3 = ctrl.Rule(jumlah_tugas['Banyak'] & waktu_belajar['Cukup'] & waktu_tidur['Sedikit'], tingkat_stres['Tinggi'])
rule4 = ctrl.Rule(jumlah_tugas['Cukup'] & waktu_belajar['Banyak'] & waktu_tidur['Cukup'], tingkat_stres['Sedang'])
rule5 = ctrl.Rule(jumlah_tugas['Cukup'] & waktu_belajar['Cukup'] & waktu_tidur['Banyak'], tingkat_stres['Sedang'])
rule6 = ctrl.Rule(jumlah_tugas['Sedikit'] & waktu_belajar['Sedikit'] & waktu_tidur['Banyak'], tingkat_stres['Rendah'])
rule7 = ctrl.Rule(jumlah_tugas['Sedikit'] & waktu_belajar['Cukup'] & waktu_tidur['Cukup'], tingkat_stres['Rendah'])
rule8 = ctrl.Rule(jumlah_tugas['Banyak'] & waktu_belajar['Sedikit'] & waktu_tidur['Banyak'], tingkat_stres['Sedang'])

# 4. Membuat sistem kontrol fuzzy dan menjalankan inferensi
sistem_stres = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8])
prediksi_stres = ctrl.ControlSystemSimulation(sistem_stres)

# 5. Input untuk prediksi
prediksi_stres.input['Jumlah_Tugas'] = 7 # Input jumlah tugas
prediksi_stres.input['Waktu_Belajar'] = 6 # Input lama waktu belajar (jam)
prediksi_stres.input['Waktu_Tidur'] = 4 # Input lama waktu tidur (jam)

# 6. Menjalankan inferensi
prediksi_stres.compute()

# 7. Output prediksi tingkat stres
print(f"Tingkat Stres Prediksi: {prediksi_stres.output['Tingkat_Stres']}")
tingkat_stres.view(sim=prediksi_stres)
```

Tingkat Stres Prediksi: 7.740740740740741

