

BAB IX

FUZZY LOGIC

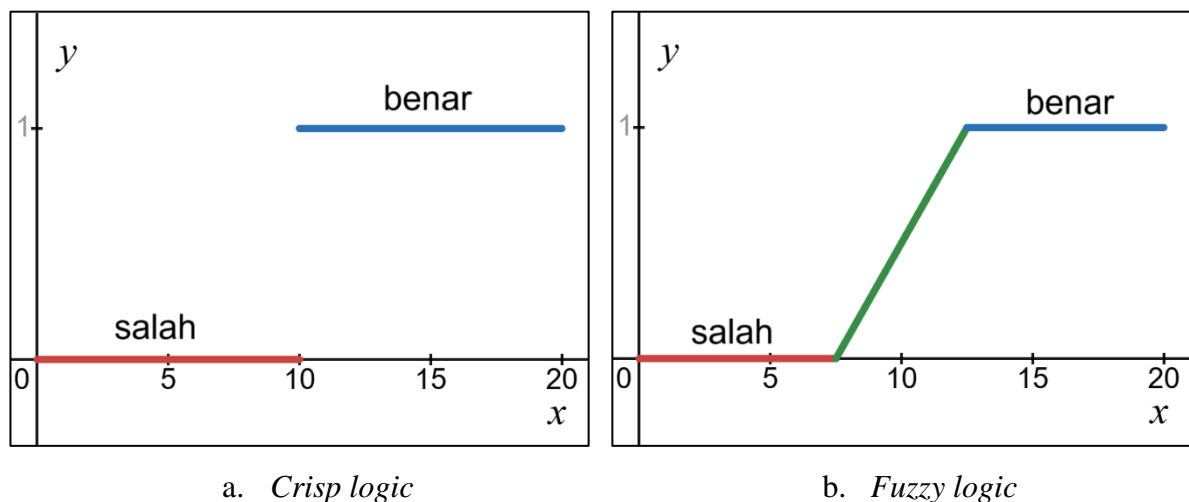
I. TUJUAN

- Memahami metode logika *fuzzy* dalam menentukan keputusan
- Mampu mengaplikasikan metode logika *fuzzy* dalam Sistem Pendukung Keputusan menggunakan Python

II. PENDAHULUAN

Metode *Fuzzy Logic*

Logika *fuzzy* (*fuzzy logic*) adalah kerangka berpikir yang meniru cara manusia mengambil keputusan di bawah ketidakpastian dan ketidakjelasan. Fokusnya adalah pada penalaran mendekati (*approximate reasoning*) daripada logika klasik yang kaku atau biner (Zadeh, 2013). Berbeda dengan logika tegas (*crisp logic*) yang memetakan nilai kebenaran secara biner—0 atau 1, salah atau benar—logika *fuzzy* dapat memiliki nilai yang sifatnya tidak tegas atau berada di tengah-tengah, seperti “sedikit”, “agak”, “biasa saja”, dan “cukup”. Perbedaan antara logika *fuzzy* dan *crisp* dapat dilihat pada gambar berikut:



Gambar 8.1 Perbedaan *crisp logic* (a) dan *fuzzy logic* (b)

Pada gambar 8.1 bagian a, dapat dilihat bahwa y akan bernilai salah jika x kurang dari 10 dan akan bernilai benar jika x lebih dari atau sama dengan 10. Sementara itu, pada bagian b, terdapat bagian berwarna hijau yang menunjukkan bahwa nilai y berada di antara salah dan benar ketika x lebih dari 7,5 dan kurang dari 12,5. Menurut logika *fuzzy*, nilai kebenaran pada rentang tersebut dapat dikategorikan sebagai benar atau salah berdasarkan derajat keanggotaannya.

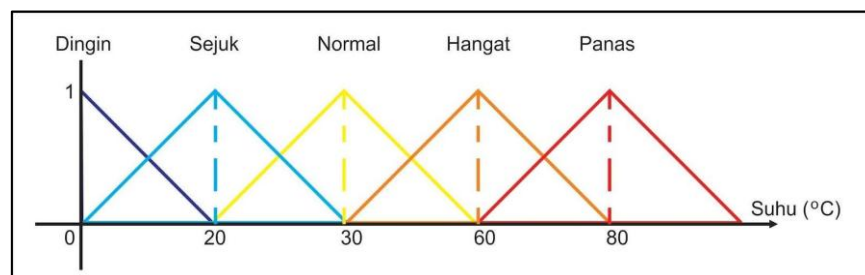
Logika *fuzzy* memiliki 3 sistem inferensi (*inference system*) yang terkenal, yaitu Mamdani, Tsukamoto, dan Sugeno. Sistem inferensi adalah kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk IF-THEN, dan penalaran *fuzzy* (Athiyah et al., 2021). Pada library scikit-fuzzy yang akan digunakan pada praktikum ini, sistem inferensi yang akan digunakan adalah Mamdani.

Himpunan Logika Fuzzy (*Fuzzy Set*)

Dalam teori yang ada pada logika fuzzy, dikenal istilah fuzzy set yang merupakan pengelompokan sesuatu berdasarkan variabel bahasa (linguistik) yang dinyatakan dalam fungsi keanggotaan yang bernilai 0 sampai dengan nilai 1. Contoh dari variabel linguistik antara lain:

1. Suhu

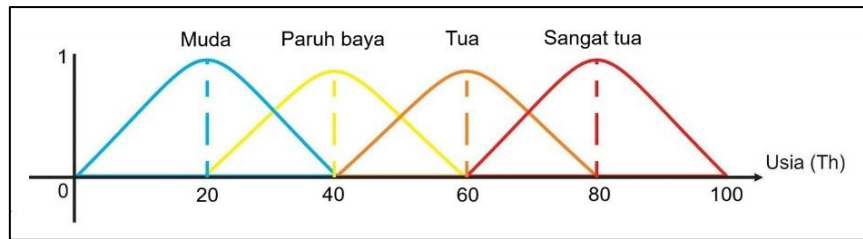
Suhu atau temperatur dapat dinyatakan dengan dingin, sejuk, normal, hangat, panas, yang ditunjukkan pada grafik berikut:



Gambar 8.2 Grafik Suhu

2. Usia

Usia dapat dikategorikan menjadi 4, yaitu muda, paruh baya, tua, dan sangat tua. Grafik *fuzzy* untuk usia dapat dilihat pada gambar berikut:

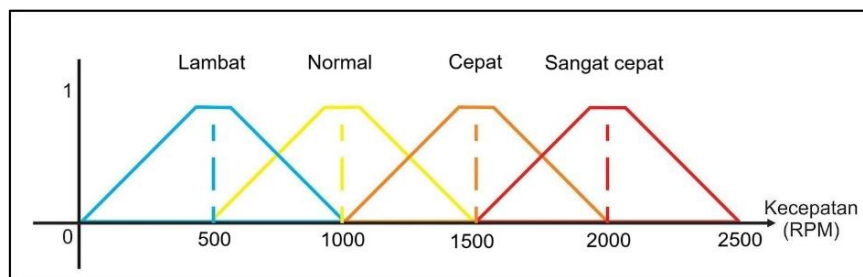


Gambar 8.3 Grafik Usia

3. Kecepatan

Kecepatan dapat dikategorikan menjadi lambat, normal, cepat, dan sangat cepat.

Grafik *fuzzy* untuk kecepatan dapat dilihat pada gambar berikut:



Gambar 8.4 Grafik Kecepatan

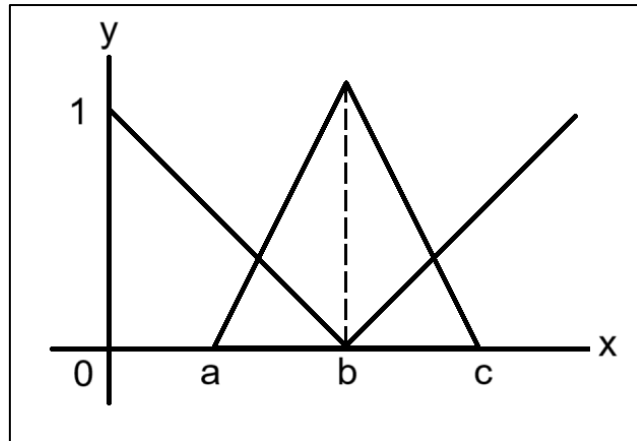
Fungsi Keanggotaan (*Membership Function*)

Fungsi keanggotaan merupakan himpunan yang menyatakan derajat keanggotaan suatu nilai terhadap nilai tegasnya yang berkisar 0 sampai dengan 1. Dalam logika fuzzy, nilai keanggotaan dapat ditentukan dengan beberapa fungsi, seperti segitiga (*trimf*), trapesium (*trapmf*), dan Gaussian (*gausmf*).

1. Segitiga (*trimf*)

Fungsi segitiga memiliki 3 titik yang dinyatakan sebagai a, b, dan c. Fungsi tersebut dapat dinyatakan dengan persamaan:

$$f(x, a, b, c) = \begin{cases} 0; & x < a \cup x > c \\ \frac{x - a}{b - a}; & a \leq x \leq b \\ \frac{c - x}{c - b}; & b < x \leq c \end{cases}$$

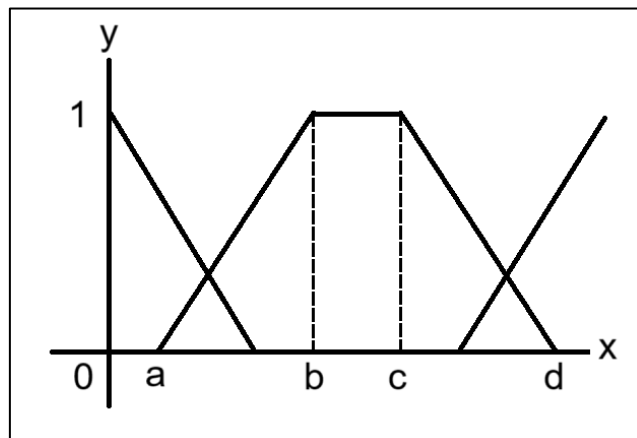


Gambar 8.5 Fungsi Keanggotaan Segitiga

2. Trapezium (*trapmf*)

Fungsi trapesium memiliki 4 titik yang dinyatakan sebagai a, b, c, dan d. Fungsi tersebut dapat dinyatakan dengan persamaan:

$$f(x, a, b, c, d) = \begin{cases} \frac{x-a}{b-a}; & a \leq x < b \\ 1; & b \leq x \leq c \\ \frac{d-x}{d-c}; & c < x \leq d \\ 0; & x < a \cup x > d \end{cases}$$

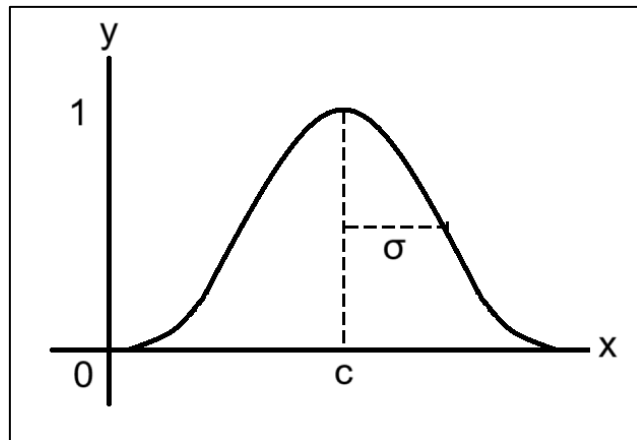


Gambar 8.6 Fungsi Keanggotaan Trapezium

3. Gaussian (*gausmf*)

Fungsi keanggotaan gaussian memiliki 2 parameter penting, yaitu titik tengah (c) dan standar deviasi (σ). Fungsi tersebut dapat dinyatakan dengan persamaan:

$$\mu(x, c, \sigma) = \frac{1}{1 + \left(\frac{x + c}{\sigma}\right)^2}$$



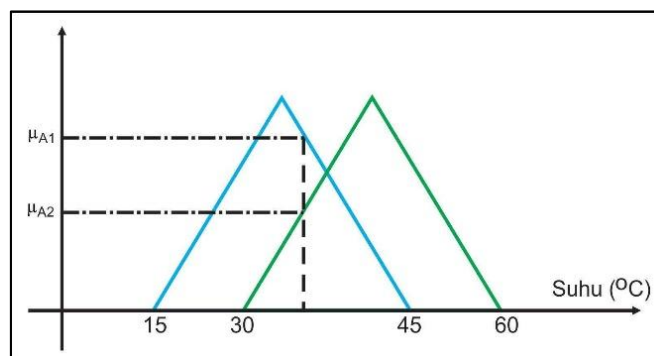
Gambar 8.7 Fungsi Keanggotaan Gaussian

Cara Kerja Kontrol Logika *Fuzzy*

Cara kerja kontrol logika *fuzzy* dapat dipisahkan menjadi beberapa tahapan operasional, yang meliputi:

1. Fuzzifikasi

Fuzzifikasi merupakan proses pengubahan nilai tegas yang ada ke dalam fungsi keanggotaan. Misal, diketahui suhu adalah 35°C, dan kategori suhu yang ada adalah dingin (biru) dan panas (hijau). Kita bisa menghitung derajat keanggotaan dingin dan panas dari suhu tersebut menggunakan rumus fungsi keanggotaan segitiga.



Gambar 8.8 Contoh Fungsi Keanggotaan Suhu

$$\mu_{A1} = \frac{c - x}{c - b}$$

$$\mu_{A1} = \frac{45 - 35}{45 - 30}$$

$$\mu_{A1} = \frac{2}{3}$$

$$\mu_{A2} = \frac{x - a}{b - a}$$

$$\mu_{A2} = \frac{35 - 30}{45 - 30}$$

$$\mu_{A2} = \frac{1}{3}$$

2. Dasar Aturan (*Rule Base*)

Dasar aturan pada logika *fuzzy* berupa sekumpulan pernyataan implikasi “JIKA-MAKA”, atau “*IF-THEN*”. Contohnya adalah, JIKA kebersihannya BAIK DAN pelayanannya BAIK, MAKA *rating* restoran TINGGI.

3. Penalaran (*Inference Machine*)

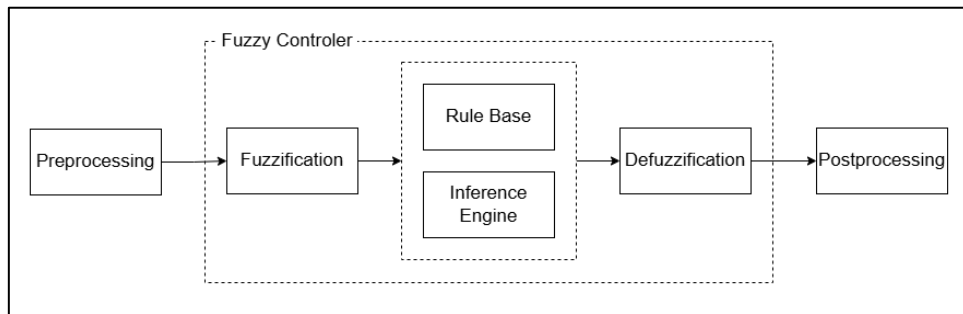
Inference machine digunakan untuk menentukan nilai output sebagai bentuk pengambilan keputusan. Dalam logika *fuzzy*, model penalaran yang biasa digunakan adalah penalaran min-max. Dalam penalaran tersebut, operasi min digunakan untuk tiap input (*antecedent*) pada aturan (*rule*) pada proses fuzzifikasi, kemudian dilakukan operasi max pada output (*consequent*) yang dihasilkan oleh proses tersebut untuk dilakukan defuzzifikasi.

4. Defuzzifikasi

Defuzzifikasi adalah proses pemetaan himpunan *fuzzy* ke himpunan *crisp*. Proses defuzzifikasi ini merupakan kebalikan dari fuzzifikasi yang dapat dilakukan dengan berbagai metode, seperti:

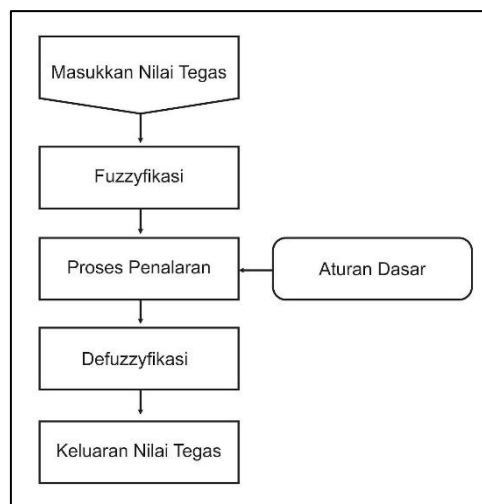
- Metode Max (*Maximum*)
- Metode Titik Tengah
- Metode Rata-Rata (*Average*)
- Metode Penjumlahan Titik Tengah (*Summing of Center Area*)
- Metode Titik Tengah Area Terbesar

Cara kerja kontrol logika *fuzzy* dapat digambarkan dengan diagram blok sebagai berikut:



Gambar 8.9 Blok Diagram Kontrol Logika *Fuzzy*

Sementara itu, kerangka operasional logika *fuzzy* dapat digambarkan dalam diagram berikut:



Gambar 8.10 Kerangka Operasional Kontrol Logika *Fuzzy*

III. LANGKAH PRAKTIKUM

Contoh studi kasus logika *fuzzy* yang paling umum adalah "*The Tipping Problem*". Logika *fuzzy* digunakan untuk membantu pelanggan menentukan persentase tip yang direkomendasikan berdasarkan pelayanan dan rasa makanan pada sebuah restoran. Permasalahan tersebut dapat dirangkum sebagai berikut:

1. *Antecedent* (Input)
 - a. Rasa makanan

- *Universe (range input)*: Seberapa enak rasa makanan dalam skala 0 – 10?
- *Fuzzy set*:

$$\begin{aligned}
 \text{tidak enak} &= \begin{cases} 1; & x < 0 \\ \frac{5-x}{5}; & 0 \leq x \leq 5 \\ 0; & x > 5 \end{cases} \\
 \text{biasa} &= \begin{cases} \frac{x}{4}; & 0 \leq x \leq 4 \\ 1; & 4 < x \leq 6 \\ \frac{10-x}{4}; & 6 < x \leq 10 \\ 0; & x < 0 \cup x > 10 \end{cases} \\
 \text{enak} &= \begin{cases} 0; & x < 5 \\ \frac{x-5}{5}; & 5 \leq x \leq 10 \\ 1; & x > 10 \end{cases}
 \end{aligned}$$

b. Pelayanan

- *Universe (range input)*: Seberapa baik pelayanan restoran dalam skala 0 – 10?
- *Fuzzy set*:

$$\begin{aligned}
 \text{buruk} &= \begin{cases} 1; & x < 0 \\ \frac{5-x}{5}; & 0 \leq x \leq 5 \\ 0; & x > 5 \end{cases} \\
 \text{biasa} &= \begin{cases} \frac{x}{3}; & 0 \leq x \leq 3 \\ 1; & 3 < x \leq 7 \\ \frac{10-x}{3}; & 7 < x \leq 10 \\ 0; & x < 0 \cup x > 10 \end{cases} \\
 \text{baik} &= \begin{cases} 0; & x < 5 \\ \frac{x-5}{5}; & 5 \leq x \leq 10 \\ 1; & x > 10 \end{cases}
 \end{aligned}$$

2. *Consequent (Output)*

a. Tip

- *Universe (range output)*: Berapa persentase tip yang harus dikeluarkan dari skala 0% – 25%?
- *Fuzzy set*:

$$\begin{aligned}
sedikit &= \begin{cases} 1; x < 0 \\ \frac{12-x}{12}; 0 \leq x \leq 12 \\ 0; x > 12 \end{cases} \\
sedang &= \begin{cases} 0; x < 0 \cup x > 25 \\ \frac{x}{12}; 0 \leq x \leq 12 \\ \frac{25-x}{13}; 12 < x \leq 25 \end{cases} \\
banyak &= \begin{cases} 0; x < 12 \\ \frac{x-12}{13}; 12 \leq x \leq 25 \\ 1; x > 25 \end{cases}
\end{aligned}$$

3. Rule Base (Dasar Aturan)

- JIKA rasa ENAK DAN pelayananan BAIK, MAKA tip BANYAK
- JIKA pelayanan biasa, MAKA tip SEDANG
- JIKA rasa BURUK ATAU pelayanan BURUK, MAKA tip SEDIKIT

Untuk membuat kontrol logika *fuzzy* berdasarkan kasus di atas menggunakan Python, kita membutuhkan beberapa *library*, yaitu *numpy* untuk membuat *array*, *scikit-fuzzy* untuk perhitungan *fuzzy* Mamdani, *matplotlib* untuk visualisasi fungsi keanggotaan, dan *streamlit* untuk tampilan UI.

Pertama, *install* semua *library* yang dibutuhkan melalui *command prompt* atau *terminal*, dengan mengetikkan perintah sebagai berikut:

```
pip install numpy scikit-fuzzy matplotlib streamlit
```

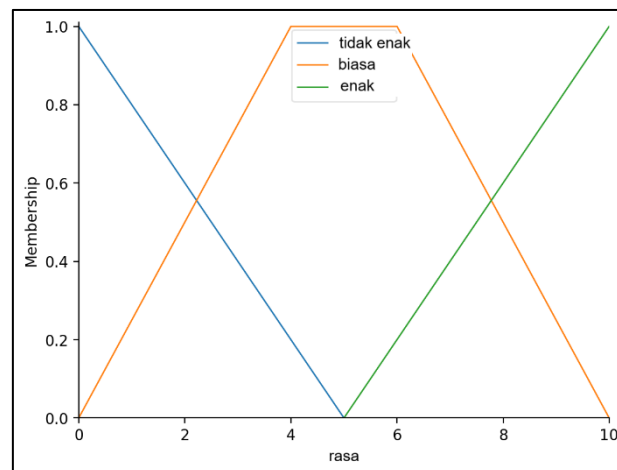
Setelah itu, buka aplikasi Visual Studio Code dan buat *file* Python baru. *Import library* yang dibutuhkan di bagian paling atas.

```
import streamlit as st
import numpy as np
import skfuzzy as fuzz
import skfuzzy.control as ctrl
import matplotlib.pyplot as plt
```

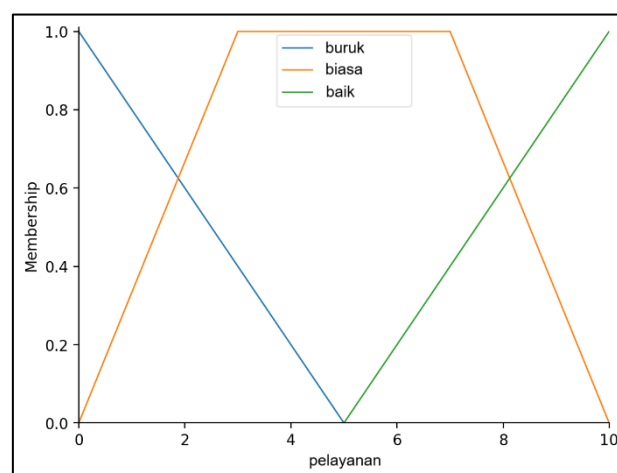
Kemudian, definisikan semua variabel input dan output beserta *range* yang sesuai. Perlu diingat, *array* pada Python memiliki interval setengah terbuka, sehingga batas atasnya perlu dilebihkan satu.

```
# Definisikan variabel fuzzy
rasa = ctrl.Antecedent(np.arange(0, 11, 1), 'rasa')
pelayanan = ctrl.Antecedent(np.arange(0, 11, 1), 'pelayanan')
tip = ctrl.Consequent(np.arange(0, 26, 1), 'tip')
```

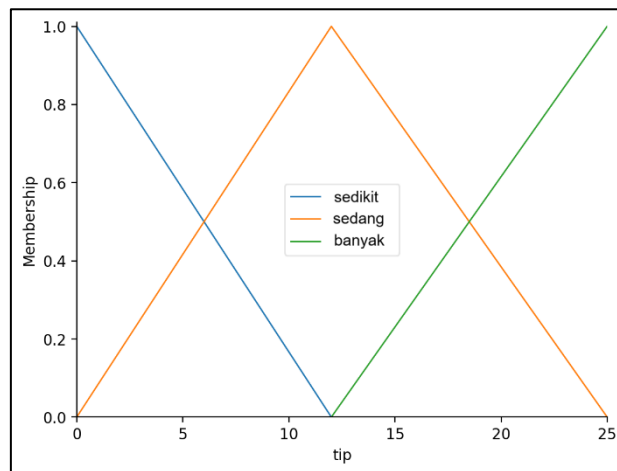
Tahap selanjutnya adalah mendefinisikan fungsi keanggotaan dari tiap variabel. Jika kita masukkan fungsi tersebut ke dalam sebuah grafik, fungsi keanggotaan tiap variabel dapat kita lihat pada gambar-gambar berikut:



Gambar 8.11 Grafik Fungsi Keanggotaan Rasa



Gambar 8.12 Grafik Fungsi Keanggotaan Pelayanan



Gambar 8.13 Grafik Fungsi Keanggotaan Tip

Definisikan fungsi keanggotaan dengan bentuk yang sesuai. Fungsi segitiga memiliki 3 titik (a, b, c), sedangkan fungsi trapesium memiliki 4 titik (a, b, c, d). Masukkan semua titik secara berurutan.

```
# Definisikan fungsi keanggotaan dengan bentuk trapesium dan segitiga
rasa['tidak_enak'] = fuzz.trimf(rasa.universe, [0, 0, 5])
rasa['biasa'] = fuzz.trapmf(rasa.universe, [0, 4, 6, 10])
rasa['enak'] = fuzz.trimf(rasa.universe, [5, 10, 10])
pelayanan['buruk'] = fuzz.trimf(pelayanan.universe, [0, 0, 5])
pelayanan['biasa'] = fuzz.trapmf(pelayanan.universe, [0, 3, 7, 10])
pelayanan['baik'] = fuzz.trimf(pelayanan.universe, [5, 10, 10])
tip['sedikit'] = fuzz.trimf(tip.universe, [0, 0, 12])
tip['sedang'] = fuzz.trimf(tip.universe, [0, 12, 25])
tip['banyak'] = fuzz.trimf(tip.universe, [12, 25, 25])
```

Selanjutnya, definisikan dasar aturan sesuai dengan kasus. Gunakan “|” untuk mendefinisikan operator OR/ATAU, dan “&” untuk operator AND/DAN. Pisahkan input dan output dengan tanda koma. Kemudian, susun aturan-aturan tersebut menjadi sebuah sistemn kontrol menggunakan fungsi `ctrl.ControlSystem` dan simulasikan sistem tersebut berdasarkan input yang akan diberikan oleh pengguna menggunakan fungsi `ctrl.ControlSystemSimulation`.

```
# Buat aturan fuzzy
```

```

rules = [
    ctrl.Rule(rasa['tidak_enak'] | pelayanan['buruk'], tip['sedikit']),
    ctrl.Rule(pelayanan['biasa'], tip['sedang']),
    ctrl.Rule(rasa['enak'] & pelayanan['baik'], tip['banyak'])
]
# Sistem kontrol berbasis aturan
tipping_ctrl = ctrl.ControlSystem(rules)
tipping = ctrl.ControlSystemSimulation(tipping_ctrl)

```

Jika sistem kontrol logika *fuzzy* sudah dibuat, saatnya membuat tampilan UI untuk input dari pengguna. Gunakan fungsi `st.write` untuk menulis dengan prinsip *formatting* seperti *markdown*. Untuk input, bisa gunakan `st.slider` maupun `st.number_input`. Masukkan nilai input pengguna ke dalam input sistem logika *fuzzy*, kemudian hitung inferensi berdasarkan input tersebut.

```

# Judul halaman
st.title("Sistem Logika Fuzzy dengan Streamlit dan scikit-fuzzy")

st.write("### Input Nilai")
# Input nilai dari pengguna melalui UI
rasa_value = st.slider("Rasa (0-10):", 0.0, 10.0, 5.0, 0.5)
pelayanan_value = st.slider("Pelayanan (0-10):", 0.0, 10.0, 5.0, 0.5)

# Masukkan nilai input ke sistem fuzzy
tipping.input['rasa'] = rasa_value
tipping.input['pelayanan'] = pelayanan_value

# Hitung hasil inferensi
tipping.compute()

```

Tampilkan hasil ke pengguna melalui UI. Tampilkan juga fungsi keanggotaan dalam bentuk grafik menggunakan *library* `pyplot`.

```

st.write('### Hasil')
# Tampilkan hasil ke pengguna
st.write(f"Rasa: {rasa_value}")
st.write(f"Pelayanan: {pelayanan_value}")
st.write(f"Tip: {tipping.output['tip']:.2f}%")

def plot_and_display_membership(variable, title):
    st.write(f"### {title}")
    fig, ax = plt.subplots()

```

```
variable.view(ax=ax)
st.pyplot(plt.gcf())

plot_and_display_membership(rasa, "Fungsi Keanggotaan Rasa")
plot_and_display_membership(pelayanan, "Fungsi Keanggotaan Pelayanan")
plot_and_display_membership(tip, "Fungsi Keanggotaan Tip")
```

IV. TUGAS PRAKTIKUM

1. Buatlah contoh studi kasus logika *fuzzy* dengan ketentuan sebagai berikut:
 - a. Gunakan Python untuk merancang sistem kontrol logika *fuzzy*
 - b. Sistem memiliki minimal 3 input dan 1 output
 - c. Tiap variabel memiliki himpunan *fuzzy* yang terdiri dari minimal 3 fungsi keanggotaan
 - d. Jumlah aturan minimal 10
 - e. Buat tampilan UI-nya menggunakan *library* streamlit

REFERENSI

- Athiyah, U., Handayani, A. P., Aldean, M. Y., Putra, N. P., & Ramadhani, R. (2021). Sistem Inferensi Fuzzy: Pengertian, Penerapan, dan Manfaatnya. *Journal of Dinda : Data Science, Information Technology, and Data Analytics*, 1(2), 73–76. <https://doi.org/10.20895/dinda.v1i2.201>
- Bozorg-Haddad, O., Zolghadr-Asli, B., & Loáiciga, H. A. (2021). A Handbook on multi-attribute decision-making methods. In *A Handbook on Multi-Attribute Decision-Making Methods* (Issue March). <https://doi.org/10.1002/9781119563501>
- Sasongko, A., Astuti, I. F., & Maharani, S. (2017). Pemilihan Karyawan Baru Dengan Metode AHP (Analytic Hierarchy Process). *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, 12(2), 88. <https://doi.org/10.30872/jim.v12i2.650>
- Sri, K., Hartati, S., Harjoko, A., & Wardoyo, R. (2006). Fuzzy Multi-Attribute Decision Making (Fuzzy MADM). *Graha Ilmu. Yogyakarta*.
- Zadeh, L. A. (2013). Fuzzy logic. *Computational Complexity: Theory, Techniques, and Applications*, 9781461418, 1177–1200. https://doi.org/10.1007/978-1-4614-1800-9_73