

NAMA : AINI AMALIA
NIM : 221011400621

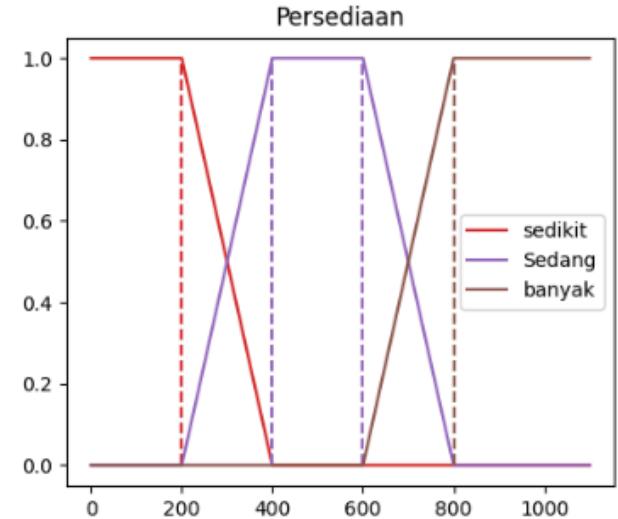
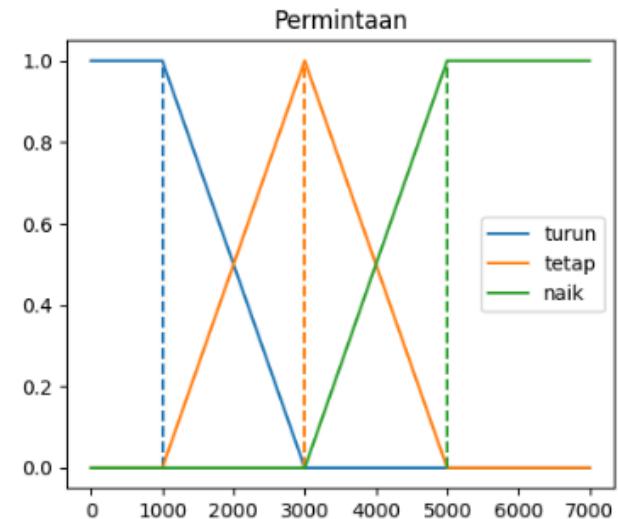
kecerdasan buatan
perhitungan manual dan sourcode

SOAL FUZZY

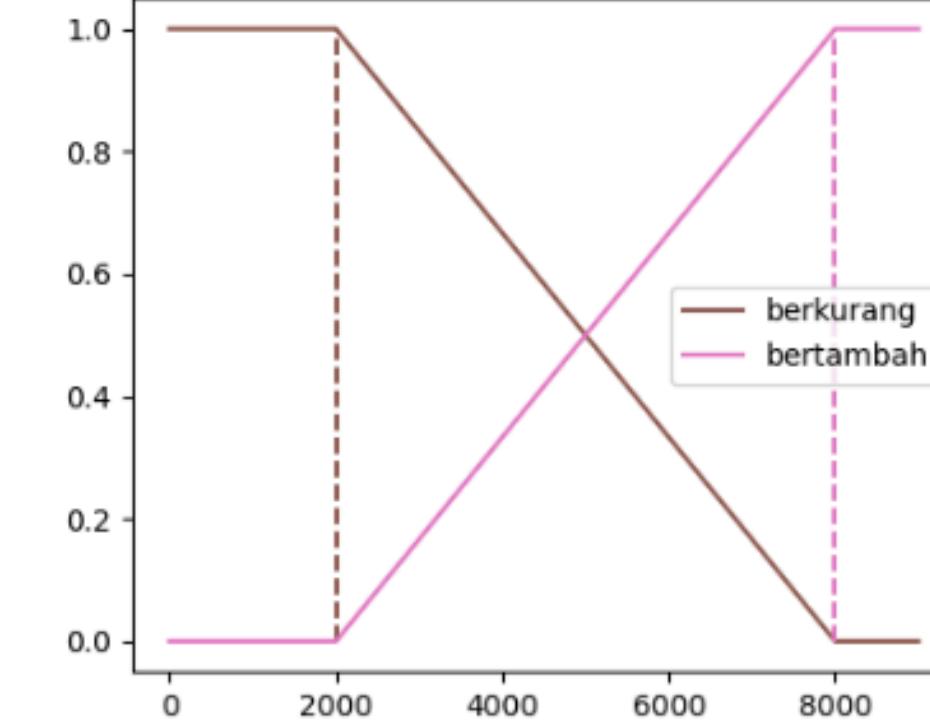
Soal Latihan

1. Buat Implementasi dengan menggunakan python untuk menghitung Fuzzy Inference System berikut dengan metode tsukamoto

In [21]:



Produksi



Rule:

- Jika Permintaan **Turun** dan Persediaan **Sedikit** maka produksi **Bertambah**
- Jika Permintaan **Turun** dan Persediaan **Sedang** maka produksi **Berkurang**
- Jika Permintaan **Turun** dan Persediaan **Banyak** maka produksi **Berkurang**
- Jika Permintaan **Tetap** dan Persediaan **Sedikit** maka produksi **Bertambah**
- Jika Permintaan **Tetap** dan Persediaan **Sedang** maka produksi **Berkurang**
- Jika Permintaan **Tetap** dan Persediaan **Banyak** maka produksi **Berkurang**
- Jika Permintaan **Naik** dan Persediaan **Sedikit** maka produksi **Bertambah**
- Jika Permintaan **Naik** dan Persediaan **Sedang** maka produksi **Bertambah**
- Jika Permintaan **Naik** dan Persediaan **Banyak** maka produksi **Berkurang**

2. Buat Slide yang berisikan contoh penghitungan manual dari soal diatas!

PERHITUNGAN MANUAL

1. Diberikan:

- Input Permintaan (Demand): 2500
- Input Persediaan (Stock): 300
- Output yang dihitung (Produksi):

menggunakan fungsi keanggotaan triangular untuk setiap variabel fuzzy (Permintaan, Persediaan, dan Produksi).

Permintaan memiliki kategori Turun dan Naik.

Persediaan memiliki kategori Sedikit dan Banyak.

Untuk input Permintaan=2500\text{Permintaan} = 2500Permintaan=2500 dan

Persediaan=300\text{Persediaan} = 300Persediaan=300, kita menghitung derajat keanggotaan untuk setiap kategori menggunakan rumus-rumus fungsi keanggotaan.

PERHITUNGAN MANUAL

2. Fungsi Keanggotaan

a) Permintaan

- Turun: $\mu_{\text{Turun}}(x) = \frac{4000-x}{4000-1000}$ untuk $1000 \leq x \leq 4000$
- Naik: $\mu_{\text{Naik}}(x) = \frac{x-1000}{4000-1000}$ untuk $1000 \leq x \leq 4000$

Dengan $x = 2500$:

$$\mu_{\text{Turun}}(2500) = \frac{4000 - 2500}{4000 - 1000} = \frac{1500}{3000} = 0.5$$

$$\mu_{\text{Naik}}(2500) = \frac{2500 - 1000}{4000 - 1000} = \frac{1500}{3000} = 0.5$$

Permintaan berada di antara kategori Turun dan Naik dengan keanggotaan masing-masing

0.5. Demikian pula,

Persediaan\text{Persediaan}Persediaan berada di antara kategori Sedikit dan Banyak.

b) Persediaan

- Sedikit: $\mu_{\text{Sedikit}}(x) = \frac{600-x}{600-100}$ untuk $100 \leq x \leq 600$
- Banyak: $\mu_{\text{Banyak}}(x) = \frac{x-100}{600-100}$ untuk $100 \leq x \leq 600$

Dengan $x = 300$:

$$\mu_{\text{Sedikit}}(300) = \frac{600 - 300}{600 - 100} = \frac{300}{500} = 0.6$$

$$\mu_{\text{Banyak}}(300) = \frac{300 - 100}{600 - 100} = \frac{200}{500} = 0.4$$

PERHITUNGAN MANUAL

3. Aturan Fuzzy

Aturan yang digunakan sesuai soal:

1. Jika Permintaan Turun DAN Persediaan Sedikit, maka Produksi Bertambah.

$$\alpha_1 = \min(\mu_{\text{Turun}}, \mu_{\text{Sedikit}}) = \min(0.5, 0.6) = 0.5$$

$$\text{Produksi } z_1 = 2000 + 0.5 \times (7000 - 2000) = 4500.$$

2. Jika Permintaan Turun DAN Persediaan Banyak, maka Produksi Berkurang.

$$\alpha_2 = \min(\mu_{\text{Turun}}, \mu_{\text{Banyak}}) = \min(0.5, 0.4) = 0.4$$

$$\text{Produksi } z_2 = 7000 - 0.4 \times (7000 - 2000) = 5000.$$

3. Jika Permintaan Naik DAN Persediaan Sedikit, maka Produksi Bertambah.

$$\alpha_3 = \min(\mu_{\text{Naik}}, \mu_{\text{Sedikit}}) = \min(0.5, 0.6) = 0.5$$

$$\text{Produksi } z_3 = 2000 + 0.5 \times (7000 - 2000) = 4500.$$

4. Jika Permintaan Naik DAN Persediaan Banyak, maka Produksi Berkurang.

$$\alpha_4 = \min(\mu_{\text{Naik}}, \mu_{\text{Banyak}}) = \min(0.5, 0.4) = 0.4$$

$$\text{Produksi } z_4 = 7000 - 0.4 \times (7000 - 2000) = 5000.$$

- Aturan : "Jika Permintaan Turun DAN Persediaan Sedikit, maka Produksi Bertambah."

Setiap aturan menghasilkan nilai α yang akan digunakan untuk menghitung output fuzzy z (dalam hal ini, nilai produksi).

PERHITUNGAN MANUAL

4. Defuzzifikasi

Menggunakan metode rata-rata tertimbang:

$$z = \frac{\sum(\alpha_i \cdot z_i)}{\sum \alpha_i}$$

$$z = \frac{(0.5 \cdot 4500) + (0.4 \cdot 5000) + (0.5 \cdot 4500) + (0.4 \cdot 5000)}{0.5 + 0.4 + 0.5 + 0.4}$$

$$z = \frac{2250 + 2000 + 2250 + 2000}{1.8} = \frac{8500}{1.8} \approx 4722.22$$

Setelah semua aturan dihitung, kita melakukan defuzzifikasi untuk mendapatkan hasil crisp (tegas).

menjumlahkan semua kontribusi aturan ($\alpha \cdot z$) lalu membaginya dengan total α . Ini menghasilkan nilai final produksi, yaitu 4722 unit.

JAWABAN BESERTA SOURCODE

```
[1]: import numpy as np
import skfuzzy as fuzz

# Fungsi Keanggotaan Permintaan
def permintaan(x):
    turun = fuzz.trimf(x, [0, 0, 3000])
    tetap = fuzz.trimf(x, [2000, 3000, 4000])
    naik = fuzz.trimf(x, [3000, 6000, 6000])
    return turun, tetap, naik

# Fungsi Keanggotaan Persediaan
def persediaan(x):
    sedikit = fuzz.trimf(x, [0, 0, 500])
    sedang = fuzz.trimf(x, [200, 500, 800])
    banyak = fuzz.trimf(x, [500, 1000, 1000])
    return sedikit, sedang, banyak

# Fungsi Keanggotaan Produksi
def produksi(x):
    berkurang = fuzz.trimf(x, [0, 0, 5000])
    bertambah = fuzz.trimf(x, [2000, 10000, 10000])
    return berkurang, bertambah

# Inferensi Tsukamoto
def inferensi_tsukamoto(permintaan_input, persediaan_input):
    # Domain variabel
    x_permintaan = np.arange(0, 7001, 1)
    x_persediaan = np.arange(0, 1001, 1)
    x_produksi = np.arange(0, 10001, 1)

    # Keanggotaan fuzzy
    turun, tetap, naik = permintaan(x_permintaan)
    sedikit, sedang, banyak = persediaan(x_persediaan)
    berkurang, bertambah = produksi(x_produksi)
```

hasil Output produksi (unit): 4677.0833333331

```
# Derajat keanggotaan input
permintaan_turun = fuzz.interp_membership(x_permintaan, turun, permintaan_input)
permintaan_tetap = fuzz.interp_membership(x_permintaan, tetap, permintaan_input)
permintaan_naik = fuzz.interp_membership(x_permintaan, naik, permintaan_input)

persediaan_sedikit = fuzz.interp_membership(x_persediaan, sedikit, persediaan_input)
persediaan_sedang = fuzz.interp_membership(x_persediaan, sedang, persediaan_input)
persediaan_banyak = fuzz.interp_membership(x_persediaan, banyak, persediaan_input)

# Inferensi aturan
rules = []
rules.append((min(permintaan_turun, persediaan_sedikit), bertambah))
rules.append((min(permintaan_turun, persediaan_sedang), berkurang))
rules.append((min(permintaan_turun, persediaan_banyak), berkurang))
rules.append((min(permintaan_tetap, persediaan_sedikit), bertambah))
rules.append((min(permintaan_tetap, persediaan_sedang), berkurang))
rules.append((min(permintaan_tetap, persediaan_banyak), berkurang))
rules.append((min(permintaan_naik, persediaan_sedikit), bertambah))
rules.append((min(permintaan_naik, persediaan_sedang), bertambah))
rules.append((min(permintaan_naik, persediaan_banyak), berkurang))

# Perhitungan output
produk = []
for rule_strength, output in rules:
    z = fuzz.defuzz(x_produksi, output, 'centroid')
    produk.append(rule_strength * z)

hasil = sum(produk) / sum(rule[0] for rule in rules)
return hasil

# Contoh Penghitungan
permintaan_input = 2500 # Contoh input permintaan
persediaan_input = 300 # Contoh input persediaan
produksi_output = inferensi_tsukamoto(permintaan_input, persediaan_input)

print("Output produksi (unit):", produksi_output)
Output produksi (unit): 4677.0833333331
```

MEMBUAT GRAFIK MENGGUNAKAN CLASS DAN FUNCTION

potongan Sourcode

```
# Define the ranges for each variable
x_demand = np.linspace(1000, 5000, 500) # Range for Demand
x_stock = np.linspace(100, 600, 500) # Range for Stock
fuzzy_values = np.linspace(0, 1, 500) # Range for fuzzy outputs (Production)

# Calculate fuzzy membership values
demand_decrease = [demand.decrease(x) for x in x_demand]
demand_increase = [demand.increase(x) for x in x_demand]

stock_a_few = [stock.a_few(x) for x in x_stock]
stock_a_lot = [stock.a_lot(x) for x in x_stock]

production_reduce = [production.reduce(f) for f in fuzzy_values]
production_add = [production.add(f) for f in fuzzy_values]

# Create subplots for the variables
fig, axs = plt.subplots(3, 1, figsize=(6, 12))

# Demand Plot
axs[0].plot(x_demand, demand_decrease, label="Turun", color="blue", linestyle="--")
axs[0].plot(x_demand, demand_increase, label="Naik", color="orange", linestyle="-")
axs[0].set_title("Permintaan (Demand)")
axs[0].legend(loc="best")
axs[0].grid(True)
axs[0].set_xlabel("Permintaan")
axs[0].set_ylabel("Derajat Keanggotaan")

# Stock Plot
axs[1].plot(x_stock, stock_a_few, label="Sedikit", color="green", linestyle="--")
axs[1].plot(x_stock, stock_a_lot, label="Banyak", color="red", linestyle="-")
axs[1].set_title("Persediaan (Stock)")
axs[1].legend(loc="best")
axs[1].grid(True)
axs[1].set_xlabel("Persediaan")
axs[1].set_ylabel("Derajat Keanggotaan")

# Production Plot
axs[2].plot(fuzzy_values, production_reduce, label="Berkurang", color="purple", linestyle="--")
axs[2].plot(fuzzy_values, production_add, label="Bertambah", color="brown", linestyle="-")
axs[2].set_title("Produksi (Production)")
axs[2].legend(loc="best")
axs[2].grid(True)
axs[2].set_xlabel("Fuzzy Value")
axs[2].set_ylabel("Produksi")

# Display the plots
plt.tight_layout()
plt.show()

# Calculate fuzzy membership values
demand_decrease = [demand.decrease(x) for x in x_demand]
demand_increase = [demand.increase(x) for x in x_demand]

stock_a_few = [stock.a_few(x) for x in x_stock]
stock_a_lot = [stock.a_lot(x) for x in x_stock]

production_reduce = [production.reduce(f) for f in fuzzy_values]
production_add = [production.add(f) for f in fuzzy_values]

# Create subplots for the variables
fig, axs = plt.subplots(3, 1, figsize=(6, 12))

# Demand Plot
axs[0].plot(x_demand, demand_decrease, label="Turun", color="blue", linestyle="--")
axs[0].plot(x_demand, demand_increase, label="Naik", color="orange", linestyle="-")
axs[0].set_title("Permintaan (Demand)")
axs[0].legend(loc="best")
axs[0].grid(True)
axs[0].set_xlabel("Permintaan")
axs[0].set_ylabel("Derajat Keanggotaan")

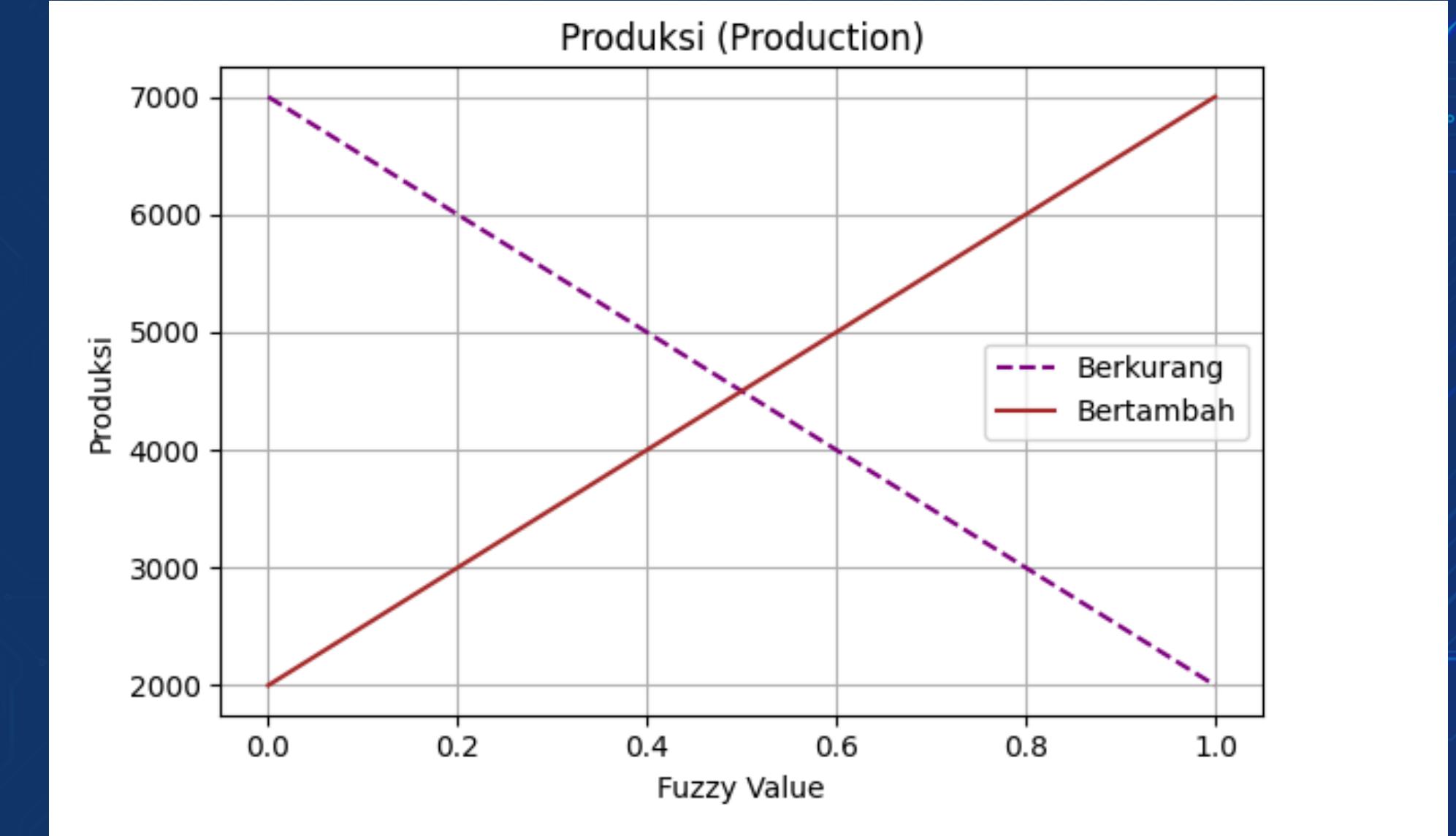
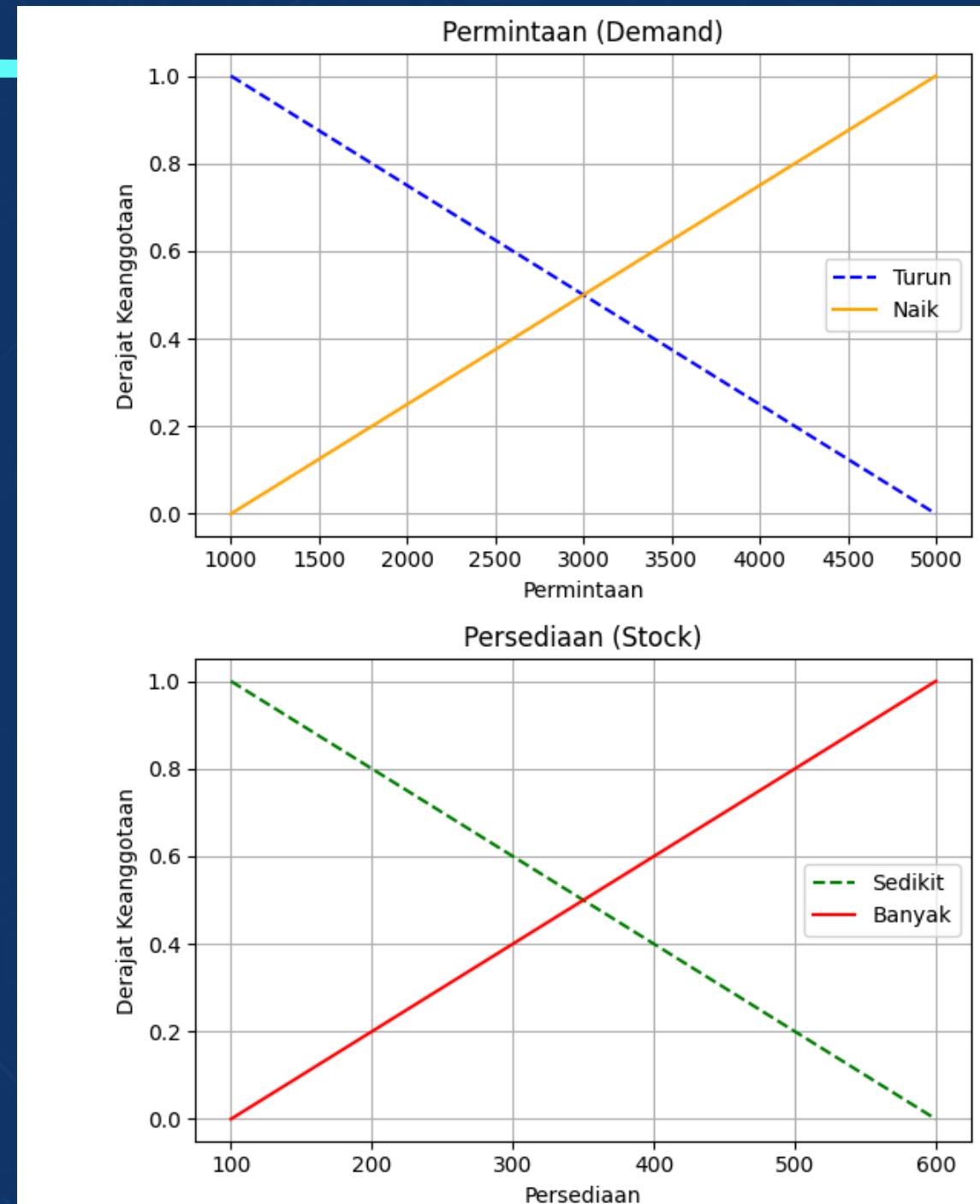
# Stock Plot
axs[1].plot(x_stock, stock_a_few, label="Sedikit", color="green", linestyle="--")
axs[1].plot(x_stock, stock_a_lot, label="Banyak", color="red", linestyle="-")
axs[1].set_title("Persediaan (Stock)")
axs[1].legend(loc="best")
axs[1].grid(True)
axs[1].set_xlabel("Persediaan")
axs[1].set_ylabel("Derajat Keanggotaan")

# Production Plot
axs[2].plot(fuzzy_values, production_reduce, label="Berkurang", color="purple", linestyle="--")
axs[2].plot(fuzzy_values, production_add, label="Bertambah", color="brown", linestyle="-")
axs[2].set_title("Produksi (Production)")
axs[2].legend(loc="best")
axs[2].grid(True)
axs[2].set_xlabel("Fuzzy Value")
axs[2].set_ylabel("Produksi")

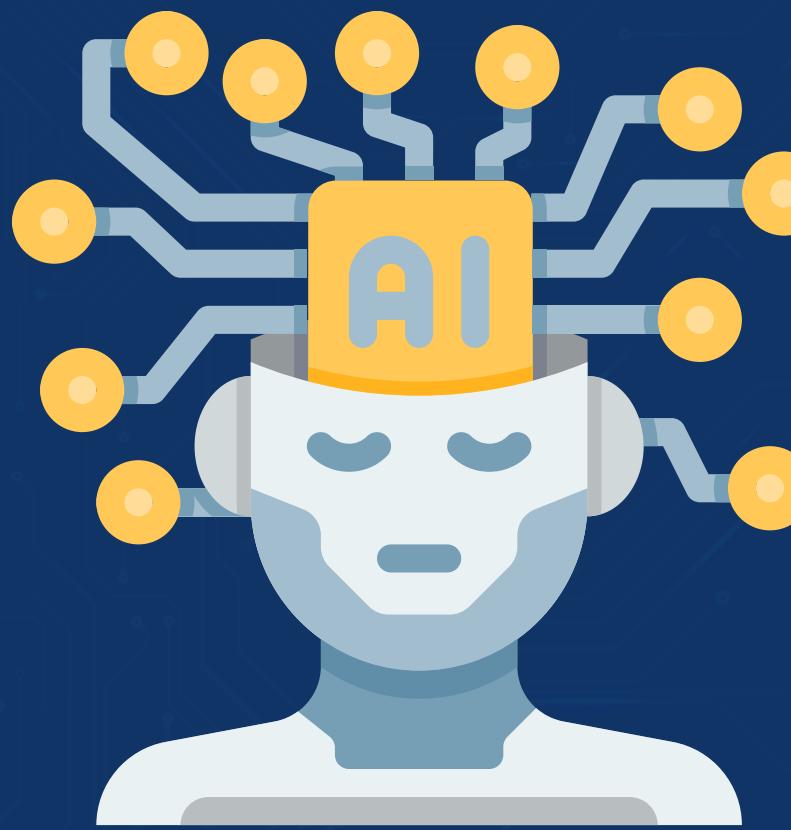
# Display the plots
plt.tight_layout()
plt.show()
```

MEMBUAT GRAFIK MENGGUNAKAN CLASS DAN FUNCTION

Hasil Grafik



KESIMPULAN



Metode Tsukamoto bekerja dengan langkah-langkah ini:

1. Menghitung derajat keanggotaan input (fuzzyifikasi).
2. Menggunakan aturan fuzzy untuk menghasilkan kekuatan aturan (a).
3. Menghitung output fuzzy untuk setiap aturan (z).
4. Melakukan defuzzifikasi untuk mendapatkan hasil crisp (tegas).

Dengan ini, kita memperoleh output produksi berdasarkan input Permintaan dan Persedian. Hasilnya sederhana namun mencerminkan proses logika fuzzy.

TERIMAKASIH