NAMA:IOS WIDRIA SARAGIH
NIM:2210114000558

Perhitungan manual dan Sourcode

Kelas:05TPLM007
LinkGithub:
https://github.com/iossaragih/ioswidriasaragih_tugas

1. Langkah Perhitungan Manual

1. **Fungsi Keanggotaan untuk Demand (Permintaan):**
   - **Decrease (Turun):** $\text{Decrease}(x) = \begin{cases} 1, & \text{jika } x \leq 1000 \\ \frac{5000-x}{5000-1000}, & \text{jika } 1000 < x < 5000 \\ 0, & \text{jika } x \geq 5000 \end{cases}$
   - **Increase (Naik):** $\text{Increase}(x) = \begin{cases} 0, & \text{jika } x \leq 1000 \\ \frac{x-1000}{5000-1000}, & \text{jika } 1000 < x < 5000 \\ 1, & \text{jika } x \geq 5000 \end{cases}$

   Contoh Hitung Manual:
   - Untuk $x = 3000$:
     - $\text{Decrease}(3000) = \frac{5000-3000}{4000} = 0.5$
     - $\text{Increase}(3000) = \frac{3000-1000}{4000} = 0.5$

2. **Fungsi Keanggotaan untuk Stock (Persediaan):**
   - **A Few (Sedikit):** $\text{A Few}(x) = \begin{cases} 1, & \text{jika } x \leq 100 \\ \frac{600-x}{600-100}, & \text{jika } 100 < x < 600 \\ 0, & \text{jika } x \geq 600 \end{cases}$
   - **A Lot (Banyak):** $\text{A Lot}(x) = \begin{cases} 0, & \text{jika } x \leq 100 \\ \frac{x-100}{600-100}, & \text{jika } 100 < x < 600 \\ 1, & \text{jika } x \geq 600 \end{cases}$

   Contoh Hitung Manual:
   - Untuk $x = 300$:
     - $\text{A Few}(300) = \frac{600-300}{500} = 0.6$
     - $\text{A Lot}(300) = \frac{300-100}{500} = 0.4$

Langkah Perhitungan Manual

3. **Fungsi Keanggotaan untuk Production (Produksi):**
   - **Reduce (Mengurangi):** $\text{Reduce}(f) = 7000 - f \times (7000 - 2000)$
   - **Add (Menambah):** $\text{Add}(f) = f \times (7000 - 2000) + 2000$

   Contoh Hitung Manual:
   - Untuk $f = 0.5$:
     - $\text{Reduce}(0.5) = 7000 - 0.5 \times 5000 = 4500$
     - $\text{Add}(0.5) = 0.5 \times 5000 + 2000 = 4500$

Aturan Fuzzy:

- (Turun, Banyak) ⇒ Berkurang
- (Turun, Sedikit) ⇒ Berkurang
- (Naik, Banyak) ⇒ Bertambah
- (Naik, Sedikit) ⇒ Bertambah

Evaluasi Aturan:

- Untuk setiap aturan:
  - Derajat keanggotaan diambil sebagai nilai minimum antar premis.
  - Output dihitung berdasarkan fungsi rev_down atau rev_up.

```python
class Fuzzy:
    def __init__(self):  # Perbaikan pada nama metode konstruktor
        self.min = 0
        self.max = 0

    def down(self, x):
        return max(0, (self.max - x) / (self.max - self.min))

    def up(self, x):
        return max(0, (x - self.min) / (self.max - self.min))

    def rev_down(self, f):
        return self.max - f * (self.max - self.min)

    def rev_up(self, f):
        return f * (self.max - self.min) + self.min


class Demand(Fuzzy):
    def __init__(self):  # Perbaikan pada nama metode konstruktor
        super().__init__()  # Memanggil konstruktor dari kelas induk
        self.min = 1000
        self.max = 5000

    def decrease(self, x):
        if x <= self.min:
            return 1
        elif x >= self.max:
            return 0
        return self.down(x)

    def increase(self, x):
        if x <= self.min:
            return 0
        elif x >= self.max:
            return 1
        return self.up(x)


class Stock(Fuzzy):
    def __init__(self):  # Perbaikan pada nama metode konstruktor
        super().__init__()  # Memanggil konstruktor dari kelas induk
        self.min = 100
        self.max = 600
```

```python
    def a_few(self, x):
        if x <= self.min:
            return 1
        elif x >= self.max:
            return 0
        return self.down(x)

    def a_lot(self, x):
        if x <= self.min:
            return 0
        elif x >= self.max:
            return 1
        return self.up(x)


class Production(Fuzzy):
    def __init__(self):  # Perbaikan pada nama metode konstruktor
        super().__init__()  # Memanggil konstruktor dari kelas induk
        self.min = 2000
        self.max = 7000

    def reduce(self, fuzzy_value):
        return self.rev_down(fuzzy_value)

    def add(self, fuzzy_value):
        return self.rev_up(fuzzy_value)


# Instantiate fuzzy sets
demand = Demand()
stock = Stock()
production = Production()

# 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()
```