



SPK Pemilihan Rekomendasi Laptop dengan menggabungkan Metode Fuzzy-TOPSIS: Menangani Ketidakpastian Linguistik dalam MCDM

📖 Konsep Dasar

Fuzzy-TOPSIS adalah pengembangan metode TOPSIS klasik dengan mengintegrasikan **Fuzzy Logic** untuk menangani ketidakpastian dan subjektivitas dalam pengambilan keputusan.

Masalah TOPSIS Klasik

Dalam TOPSIS tradisional, semua nilai kriteria harus berupa angka pasti (crisp values):

- RAM: 8 GB = 8.0
- Processor Speed: 2.5 GHz = 2.5
- Harga: Rp 7.000.000 = 7000000

Problem: Penilaian manusia sering tidak pasti!

- "RAM 8GB itu cukup... atau kurang ya?"
- "Processor Intel i5 termasuk bagus atau standar?"
- "Harga 7 juta itu mahal atau murah untuk mahasiswa?"

Solusi Fuzzy Logic

Fuzzy Logic memungkinkan representasi nilai menggunakan **variabel linguistik**:

- RAM: "Sedang" atau "Tinggi"
- Processor: "Cukup Baik"
- Harga: "Terjangkau"

Setiap variabel linguistik dipetakan ke **fuzzy membership function** (fungsi keanggotaan) yang menunjukkan derajat keanggotaan nilai dalam suatu kategori.

Contoh Fuzzy Membership untuk RAM:

RAM 8GB memiliki:

- Membership "Sedang": 0.6 (60%)
- Membership "Tinggi": 0.4 (40%)

Artinya: RAM 8GB sebagian termasuk kategori "Sedang", sebagian "Tinggi".

Novelty Elements

1. Handling Subjektivitas

Fuzzy-TOPSIS mengakomodasi penilaian subjektif ahli atau pengguna yang tidak selalu pasti. Cocok untuk kriteria kualitatif seperti "kualitas layar" atau "desain ergonomis".

2. Linguistic Variables

Menggunakan istilah natural language yang lebih intuitif:

- Very Low, Low, Medium, High, Very High
- Sangat Buruk, Buruk, Cukup, Baik, Sangat Baik

3. Robust Decision Making

Hasil lebih stabil terhadap variasi input karena memodelkan ketidakpastian secara eksplisit.

Implementasi Fuzzy-TOPSIS

Langkah 1: Definisi Fuzzy Sets

```
# Contoh Triangular Fuzzy Number (TFN)
# Format: (lower, middle, upper)

fuzzy_ratings = {
    'Very Low': (0, 0, 2.5),
    'Low': (0, 2.5, 5),
    'Medium': (2.5, 5, 7.5),
    'High': (5, 7.5, 10),
    'Very High': (7.5, 10, 10)
}
```

Langkah 2: Konversi Nilai ke Fuzzy

```
# Input: RAM 8GB, Rating linguistik: "High"
ram_fuzzy = fuzzy_ratings['High'] # (5, 7.5, 10)

# Input: Harga Rp 7jt, Rating: "Medium" (karena range mahasiswa)
harga_fuzzy = fuzzy_ratings['Medium'] # (2.5, 5, 7.5)
```

Langkah 3: Fuzzy TOPSIS Calculation

```
import numpy as np
from skfuzzy import trimf

# Normalisasi fuzzy matrix
fuzzy_normalized = normalize_fuzzy_matrix(decision_matrix)

# Hitung fuzzy ideal solutions
A_plus = calculate_fuzzy_positive_ideal(fuzzy_normalized, weights)
A_minus = calculate_fuzzy_negative_ideal(fuzzy_normalized, weights)

# Distance calculation (menggunakan fuzzy distance metric)
distance_plus = fuzzy_distance(alternatives, A_plus)
distance_minus = fuzzy_distance(alternatives, A_minus)

# Closeness coefficient
closeness = distance_minus / (distance_plus + distance_minus)

# Ranking berdasarkan closeness
ranking = np.argsort(closeness)[::-1]
```

Langkah 4: Defuzzification

Konversi hasil fuzzy kembali ke nilai crisp untuk interpretasi:

```
# Centroid method
crisp_score = (lower + middle + upper) / 3
```

Contoh Aplikasi: SPK Laptop

Kriteria Fuzzy untuk Laptop:

Kriteria	Variabel Linguistik	Fuzzy Set (0-10)
Harga	Sangat Murah, Murah, Sedang, Mahal	(0,0,3), (1,3,5), (3,5,7), (5,7,10)

Kriteria	Variabel Linguistik	Fuzzy Set (0-10)
Processor	Lemah, Cukup, Bagus, Sangat Bagus	(0,2,4), (2,4,6), (4,6,8), (6,8,10)
RAM	Rendah, Sedang, Tinggi, Sangat Tinggi	(0,2,4), (3,5,7), (6,8,10), (8,10,10)

Laptop A - Asus VivoBook:

- Harga: Rp 6jt → "Sedang" → (3, 5, 7)
- Processor: i3 Gen 11 → "Cukup" → (2, 4, 6)
- RAM: 8GB → "Sedang" → (3, 5, 7)

Laptop B - Lenovo ThinkPad:

- Harga: Rp 12jt → "Mahal" → (5, 7, 10)
- Processor: i7 Gen 12 → "Sangat Bagus" → (6, 8, 10)
- RAM: 16GB → "Tinggi" → (6, 8, 10)

Fuzzy-TOPSIS akan menghitung ranking dengan mempertimbangkan overlap dan ketidakpastian dalam penilaian.

Kelebihan Fuzzy-TOPSIS

1. **Realistis:** Mencerminkan ketidakpastian pengambilan keputusan manusia
2. **Flexible:** Dapat menangani kriteria kualitatif dan kuantitatif
3. **Robust:** Hasil stabil terhadap small variations dalam input
4. **Akademik Proven:** Banyak publikasi jurnal mendukung validitas metode
5. **Implementation Ready:** Library Python tersedia (scikit-fuzzy, numpy)

Kekurangan & Tantangan

1. **Complexity:** Lebih kompleks dari TOPSIS klasik
2. **Fuzzy Set Design:** Membutuhkan expert knowledge untuk mendefinisikan membership functions
3. **Computational Cost:** Lebih lambat karena operasi fuzzy
4. **Novelty Level:** Fuzzy-TOPSIS sudah cukup banyak diteliti (novelty sedang)

Rekomendasi Penggunaan

Gunakan Fuzzy-TOPSIS jika:

- ✓ Kriteria banyak yang subjektif (design, user experience, brand reputation)
- ✓ Input data dari survey dengan penilaian linguistik
- ✓ Perlu handling uncertainty dalam expert judgment
- ✓ Timeline 3-4 minggu (implementasi moderate)

Kombinasikan dengan strategi lain untuk novelty maksimal:

- Fuzzy-TOPSIS + Adaptive Weighting
- Fuzzy-TOPSIS + Sentiment Analysis

Referensi Implementasi

Python Libraries:

- `scikit-fuzzy` : Fuzzy logic operations
- `numpy` : Matrix calculations
- `pandas` : Data handling