

LAPORAN AKHIR

SISTEM REKOMENDASI PEMILIHAN MOBIL BERBASIS ALGORITMA FUZZY TOPSIS UNTUK MENINGKATKAN KEPUTUSAN KONSUMEN



Anggota Grup 2:

Nayasha Clarisa Dwisutrisna	00000056883
Ayu Febriana Lingga	00000057105
Fadhil Rahman Dwiputro	00000062832
Ville Jason Kannisto	00000063305

**KELAS IF541-A
SEMESTER GASAL 2023-2024**

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG**

2023

BAB 1

PROBLEM DESCRIPTION

1.1 Latar Belakang Masalah

Pada era saat ini, mobilitas merupakan sebuah aspek penting yang dilakukan oleh manusia setiap harinya. Berbagai macam cara untuk melakukan mobilitas dapat dilakukan, salah satunya adalah menggunakan kendaraan seperti sepeda, motor hingga mobil. Setiap tahunnya, terjadi peningkatan jualan kendaraan bermotor khususnya pada mobil [1]. Bahkan, menurut dari Gabungan Industri Kendaraan Bermotor Indonesia atau GAIKINDO, penjualan mobil pada bulan Januari hingga Oktober tahun 2021 mencapai angka 677.346 unit mobil. Sedangkan penjualan mobil pada bulan Januari hingga Oktober 2022 mencapai 822.013 unit [2]. Data tersebut menunjukkan bahwa masyarakat memiliki minat yang cukup tinggi untuk membeli mobil sebagai sarana untuk melakukan mobilitas.

Namun seiring berjalannya waktu, banyak sekali jenis mobil yang hadir di pasaran dengan harga, model hingga jenis bahan bakar yang beragam. Akibatnya, banyak calon pembeli yang mengalami kesulitan dalam memilih mobil yang cocok digunakan sebagai kendaraan pribadi. Sedangkan saat ini pemilihan mobil tidak lagi hanya berdasarkan pada pertimbangan konvensional seperti harga, model dan performa. Pemilihan mobil juga harus mempertimbangkan keamanan, kenyamanan, perawatan mobil hingga berbagai preferensi pribadi calon pembeli.

Dalam menghadapi kesulitan tersebut, diperlukan sebuah metode pendekatan yang dapat membantu calon pembeli dalam memilih mobil yang cocok dan tepat. Pendekatan yang diharapkan dapat menjawab kesulitan tersebut adalah algoritma *Fuzzy Based Topsis*. *Fuzzy Based Topsis* merupakan sebuah algoritma pengambilan keputusan multi-kriteria yang terdiri dari *fuzzy* dan *topsis* dan digunakan untuk memilih alternatif yang terbaik dari sekelompok alternatif berdasarkan pada jarak relatif positif dan negatif [3]. Algoritma *Fuzzy Based Topsis* menggunakan logika *fuzzy* untuk menangani ketidakpastian dan subjektivitas dalam melakukan evaluasi kriteria [4], serta metode ini menggunakan *topsis* untuk menentukan alternatif terbaik dengan melibatkan banyak alternatif pilihan dan berdasar pada kedekatan solusi ideal positif dan kerengangan solusi ideal negatif [5].

Dengan algoritma *Fuzzy Based Topsis*, calon pembeli mobil dapat melakukan pertimbangan sekaligus seperti harga, model, performa, keamanan,

kenyamanan dan preferensi pribadi. Oleh karena itu, penggunaan *Fuzzy Based Topsis* diharapkan dapat menjadi solusi inovatif agar calon pembeli dapat membuat keputusan yang tidak merugikan calon pembeli di masa yang akan datang.

1.2 Rumusan Masalah

Berdasarkan pada latar belakang yang telah dijabarkan, maka rumusan masalah pada penelitian ini sebagai berikut.

1. Bagaimana implementasi *Fuzzy Based Topsis* dalam memberikan rekomendasi mobil yang tepat?
2. Bagaimana efektivitas penggunaan algoritma *Fuzzy Based Topsis* dalam memberikan rekomendasi mobil?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini sebagai berikut.

1. Algoritma yang digunakan hanya terbatas pada algoritma *Fuzzy Based Topsis*
2. Dataset yang digunakan bersumber dari *kaggle* dengan judul "The Cars"

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut.

1. Mengetahui implementasi *Fuzzy Based Topsis* dalam memberikan rekomendasi mobil yang tepat
2. Mengetahui efektivitas penggunaan algoritma *Fuzzy Based Topsis* dalam memberikan rekomendasi mobil

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut.

1. Penelitian ini dapat memberikan wawasan bagi peneliti dan pembaca mengenai penggunaan algoritma *Fuzzy Based Topsis* dalam peningkatan keputusan.

2. Penelitian ini dapat menjadi sumber rujukan mengenai algoritma *Fuzzy Based Topsis* untuk penelitian dimasa yang akan datang.
3. Penelitian ini dapat memberikan kontribusi dalam cara peningkatan keputusan yang lebih efisien dengan algoritma *Fuzzy Based Topsis*

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada laporan tugas akhir ini sebagai berikut.

BAB I PROBLEM DESCRIPTION

Pada bab pertama terdiri dari enam bagian, yaitu latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan laporan.

BAB II THEORY-ALGORITHM-METHOD

Pada bab kedua berisi mengenai konsep yang mendukung penelitian ini, seperti *multi criteria decision making*, algoritma *fuzzy*, algoritma *fuzzy topsis*, dan algoritma *fuzzy AHP*.

BAB III SOLUTION

Pada bab ketiga berisi mengenai langkah-langkah yang dilakukan dalam menyelesaikan permasalahan dengan algoritma *fuzzy topsis*.

BAB IV DEMO-SIMULATION

Pada bab keempat berisi mengenai *snapshot* dari *demo* atau simulasi yang telah dilakukan pada penelitian ini.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

BAB 2

THEORY - ALGORITHM - METHOD

2.1 Multi Criteria Decision Making (MCDM)

MCDM atau Multi Criteria Decision Making merupakan salah satu metode pengambilan keputusan yang paling akurat dan sebuah revolusi pengambilan keputusan yang sudah cukup dikenal. Pertama kali dilakukannya pengambilan keputusan menggunakan MCDM yaitu oleh Benjamin Franklin ketika ia mempublikasi penelitiannya mengenai konsep moral aljabar. Selain itu, ilmuwan lain juga mencoba metode MCDM ini sebagai metode pengambilan keputusannya dalam model matematis sehingga dapat mengambil keputusan dari sebuah masalah dengan hasil preferensi alternatif. Metode MCDM ini memperhatikan beberapa kriteria untuk menemukan solusi terbaiknya seperti kriteria secara kualitatif dan kriteria secara kuantitatif. Dari kriteria yang dimiliki tersebut, akan diberikan bobot yang berbeda sebagai dasar dari sebuah kriteria kasus tersebut. Metode MCDM dapat digunakan untuk menyelesaikan masalah sehari-hari hingga masalah yang kompleks. Maka, dalam setiap pengambilan solusi dalam sebuah masalah diperlukan evaluasi kriteria yang tepat dan sesuai. MCDM telah menjadi bidang yang diteliti secara aktif hingga menghasilkan banyak artikel dan buku sejak tahun 1960. Metode ini dibangun untuk menentukan alternatif, mengklasifikasikan alternatif dalam kategori kecil, dan memberi peringkat dari alternatif menggunakan urutan preferensi secara subjektif. Sehingga, kegunaan MCDM ini dapat digunakan untuk mengatasi suatu permasalahan yang kompleks dengan cara memecah masalah tersebut menjadi bagian yang lebih kecil lagi. Bagian-bagian yang kecil tersebut akan disusun kembali untuk memberikan gambaran solusi/alternatif.[6]

Dalam hal ini, MCDM telah berperan sebagai metode untuk menangani dalam menata pengambilan solusi serta langkah-langkah dalam sebuah rencana pengambilan keputusan ketika masalah tersebut memiliki berbagai kriteria untuk mencapai hasil yang optimal berdasarkan preferensinya. Metode MCDM memiliki banyak algoritma dengan perbedaannya masing-masing, perbedaan tersebut umumnya terletak pada tingkat kesulitan algoritma yang digunakan, metode dalam memberikan bobot suatu kriteria, langkah kriteria dalam evaluasi preferensi, probabilitas data yang tidak pasti, dan tipe data agregasi. Setiap algoritma dalam MCDM memiliki kelebihan dan kekurangannya tersendiri, dapat disesuaikan dengan kebu-

tuhan mengenai perkiraan algoritma yang akan dipakai dan lebih cocok digunakan untuk pengambilan keputusan atau solusi dalam sebuah masalah. MCDM dapat diaplikasikan dalam berbagai macam bidang untuk mengambil keputusan atau solusi dalam sebuah masalah. Misalnya menggunakan MCDM untuk menyelesaikan masalah dalam bidang kesehatan, pendidikan, transportasi, dan lain sebagainya.[7]

2.2 Fuzzy

Teori Fuzzy pertama kali diusulkan oleh Zadeh setelah diterapkan untuk mencerminkan ambiguitas dalam menilai manusia hingga efektif untuk menyelesaikan ketidakpastian dari informasi yang tersedia dalam beberapa kriteria yang tidak pasti untuk mengambil keputusan. Metode MCDM Fuzzy digunakan untuk menilai antara alternatif dengan kriteria yang dipilih dalam pengambilan keputusan. Sehingga, akan disesuaikan kembali dari setiap aspek yang ada seperti alternatif atau kriteria, bobot kriteria, hingga evaluasi berupa nilai linguistik bilangan fuzzy. Keluaran dari pengambilan keputusan menggunakan metode MCDM Fuzzy yaitu masih berupa evaluasi alternatif bilangan fuzzy dan bilangan tersebut perlu pendekatan urutan ranking yang tepat dalam defuzzifikasi menjadi nilai nyata untuk memutuskan sebuah solusi dari sebuah masalah. Penerapan metode MCDM Fuzzy dalam pengambilan keputusan terkadang masih menjadi pertimbangan karena pendekatan ini cukup rumit untuk dilakukan komputasi dan diterapkan, sehingga dalam melakukan proses mengurutkan ranking bilangan fuzzy masih belum memuaskan dalam beberapa kasus. Hal ini dikarenakan value dari bilangan fuzzy tersebut yang masih dalam ambang keraguan atau belum dapat diputuskan mengenai nilai kepastiannya/kepuasannya. Nilai nilai alternatif dalam fuzzy dapat dikategorikan menjadi kriteria subjektif dan objektif. Dalam kriteria subjektif, biasanya berbentuk kualitatif sebuah definisi. Sedangkan kriteria objektif biasanya berupa sebuah istilah numerik yang dinormalisasikan menjadi skala yang setara dengan metode tersebut untuk dilakukan perhitungan. Fungsi dari keanggotaan evaluasi metode fuzzy dikembangkan melalui interval aritmatika dan nilai alpha dari bilangan fuzzy. Hasil dari defuzzifikasi tersebut akan digunakan sebagai nilai keputusan akhir untuk mengambil sebuah keputusan atau solusi dari sebuah masalah. Operasi dasar bilangan segitiga fuzzy:

$$\begin{aligned}
\widetilde{n}_1 \oplus \widetilde{n}_2 &= (n_{1l} + n_{2l}, n_{1m} + n_{2m}, n_{1u} + n_{2u}) \text{ for addition} \\
\widetilde{n}_1 \otimes \widetilde{n}_2 &= (n_{1l} \times n_{2l}, n_{1m} \times n_{2m}, n_{1u} \times n_{2u}) \text{ for multiplication} \\
\ominus \widetilde{n}_1 &= (-n_{1u}, -n_{1m}, -n_{1l}) \text{ for negation} \\
1/\widetilde{n}_1 &\cong (1/n_{1u}, 1/n_{1m}, 1/n_{1l}) \text{ for division} \\
\ln(\widetilde{n}_1) &\cong (\ln(n_{1l}), \ln(n_{1m}), \ln(n_{1u})) \text{ for natural logarithm} \\
\exp(\widetilde{n}_1) &\cong (\exp(n_{1l}), \exp(n_{1m}), \exp(n_{1u})) \text{ for exponential}
\end{aligned}$$

Gambar 2.1. Contoh snapshot 1

Variabel linguistik dalam metode MCDM Fuzzy merupakan variabel yang nilainya dinyatakan dalam bentuk bahasa/linguistik. Contohnya berupa sebuah kata kata seperti dalam kata "Penting" dapat dikategorikan lagi menjadi beberapa variabel linguistik dengan value tertentu yaitu SR merupakan kata "Penting" dengan kategori Sangat Rendah hingga ST yaitu kata "Penting" dengan kriteria Sangat Tinggi. Nilai linguistik ini dapat diwujudkan dalam sebuah bilangan fuzzy seperti kriteria Sangat Rendah misal (0, 0.1, 0.3) dan kriteria Sangat Tinggi yaitu (0.7, 0.9, 1). [8]

2.2.1 Fuzzy TOPSIS

Algoritma fuzzy banyak dipergunakan karena algoritma tersebut cukup untuk memodelkan situasi dalam kehidupan nyata dari data yang kurang jelas, pendekatan lebih realistis karena menggunakan value linguistik untuk menilai bobot seluruh kriteria dan peringkat. Dari peringkat tersebut merupakan berasal dari setiap alternatif yang berhubungan dengan kriteria tersebut. Fuzzy pada Topsis dipergunakan untuk menentukan solusi ideal positif fuzzy dan solusi ideal negatif fuzzy. Dalam penentuan solusi ini digunakan metode vertex yaitu dengan menghitung jarak dari setiap alternatif positif dan negatifnya. Kemudian, koefisien dari setiap alternatif yang ada akan ditentukan untuk menentukan peringkat dari seluruh alternatif yang ada. Hal ini ditunjukkan dengan nilai koefisien, jika nilai koefisien semakin dekat maka alternatif tersebut semakin dekat dengan value solusi ideal positif. Sebaliknya, jika nilai koefisien menunjukkan semakin jauh dari alternatif maka alternatif tersebut lebih jauh dari solusi ideal negatif secara bersamaan. Algoritma Fuzzy Topsis banyak digunakan oleh perindustrian. Pemanfaatan tersebut biasanya digunakan untuk menentukan peringkat pasokan energi terbarukan, mengungkapkan ketidakpastian/ketidakjelasan dalam pemilihan suatu lokasi, menentukan solusi

dari masalah industri yang terbaik secara berkelanjutan, hingga menentukan prioritas dalam sebuah skenario perusahaan.[9]

2.2.2 Fuzzy AHP

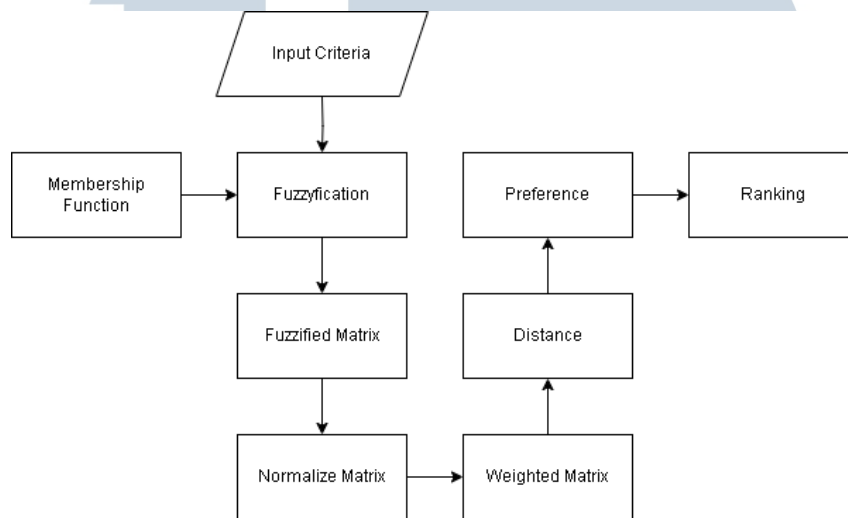
Pada algoritma Fuzzy Analytical Hierarchy Process yaitu algoritma dalam salah satu metode pengurutan dan gabungan dari metode AHP dengan konsep Fuzzy. Konsep Fuzzy pada algoritma AHP ini akan meminimalisir kesalahan yang disebabkan oleh kriteria kriteria yang didominasi oleh kriteria subjektif. Sehingga, pengukuran dalam Fuzzy AHP ini akan lebih berhubungan dalam menilai manusia secara subjektif menggunakan value bahasa/linguistik. Variabel bahasa atau linguistik ini dipergunakan sebagai proses translasi sebuah informasi dalam lingkup Fuzzy. Pada lingkup Fuzzy ini, dalam mengolah informasi menggunakan bilangan triangular fuzzy dengan simbol M . Sehingga, metode Fuzzy AHP ini berfokus pada rasio yang berpasangan dengan skala rasio berelasi bersama nilai skala Fuzzy tersebut. Fuzzy AHP ini digunakan ketika akan mengangkat permasalahan yang rasio perbandingannya merupakan sebuah penilaian yang tidak pasti/tepat. Hal ini karena pada nyatanya selalu terdapat lebih dari satu kemungkinan, contohnya dalam sebuah masalah dapat diambil keputusan secara tepat dan tidak tepat. Sama halnya dengan manusia dalam mengambil sebuah keputusan, ada manusia yang melakukan prediksi solusi dari masalah secara kuantitatif dan ada juga yang memprediksi secara kualitatif.[10]



BAB 3 SOLUTION

3.1 Flowchart MCDM Fuzzy TOPSIS

Gambar Flowchart proses MCDM Fuzzy TOPSIS



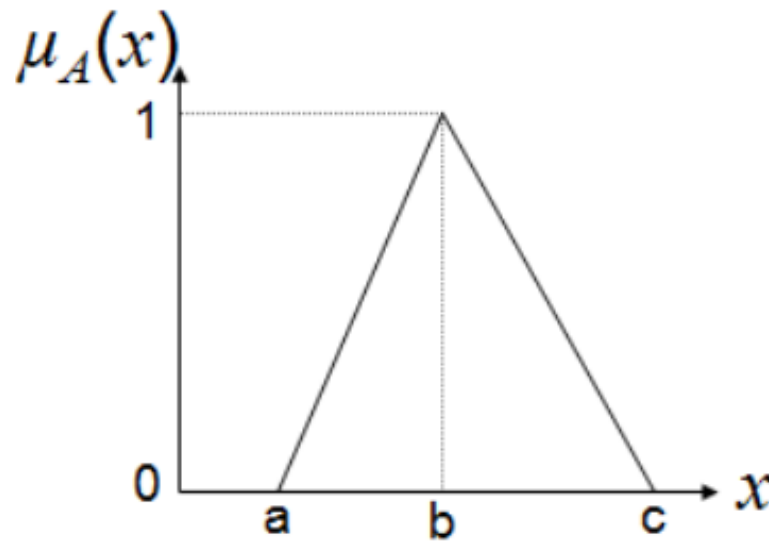
Gambar 3.1. Flowchart MCDM Fuzzy TOPSIS

3.2 Fuzzification

Pada tahap pertama, data yang diambil dari kaggle akan diubah terlebih dahulu bentuknya dari kategorikal menjadi numerikal. Ini bisa dilakukan dengan menggunakan Fuzzy. Aturan yang digunakan dalam menentukan Fuzzy adalah triangular membership function:

Gambar Fuzzy Triangular Membership Function[11]

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.2. Fuzzy Triangular Membership Function

Rumus yang digunakan dalam mengubah nilai dataset

$$\text{fuzzyfication_process}(x) = \begin{cases} \frac{1+2+3}{3} & \text{if } x = \text{'low'}$$

$$\frac{3+4+5}{3} & \text{if } x = \text{'med'}$$

$$\frac{5+6+7}{3} & \text{if } x = \text{'high'}$$

$$\frac{7+8+9}{3} & \text{if } x = \text{'very high/more'}$$

$$\text{None} & \text{if } x \text{ is unknown} \end{cases}$$

Dataset kemudian akan dicek dan diambil nilai kategorikalnya lalu diubah berdasarkan nilai numerikal yang ada berdasarkan dengan aturan fuzzy di atas. Sebagai contoh:

- **Low:** akan mendapatkan nilai $1+2+3/3 = 2.0$
- **Medium:** akan mendapatkan nilai $3+4+5/3 = 4.0$
- **High:** akan mendapatkan nilai $5+6+7/3 = 6.0$
- **Very High / More:** akan mendapatkan nilai $7+8+9/3 = 8.0$
- **None:** jika data tidak memiliki nilai kategorikal yang diketahui

Berikut merupakan hasil dataset yang telah di fuzzifikasi

	buying	maint	doors	persons	lug_boot	safety	rating
0	8.0	2.0	2.0	2.0	2.0	2.0	2.0
1	8.0	2.0	2.0	2.0	2.0	4.0	2.0
2	8.0	2.0	2.0	2.0	2.0	6.0	2.0
3	8.0	2.0	2.0	2.0	4.0	2.0	2.0
4	8.0	2.0	2.0	2.0	4.0	4.0	2.0

Gambar 3.3. Dataset fuzzifikasi

3.3 Normalisasi Matriks

Pada tahap ini, dataset yang telah difuzzy akan diubah menjadi matriks. Hal ini dilakukan agar perhitungan secara Topsis jauh lebih mudah dikalkulasi. Matrix di normalisasi menggunakan rumus berikut :

$$R_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^m a_{ij}^2}}$$

Disini R_{ij} menunjukkan elemen pada row i and column j dari matrix R . Rumus $\sum_{i=1}^m a_{ij}^2$ Mengindikasikan jumlah kuadrat dari elemen-elemen dalam kolom ke- j th dari matrix A .

Persamaan ini menghitung setiap elemen R_{ij} dari matriks ternormalisasi R dengan membagi elemen yang sesuai a_{ij} dari matriks A oleh akar kuadrat dari jumlah kuadrat elemen-elemen dalam kolom yang sama dari matriks A .

Matriks kemudian diubah menjadi bentuk weighted matriks, dikarenakan nilai weight tidak ada maka diasumsikan nilai weight secara default = 1. Hasil weighted matriks sama seperti matriks ternormalisasi sebelumnya[12].

Berikut merupakan hasil akhir dari weighted matrix

```
print(weighted_matrix)
[[0.04303315 0.01075829 0.01603751 ... 0.0186339 0.01363862 0.01958527]
 [0.04303315 0.01075829 0.01603751 ... 0.0186339 0.02727724 0.01958527]
 [0.04303315 0.01075829 0.01603751 ... 0.0186339 0.04091585 0.01958527]
 ...
 [0.01075829 0.04303315 0.04009377 ... 0.0372678 0.01363862 0.01958527]
 [0.01075829 0.04303315 0.04009377 ... 0.0372678 0.02727724 0.0587558 ]
 [0.01075829 0.04303315 0.04009377 ... 0.0372678 0.04091585 0.07834107]]
```

Gambar 3.4. Weighted Matrix

3.4 Jarak Euclidean

Setelah mendapatkan hasil dari weighted matrix, kemudian akan dihitung nilai ideal positif dan nilai ideal negatifnya.

Ideal Positive Solution (A^+):

$$A_j^+ = \max(R_{ij}) \quad \text{for } i = 1, 2, \dots, m$$

Ideal Negative Solution (A^-):

$$A_j^- = \min(R_{ij}) \quad \text{for } i = 1, 2, \dots, m$$

Nilai ideal positif mengambil nilai max, sedangkan nilai ideal negatif mengambil nilai min.

Jarak Relatif terhadap Solusi Ideal Positif (A^+):

$$\text{jarak_positif}(\text{alternatif}_i) = \sqrt{\sum_{j=1}^n (W_{ij} - A_j^+)^2} \quad \text{untuk } i = 1, 2, \dots, m$$

W_{ij} adalah elemen matriks terbobot untuk alternatif i dan kriteria j . A_j^+ adalah nilai solusi ideal positif untuk kriteria j . n adalah jumlah kriteria.

Jarak Relatif terhadap Solusi Ideal Negatif (A^-):

$$\text{jarak_negatif}(\text{alternatif}_i) = \sqrt{\sum_{j=1}^n (W_{ij} - A_j^-)^2} \quad \text{untuk } i = 1, 2, \dots, m$$

A_j^- adalah nilai solusi ideal negatif untuk kriteria j .

Kedua persamaan tersebut menghitung jarak Euclidean dari setiap alternatif terhadap solusi ideal positif (A^+) dan solusi ideal negatif (A^-) dalam metode TOPSIS. Ini membantu dalam menentukan seberapa dekat atau jauh setiap alternatif dari kedua solusi ideal tersebut dalam proses pengambilan keputusan[12].

3.5 Preference dan Ranking

Skor Preferensi untuk setiap alternatif dalam TOPSIS dihitung dengan:

$$\text{preference_score} = \frac{\text{negative_distance}}{\text{positive_distance} + \text{negative_distance}}$$

negative_distance menunjukkan jarak dari setiap alternative dari nilai ideal negatif. (A^-). positive_distance menunjukkan jarak dari setiap alternative dari nilai ideal positif. (A^+) [12].

Rumus ini digunakan untuk mencari jarak dari setiap alternatif terhadap hasil idealnya dengan membandingkan jarak dari ideal positif dan ideal negatif. Nilai dari preferensi berkisar diantara 0 - 1, semakin besar nilainya maka semakin dekat akan solusi ideal.

Nilai dari preference kemudian digunakan untuk menampilkan ranking pada dataset yang telah diolah. Ranking akan dinilai dari seberapa besar nilai preference.

Hasil Ranking dan Skor Preferensi:

	Ranking	Preference Score	Negative Distance	Positive Distance
137	1	0.461549	0.054894	0.064041
211	2	0.461549	0.054894	0.064041
398	3	0.355641	0.039749	0.072017
246	4	0.436971	0.053774	0.069287
353	5	0.410161	0.048155	0.069251
...
981	1148	0.280686	0.030103	0.077144
1111	1149	0.458813	0.049190	0.058021
751	1150	0.450825	0.044234	0.053884
1113	1151	0.387381	0.046875	0.074129
1141	1152	0.459207	0.048744	0.057404

Gambar 3.5. Hasil Ranking

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

BAB 4

DEMO - SIMULATION

4.1 Importing Dataset

Dataset The Cars yang diambil dari website Kaggle, dimana dataset ini terdiri dari 7 column dan 1728 row. Dataset yang digunakan merupakan dataset kategorical. Isi dari dataset dapat dijelaskan sebagai berikut Gambar 4.1:

- **Buying:** tingkat ketertarikan akan membeli mobilnya.
- **Maintenance:** tingkat keseringan dalam maintenance mobil.
- **Doors:** jumlah pintu pada mobil.
- **Persons:** jumlah penumpang yang dapat naik kedalam mobil.
- **Luggage Boot:** besar dari bagasi mobil.
- **Safety:** tingkat keamanan dari mobil.
- **Rating:** tingkat performa dari mobil secara keseluruhan.

```
[61] data = pd.read_csv("sample_data/car1.csv")
```

```
[62] data.head()
```

	buying	maint	doors	persons	lug_boot	safety	rating
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

Gambar 4.1. Import dataset yang digunakan

4.2 Defuzzifying Dataset

Function yang dibuat berdasarkan dari membership function triangular fuzzy yang sudah disesuaikan dengan tabel sehingga menghasilkan rule sebagai berikut:


```
[66] # Defuzzification function for 'maint' column
def defuzzify_maint(value):
    if value == 'vhigh':
        return (1+2+3)/3
    elif value == 'high':
        return (3+4+5)/3
    elif value == 'med':
        return (5+6+7)/3
    elif value == 'low':
        return (7+8+9)/3
    else:
        return None # Return None for unknown cases
```

Gambar 4.2. Defuzzifying maintenance column

```
[67] # Defuzzification function for 'doors' column
def defuzzify_doors(value):
    if value == '2':
        return (1+2+3)/3
    elif value == '3':
        return (2+3+4)/3
    elif value == '4':
        return (3+4+5)/3
    elif value == '5more':
        return (4+5+6)/3
    else:
        return None # Return None for unknown cases
```

Gambar 4.3. Defuzzifying doors column

```
[68] # Defuzzification function for 'persons' column
def defuzzify_persons(value):
    if value == '2':
        return (1+2+3)/3
    elif value == '4':
        return (3+4+5)/3
    elif value == 'more':
        return (5+6+7)/3
    else:
        return None # Return None for unknown cases
```

Gambar 4.4. Defuzzifying persons column

```
[69] # Defuzzification function for 'lug_boot' column
def defuzzify_lug_boot(value):
    if value == 'small':
        return (1+2+3)/3
    elif value == 'med':
        return (3+4+5)/3
    elif value == 'high':
        return (5+6+7)/3
    else:
        return None # Return None for unknown cases
```

Gambar 4.5. Defuzzifying luggage boot column

```
[70] # Defuzzification function for 'safety' column
def defuzzify_safety(value):
    if value == 'low':
        return (1+2+3)/3
    elif value == 'med':
        return (3+4+5)/3
    elif value == 'high':
        return (5+6+7)/3
    else:
        return None # Return None for unknown cases
```

Gambar 4.6. Defuzzifying safety column

```
[70] # Defuzzification function for 'safety' column
def defuzzify_safety(value):
    if value == 'low':
        return (1+2+3)/3
    elif value == 'med':
        return (3+4+5)/3
    elif value == 'high':
        return (5+6+7)/3
    else:
        return None # Return None for unknown cases
```

Gambar 4.7. Defuzzifying safety column

```
[71] # Defuzzification function for 'rating' column
def defuzzify_rating(value):
    if value == 'unacc':
        return (1+2+3)/3
    elif value == 'acc':
        return (3+4+5)/3
    elif value == 'good':
        return (5+6+7)/3
    elif value == 'vgood':
        return (7+8+9)/3
    else:
        return None # Return None for unknown cases
```

Gambar 4.8. Defuzzifying rating column

```
[72] # Defuzzification function for 'buying' column returning a list of numbers
def defuzzify_buying(value):
    if value == 'low':
        return (1+2+3)/3
    elif value == 'med':
        return (3+4+5)/3
    elif value == 'high':
        return (5+6+7)/3
    elif value == 'vhigh':
        return (7+8+9)/3
    else:
        return None # Return None for unknown cases
```

Gambar 4.9. Defuzzifying buying column

Pemanggilan function fuzzy untuk mengubah jenis tipe data pada tabel sebelumnya menjadi fuzzy number. Gambar 4.10

U M M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

```
[73] data['buying'] = data['buying'].apply(defuzzify_buying)
data['maint'] = data['maint'].apply(defuzzify_maint)
data['doors'] = data['doors'].apply(defuzzify_doors)
data['persons'] = data['persons'].apply(defuzzify_persons)
data['lug_boot'] = data['lug_boot'].apply(defuzzify_lug_boot)
data['safety'] = data['safety'].apply(defuzzify_safety)
data['rating'] = data['rating'].apply(defuzzify_rating)

[74] data.head()
```

	buying	maint	doors	persons	lug_boot	safety	rating
0	8.0	2.0	2.0	2.0	2.0	2.0	2.0
1	8.0	2.0	2.0	2.0	2.0	4.0	2.0
2	8.0	2.0	2.0	2.0	2.0	6.0	2.0
3	8.0	2.0	2.0	2.0	4.0	2.0	2.0
4	8.0	2.0	2.0	2.0	4.0	4.0	2.0

Gambar 4.10. Implementing defuzzy to dataframe

4.3 Converting to Matrix

Function yang digunakan untuk mengubah bentuk tabel menjadi matrix sehingga dapat diolah lebih mudah dengan TOPSIS. Gambar 4.11

```
[ ] df = pd.DataFrame(data)

# Mengubah DataFrame menjadi matriks NumPy
decision_matrix = df.to_numpy()

print("Matriks hasil konversi:")
print(decision_matrix)
```

```
Matriks hasil konversi:
[[8. 2. 2. ... 2. 2. 2.]
 [8. 2. 2. ... 2. 4. 2.]
 [8. 2. 2. ... 2. 6. 2.]
 ...
 [2. 8. 5. ... 4. 2. 2.]
 [2. 8. 5. ... 4. 4. 6.]
 [2. 8. 5. ... 4. 6. 8.]]
```

Gambar 4.11. Converting dataframe to matrix numpy

4.4 Matrix Normalization

Fungsi `normalize_decision_matrix` dibuat untuk melakukan normalisasi pada matriks keputusan. Normalisasi dilakukan untuk penyesuaian nilai-nilai dalam matriks ke skala standar. Gambar 4.12

Fungsi ini menghasilkan matriks baru (`normalized_matrix`) yang memiliki ukuran yang sama dengan matriks masukan tetapi dengan nilai-nilai yang telah dinormalisasi.

```
# normalisasi matriks keputusan
def normalize_decision_matrix(matrix):
    normalized_matrix = np.zeros(matrix.shape)
    for j in range(matrix.shape[1]):
        col_sum_of_squares = np.sqrt(np.sum(matrix[:, j] ** 2))
        normalized_matrix[:, j] = matrix[:, j] / col_sum_of_squares
    return normalized_matrix

normalized_decision_matrix = normalize_decision_matrix(decision_matrix)
print("Matriks keputusan yang telah dinormalisasi:")
print(normalized_decision_matrix)
```

Matriks keputusan yang telah dinormalisasi:

```
[[0.04303315 0.01075829 0.01603751 ... 0.0186339 0.01363862 0.01958527]
 [0.04303315 0.01075829 0.01603751 ... 0.0186339 0.02727724 0.01958527]
 [0.04303315 0.01075829 0.01603751 ... 0.0186339 0.04091585 0.01958527]
 ...
 [0.01075829 0.04303315 0.04009377 ... 0.0372678 0.01363862 0.01958527]
 [0.01075829 0.04303315 0.04009377 ... 0.0372678 0.02727724 0.0587558 ]
 [0.01075829 0.04303315 0.04009377 ... 0.0372678 0.04091585 0.07834107]]
```

Gambar 4.12. Matrix Normalizations

4.5 Weighted Matrix

Function `Weighted matrix` berfungsi untuk memberikan nilai `weight` kepada `matrix` yang sudah dinormalisasikan dengan cara mengkalikan `matrix` normalisasi dengan nilai `weight`, namun dikarenakan dataset tidak memiliki `weight` maka diasumsikan bahwa `weight` defaultnya adalah 1. Gambar 4.13

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

weighted_value = [1, 1, 1, 1, 1, 1, 1]

# karena nilai Weight = 1 maka tidak ada perubahan nilai pada matriks
weighted_matrix = normalized_decision_matrix * weighted_value

print(weighted_matrix)

[[0.04303315 0.01075829 0.01603751 ... 0.0186339 0.01363862 0.01958527]
 [0.04303315 0.01075829 0.01603751 ... 0.0186339 0.02727724 0.01958527]
 [0.04303315 0.01075829 0.01603751 ... 0.0186339 0.04091585 0.01958527]
 ...
 [0.01075829 0.04303315 0.04009377 ... 0.0372678 0.01363862 0.01958527]
 [0.01075829 0.04303315 0.04009377 ... 0.0372678 0.02727724 0.0587558 ]
 [0.01075829 0.04303315 0.04009377 ... 0.0372678 0.04091585 0.07834107]]

```

Gambar 4.13. Weighted Matrix

4.6 Ideal Positif and Ideal Negatif

Function Ideal Positif dan Ideal Negatif berfungsi sebagai referensi untuk menentukan jarak dari alternative lainnya. Solusi positif ideal mewakili nilai atau atribut terbaik yang mungkin dimiliki oleh sebuah alternatif untuk setiap kriteria. Solusi ini terdiri dari nilai atau karakteristik yang paling diinginkan untuk setiap kriteria. Solusi negatif ideal, sebaliknya, mewakili nilai atau atribut terburuk yang mungkin dimiliki oleh sebuah alternatif untuk setiap kriteria. Solusi ini terdiri dari nilai atau karakteristik yang paling tidak diinginkan untuk setiap kriteria. Gambar 4.14

```

# nilai ideal positif
ideal_positive

array([0.04303315, 0.04303315, 0.04009377, 0.04091585, 0.0372678 ,
       0.04091585, 0.07834107])

# nilai ideal negatif
ideal_negative

array([0.01075829, 0.01075829, 0.01603751, 0.01363862, 0.0186339 ,
       0.01363862, 0.01958527])

```

Gambar 4.14. Ideal Positive and Negative

4.7 Euclidian Distance

Function Jarak Ideal Positif dan Jarak Ideal Negatif menghasilkan nilai jarak (distance) relatif dari setiap alternatif terhadap solusi ideal positif dan solusi ideal negatif dalam proses TOPSIS. Semakin kecil jaraknya terhadap solusi ideal positif dan semakin besar jaraknya terhadap solusi ideal negatif, maka alternatif

tersebut akan lebih dianggap mendekati solusi terbaik. Gambar 4.15

```
# nilai ideal positif
ideal_positive

array([0.04303315, 0.04303315, 0.04009377, 0.04091585, 0.0372678 ,
       0.04091585, 0.07834107])

# nilai ideal negatif
ideal_negative

array([0.01075829, 0.01075829, 0.01603751, 0.01363862, 0.0186339 ,
       0.01363862, 0.01958527])
```

Gambar 4.15. Euclidian Distance Positive and Negative

4.8 Preference

Function dari Preference adalah setiap nilai skor preferensi untuk setiap alternatif. Nilai-nilai ini mencerminkan seberapa dekat (preference) setiap alternatif terhadap solusi ideal positif dan solusi ideal negatif. Skor preferensi yang lebih tinggi menunjukkan kecenderungan alternatif yang lebih baik (lebih mendekati solusi ideal positif) dalam konteks pengambilan keputusan menggunakan metode TOPSIS. Gambar 4.16

```
# skor preferensi
preference_score = negative_distance / (positive_distance + negative_distance)
print(preference_score)

[0.27970525 0.3054121 0.34990784 ... 0.41843796 0.62384835 0.72029475]
```

Gambar 4.16. Preference

4.9 Ranking

Hasil dari preference kemudian diurutkan dan dirank menggunakan function sort sehingga dapat dilihat nilai Preference Score, Negative Distance, Positive Distance. Hasil yang paling bagus dengan rank 1 adalah index 137 dengan Preference Score 0.461, Negative Distance 0.054, dan Positive Distance 0.064. Gambar 4.17

Hasil Ranking dan Skor Preferensi:

	Ranking	Preference Score	Negative Distance	Positive Distance
137	1	0.461549	0.054894	0.064041
211	2	0.461549	0.054894	0.064041
398	3	0.355641	0.039749	0.072017
246	4	0.436971	0.053774	0.069287
353	5	0.410161	0.048155	0.069251
...
981	1148	0.280686	0.030103	0.077144
1111	1149	0.458813	0.049190	0.058021
751	1150	0.450825	0.044234	0.053884
1113	1151	0.387381	0.046875	0.074129
1141	1152	0.459207	0.048744	0.057404

[1152 rows x 4 columns]

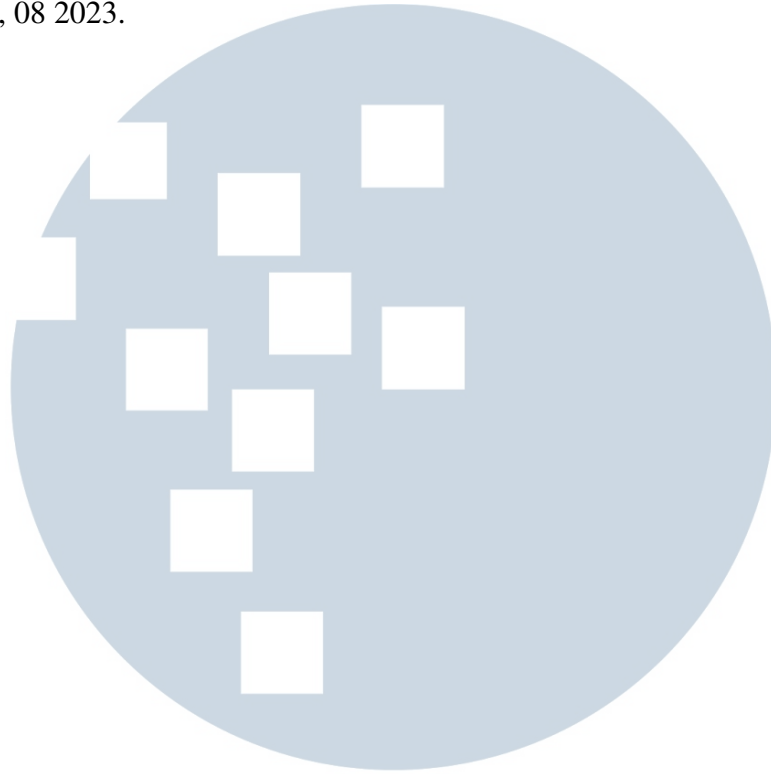
Gambar 4.17. Final Ranking



DAFTAR PUSTAKA

- [1] A. Susiana, A. Kamsyakawuni, and A. Riski, "Sistem pendukung keputusan pemasaran mobil di toyota auto2000 jember dengan metode fuzzy topsis," *Jurnal Saintika Unpam : Jurnal Sains dan Matematika Unpam*, vol. 3, p. 42, 7 2020.
- [2] A. Azshari, "Penjualan mobil di indonesia hingga oktober 2022 tembus 822 ribuan," 11 2022. [Online]. Available: <https://www.liputan6.com/otomotif/read/5127202/penjualan-mobil-di-indonesia-hingga-oktober-2022-tembus-822-ribuan>
- [3] M. B. N. Kore, K. Ravi, and A. S. B. Patil, ""a simplified description of fuzzy topsis method for multi criteria decision making"," *International Research Journal of Engineering and Technology*, 2017. [Online]. Available: www.irjet.net
- [4] N. Kadek, S. Stmik, S. Bali, and J. R. Puputan, "Penerapan fuzzy topsis untuk seleksi penerima bantuan kemiskinan," 2015.
- [5] A. Tulkhah and A. Saifudin, "Fuzzy topsis untuk meningkatkan akurasi dan objektivitas bobot pada seleksi vendor pt. telkomsel ttc bsd," *Jurnal Informatika Universitas Pamulang*, vol. 4, p. 28, 3 2019.
- [6] A. Mardani, A. Jusoh, K. M. Nor, Z. Khalifah, N. Zakwan, and A. Valipour, "Multiple criteria decision-making techniques and their applications," vol. 28, pp. 516–571, 7 2015.
- [7] H. Taherdoost and M. Madanchian, "Multi-criteria decision making (mcdm) methods and concepts," *Encyclopedia*, vol. 3, pp. 77–87, 2023. [Online]. Available: <https://www.mdpi.com/2673-8392/3/1/6>
- [8] T.-C. Chu and Y. Lin, "An extension to fuzzy mcdm," *Computers Mathematics with Applications*, vol. 57, pp. 445–454, 2009. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0898122108006408>
- [9] İhsan Kaya, M. Çolak, and F. Terzi, "A comprehensive review of fuzzy multi criteria decision making methodologies for energy policy making," *Energy Strategy Reviews*, vol. 24, pp. 207–228, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2211467X19300252>
- [10] A. Santoso, R. Rahmawati, and Sudarno, "Aplikasi fuzzy analytical hierarchy process untuk menentukan prioritas pelanggan berkunjung ke galeri (studi kasus di secondhand semarang)," vol. 5, pp. 239–248, 2016.
- [11] M. Azam, M. Hasan, S. Hassan, and S. Jadid Abdulkadir, "Fuzzy type-1 triangular membership function approximation using fuzzy c-means," 11 2020.

- [12] M. Madanchian and H. Taherdoost, “A comprehensive guide to the topsis method for multi-criteria decision making,” *Sustainable Social Development*, vol. 1, 08 2023.



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA