

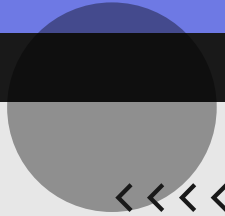
<<<<

ARTIFICIAL INTELLIGENCE (AI)

PRAKTIKUM 7

Dosen Pengampu : Renovita Edelani S.ST., M.Tr.Kom

oleh : Dukhaan Kamimpangan
2 D4 IT A
3122600003



01.

PRAKTIKUM



PROGRAM FUZZY EVALUASI PENILAIAN

```
import numpy as np
from typing import Dict
import math

class Fuzzy:
    aturan: Dict[str, Dict[str, float]]
    nilai: Dict[str, float]
    aturan_agregasi: Dict[str, list[Dict[str, str]]]

    def __init__(self, aturan: Dict[str, Dict[str, float]], nilai: Dict[str, float], aturan_agregasi: Dict[str, list[Dict[str, str]]] = {}):
        self.aturan = aturan
        self.nilai = nilai
        self.aturan_agregasi = aturan_agregasi

    def fuzzifikasi(self, aturan: Dict[str, float], nilai: float) -> Dict[str, float]:
        daftar_aturan = list(aturan.items())
        daftar_aturan.sort(key=lambda x: x[1])
```

PROGRAM FUZZY EVALUASI PENILAIAN

```
hasil: Dict[str, float] = {}
```

```
for i in range(len(daftar_aturan)):
```

```
    if i == 0 and len(daftar_aturan) == 1:
```

```
        hasil[daftar_aturan[i][0]] = 1
```

```
    elif i == 0:
```

```
        if (nilai <= daftar_aturan[i][1]):
```

```
            hasil[daftar_aturan[i][0]] = 1
```

```
        elif (nilai > daftar_aturan[i][1] and nilai < daftar_aturan[i+1][1]):
```

```
            hasil[daftar_aturan[i][0]] = round((float)(daftar_aturan[i+1][1] - nilai) / (daftar_aturan[i+1][1] - daftar_aturan[i][1]), 2)
```

```
    elif i > 0 and i < len(daftar_aturan) - 1:
```

```
        if (nilai < daftar_aturan[i][1] and nilai > daftar_aturan[i-1][1]):
```

```
            hasil[daftar_aturan[i][0]] = round((float)(nilai - daftar_aturan[i-1][1]) / (daftar_aturan[i][1] - daftar_aturan[i-1][1]), 2)
```

```
        elif (nilai == daftar_aturan[i][1]):
```

```
            hasil[daftar_aturan[i][0]] = 1
```

```
        elif (nilai > daftar_aturan[i][1] and nilai < daftar_aturan[i+1][1]):
```

```
            hasil[daftar_aturan[i][0]] = round((float)(daftar_aturan[i+1][1] - nilai) / (daftar_aturan[i+1][1] - daftar_aturan[i][1]), 2)
```

PROGRAM FUZZY EVALUASI PENILAIAN

```
elif i == len(daftar_aturan) - 1:
    if (nilai < daftar_aturan[i][1] and nilai > daftar_aturan[i-1][1]):
        hasil[daftar_aturan[i][0]] = round((float)(nilai - daftar_aturan[i-1][1]) / (daftar_aturan[i][1] - daftar_aturan[i-1][1]), 2)
    elif (nilai >= daftar_aturan[i][1]):
        hasil[daftar_aturan[i][0]] = 1

return hasil

def fuzzifikasi_semua(self) -> Dict[str, Dict[str, float]]:
    hasil: Dict[str, Dict[str, float]] = {}

    for key, value in self.nilai.items():
        hasil[key] = self.fuzzifikasi(self.aturan[key], value)

return hasil
```

PROGRAM FUZZY EVALUASI PENILAIAN

```
    hasil_fuzzy = self.fuzzifikasi_semua()
    hasil: Dict[str, float] = {}
    for kunci_agregat, value in self.aturan_agregasi.items():
        for aturan in value:
            temp: list[float] = []
            for key, value in aturan.items():
                if value in hasil_fuzzy[key]:
                    temp.append(hasil_fuzzy[key][value])
            if len(temp) > 1:
                hasil[kunci_agregat] = min(temp)

    return hasil

def defuzzifikasi_max(self):
    hasil_agregat = self.agregasi()
    hasil = max(hasil_agregat, key=hasil_agregat.get)
    return hasil
```

PROGRAM FUZZY EVALUASI PENILAIAN

```
def defuzzifikasi_centroid(self):
    if 'aggregation' not in self.aturan:
        raise ValueError('Aturan Agregasi tidak ditemukan dalam aturan')

    hasil_agregat = self.agregasi()
    nilai = 0
    for key, value in hasil_agregat.items():
        nilai += self.aturan['aggregation'][key] * value
    nilai /= sum(hasil_agregat.values())

    hasil_fuzzy = self.fuzzifikasi(self.aturan['aggregation'], nilai)
    hasil = max(hasil_fuzzy, key=hasil_fuzzy.get)

    return hasil
```

PROGRAM FUZZY EVALUASI PENILAIAN

```
aturan = {  
    'aggregation': {  
        'sangat baik': 100,  
        'baik sekali': 90,  
        'baik': 80,  
        'cukup baik': 70,  
        'kurang baik': 60,  
    },  
    'gre': {  
        'rendah': 800,  
        'sedang': 1200,  
        'tinggi': 1800,  
    },  
    'gpa': {  
        'rendah': 2.2,  
        'sedang': 3.0,  
        'tinggi': 3.8,  
    },  
}
```



PROGRAM FUZZY EVALUASI PENILAIAN

```
nilai = {'gre': 900, 'gpa': 3.2}
```

```
aturan_agregasi = {  
    'sangat baik': [  
        {'gre': 'tinggi', 'gpa': 'tinggi'}  
    ],  
    'baik sekali': [  
        {'gre': 'sedang', 'gpa': 'tinggi'}  
    ],  
    'baik': [  
        {'gre': 'tinggi', 'gpa': 'sedang'},  
        {'gre': 'sedang', 'gpa': 'sedang'}  
    ],  
    'cukup baik': [  
        {'gre': 'tinggi', 'gpa': 'rendah'},  
        {'gre': 'rendah', 'gpa': 'tinggi'}  
    ],  
    'kurang baik': [  
        {'gre': 'rendah', 'gpa': 'sedang'},  
        {'gre': 'sedang', 'gpa': 'rendah'},  
        {'gre': 'rendah', 'gpa': 'rendah'}  
    ],  
}
```

```
uji_fuzzy = Fuzzy(aturan, nilai, aturan_agregasi)  
print(uji_fuzzy.defuzzifikasi_centroid())
```

HASIL RUNNING PROGRAM



```
uji_fuzzy = Fuzzy(aturan, nilai, aturan_agregasi)
print(uji_fuzzy.defuzzifikasi_centroid())
```

```
cukup baik
```

Code lengkap : <https://pastebin.com/JT77GGu8>

/ [AI]

02.

ANALISA CODE

ANALISA CODE

Kelas Fuzzy: Program dimulai dengan mendefinisikan kelas Fuzzy. Kelas ini memiliki beberapa atribut dan metode yang digunakan untuk menghitung nilai defuzzifikasi dari input yang diberikan.

Inisialisasi: Saat objek Fuzzy dibuat, konstruktor `__init__` akan dipanggil untuk menginisialisasi atribut-atribut kelas, seperti aturan fuzzifikasi, nilai input, dan aturan agregasi.

Fuzzifikasi: Metode fuzzifikasi digunakan untuk mengonversi nilai input menjadi himpunan fuzzy berdasarkan aturan fuzzifikasi yang diberikan.

Fuzzifikasi Semua: Metode `fuzzifikasi_semua` digunakan untuk melakukan fuzzifikasi pada semua nilai input yang diberikan.

Agregasi: Metode agregasi digunakan untuk mengagregasi semua himpunan fuzzy yang diperoleh dari fuzzifikasi nilai input. Ini dilakukan dengan menerapkan aturan-aturan agregasi yang telah ditentukan.

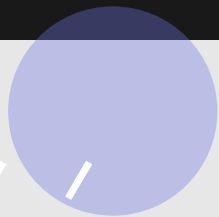
Defuzzifikasi Maksimum: Metode `defuzzifikasi_max` digunakan untuk melakukan defuzzifikasi menggunakan metode maksimum. Ini memilih nilai output yang memiliki nilai agregat maksimum dari semua himpunan fuzzy yang dihasilkan.

Defuzzifikasi Centroid: Metode `defuzzifikasi_centroid` digunakan untuk melakukan defuzzifikasi menggunakan metode centroid. Ini menghitung nilai tengah (centroid) dari area di bawah kurva himpunan fuzzy yang dihasilkan, kemudian memilih nilai output yang sesuai dengan centroid.

Analisis Data: Program kemudian mendefinisikan aturan, nilai input, dan aturan agregasi untuk diuji dengan menggunakan objek Fuzzy. Setelah objek dibuat, metode `defuzzifikasi_centroid` dipanggil untuk menghitung nilai defuzzifikasi menggunakan metode centroid.



/[AI]/[AI]/



**TERIMA
KASIH**

ARTI
CIAL
INTE
IGEN
[AI]