

**LAPORAN TUGAS BESAR PEMROGRAMAN
SISTEM CERDAS 2**



Disusun oleh :

Gregorio Bonggal Noveno Alvito - 1303202099

Daffa Satriarga - 1303202040

Muhammad Haikal - 1303202131

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
BANDUNG
2022**

ABSTRAKSI

Generasi zaman sekarang adalah generasi yang erat dengan keberadaan teknologi. hal ini menyebabkan kecenderungan mereka dalam memilih pekerjaan sesuai minat. namun hal ini tidak sejalan dengan bakat yang dimiliki. sehingga menghasilkan talenta yang kurang bisa dimanfaatkan secara efisien. oleh karena itu dibutuhkan rekomendasi pekerjaan yang sesuai bakat dan minat yaitu logika *fuzzy*. *fuzzy* merupakan kerangka matematis yang digunakan untuk mempresentasikan ketidakpastian. Strategi pemecahan masalah yang digunakan adalah metode logika *fuzzy* dan diimplementasikan dalam bahasa *Python*. Terdapat file “masukan.xlsx” yang berisi persentase minat dan bakat pada setiap ID. Seluruh data dibuat secara otomatis sehingga menghasilkan 100 data dengan nilai yang acak. File “masukan.xlsx” diakses untuk dilakukan perhitungan *fuzzy* dengan mempertimbangkan nilai persentase minat dan bakat. hasilnya, program menghasilkan 10 data terbaik dengan nilai kecocokan 60. dengan adanya fuzzy sebagai sistem rekomendasi pekerjaan berdasarkan minat dan bakat, maka akan menghasilkan talenta yang memiliki performa lebih efisien.

Kata kunci : Bakat, Fuzzy, Minat, Pekerjaan

BAB I

PENDAHULUAN

1. Latar Belakang

Generasi zaman sekarang adalah generasi yang erat dengan keberadaan teknologi. hal ini menyebabkan kecenderungan mereka dalam memilih pekerjaan sesuai minat. namun hal ini tidak sejalan dengan bakat yang dimiliki. sehingga menghasilkan talenta yang kurang bisa dimanfaatkan secara efisien. oleh karena itu dibutuhkan rekomendasi pekerjaan yang sesuai bakat dan minat yaitu logika *fuzzy*.

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Teori himpunan fuzzy merupakan kerangka matematis yang digunakan untuk mempresentasikan ketidakpastian (elisawati, 2017).

2. Strategi Pemecahan Masalah

Strategi pemecahan masalah yang digunakan adalah metode logika *fuzzy* dan diimplementasikan dalam bahasa *Python*.

3. Objek Observasi

Berikut adalah hal-hal yang diobservasi dalam pemecahan masalah.

- a. Aturan Inferensi
- b. Metode Defuzzifikasi
- c. Bentuk dan Batas Fungsi Keanggotaan Output (sesuai metode defuzzifikasi)

4. Software yang Digunakan

Berikut adalah *software* yang digunakan dalam implementasi logika *fuzzy*

- a. *Python 3.10.4* : Bahasa pemrograman
- b. *Visual Studio Code* : Editor Teks
- c. *Microsoft Excel 365* : *Sheet format viewer*

5. Library yang Digunakan

Berikut adalah *library* yang digunakan dalam implementasi logika *fuzzy*.

- a. *pandas*

6. Hasil Pengerjaan

Berikut merupakan link source code program :

<https://drive.google.com/drive/folders/1IEN2c3yJ7KCEH7LmAj4Est45CQ0RhzpI?usp=sharing>

Berikut merupakan link rekaman presentasi dari setiap anggota kelompok :

Masukkan link disini

BAB II

PEMBAHASAN

1. Program Utama

Program akan menjalankan fungsi `luaran()` dalam program utama.

```
#Program Utama  
  
if __name__ == "__main__":  
  
    luaran()
```

2. Fuzzy Logic

Fuzzy Logic dibuat dengan mengakses file “input.xlsx”. dan kemudian dihasilkan file “luaran.xlsx”.

2.1. Masukan

Terdapat file “masukan.xlsx” yang berisi presentase minat dan bakat pada setiap ID. Seluruh data dibuat secara otomatis sehingga menghasilkan 100 data dengan nilai yang acak.

	A	B	C	D	E
1	No	ID	Minat	Bakat	
2	1	A	81	32	
3	2	B	55	49	
4	3	C	46	1	
5	4	D	39	76	
6	5	E	68	72	
7	6	F	93	53	
8	7	G	92	17	
9	8	H	24	88	
10	9	I	78	51	
11	10	J	85	7	
12	11	K	45	91	
13	12	L	63	43	
14	13	M	74	82	
15	14	N	6	33	
16	15	O	23	65	
17	16	P	34	10	
18	17	Q	51	16	
19	18	R	28	34	
20	19	S	79	16	
21	20	T	13	77	
22	21	U	42	30	
23	22	V	7	35	
24	23	W	77	75	
25	24	X	45	27	
26	25	Y	46	35	

2.2. Luaran

Program menggunakan library *pandas* untuk mempermudah dalam mengolah dan menganalisa data terstruktur.

```
import pandas as pd
```

Pada fungsi `luaran()`, program melakukan akses file “masukan.xlsx” yang kemudian disimpan dalam variabel “data”. kemudian dilakukan definisi variabel untuk id, minat dan bakat.

```
def luaran():
```

```

#akses file masukan.xlsx

data = pd.read_excel('input.xlsx')

#definisi variabel

id = data["ID"]

minat = data["Minat"]

bakat = data["Bakat"]


hasil = []

```

Selanjutnya, dilakukan pengisian cell di setiap baris untuk melakukan perhitungan logika fuzzy. nilai minat didasarkan pada indikator rendah, kurang, normal, tinggi dan sangat tinggi. sedangkan bakat didasarkan pada indikator pemula, standar dan ahli. setelah itu dilakukan inferensi dengan mencocokkan nilai minat dan bakat.

```

for x in range(100):

    arr_minat = [0,0,0,0,0]

    arr_bakat = [0,0,0]

    rendah = kurang = normal = tinggi = sangatTinggi = 0

    pemula = standar = ahli = 0

    #fuzzification //haikal

    if minat[x] <= 10:

        rendah = 1

        arr_minat[0] = rendah

    elif minat[x] > 10 and minat[x] < 30:

        rendah = (30 - minat[x])/20

```

```
        kurang = (minat[x] - 10)/20

        arr_minat[0] = rendah

        arr_minat[1] = kurang

    elif minat[x] >= 30 and minat[x] <= 40:

        kurang = 1

        arr_minat[1] = kurang

    elif minat[x] > 40 and minat[x] < 60:

        kurang = (minat[x] - 40)/20

        normal = (60 - minat[x])/20

        arr_minat[1] = kurang

        arr_minat[2] = normal

    elif minat[x] >= 60 and minat[x] <=70:

        normal = 1

        arr_minat[2] = normal

    elif minat[x] > 70 and minat[x] < 80:

        normal = (80 - minat[x])/10

        tinggi = (minat[x] - 70)/10

        arr_minat[2] = normal

        arr_minat[3] = tinggi

    elif minat[x] >= 80 and minat[x] <= 85:

        tinggi = 1

        arr_minat[3] = tinggi

    elif minat[x] > 85 and minat[x] < 90:

        tinggi = (90 - minat[x])/5

        sangatTinggi = (minat[x] - 85)/5

        arr_minat[3] = tinggi

        arr_minat[4] = sangatTinggi
```



```
elif minat[x] >= 90:

    sangatTinggi = 1

    arr_minat[4] = sangatTinggi


if bakat[x] <= 10:

    pemula = 1

    arr_bakat[0] = pemula

elif bakat[x] > 10 and bakat[x] < 20:

    pemula = (20 - bakat[x])/10

    standar = (bakat[x] - 10)/10

    arr_bakat[0] = pemula

    arr_bakat[1] = standar

elif bakat[x] >= 20 and bakat[x] <= 40:

    standar = 1

    arr_bakat[1] = standar

elif bakat[x] > 40 and bakat[x] < 50:

    standar = (50 - bakat[x])/10

    ahli = (bakat[x] - 40)/10

    arr_bakat[1] = standar

    arr_bakat[2] = ahli

elif bakat[x] >= 50:

    ahli = 1

    arr_bakat[2] = ahli


#inference

L = []
```

```

        if arr_bakat[0] == pemula and arr_minat[0] ==
rendah:

            L.append(min(arr_bakat[0],arr_minat[0]))

        if arr_bakat[0] == pemula and arr_minat[1] ==
kurang:

            L.append(min(arr_bakat[0],arr_minat[1]))

        if arr_bakat[0] == pemula and arr_minat[2] ==
normal:

            L.append(min(arr_bakat[0],arr_minat[2]))

        if arr_bakat[0] == pemula and arr_minat[3] ==
tinggi:

            L.append(min(arr_bakat[0],arr_minat[3]))

        if arr_bakat[0] == pemula and arr_minat[4] ==
sangatTinggi:

            L.append(min(arr_bakat[0],arr_minat[4]))

        if arr_bakat[1] == standar and arr_minat[0] ==
rendah:

            L.append(min(arr_bakat[1],arr_minat[0]))

    max_L = max(L)

    M = []

    if arr_bakat[1] == standar and arr_minat[1] ==
kurang:

        M.append(min(arr_bakat[1],arr_minat[1]))

    if arr_bakat[1] == standar and arr_minat[2] ==
normal:

        M.append(min(arr_bakat[1],arr_minat[2]))

    if arr_bakat[1] == standar and arr_minat[3] ==
tinggi:

        M.append(min(arr_bakat[1],arr_minat[3]))

```

```

        if arr_bakat[1] == standar and arr_minat[4] ==
sangatTinggi:

            M.append(min(arr_bakat[1],arr_minat[4]))

        max_M = max(M)

    S = []

    if arr_bakat[2] == ahli and arr_minat[0] == rendah:

        S.append(min(arr_bakat[2],arr_minat[0]))

    if arr_bakat[2] == ahli and arr_minat[1] == kurang:

        S.append(min(arr_bakat[2],arr_minat[1]))

    if arr_bakat[2] == ahli and arr_minat[2] == normal:

        S.append(min(arr_bakat[2],arr_minat[2]))

    if arr_bakat[2] == ahli and arr_minat[3] == tinggi:

        S.append(min(arr_bakat[2],arr_minat[3]))

    if arr_bakat[2] == ahli and arr_minat[4] ==
sangatTinggi:

        S.append(min(arr_bakat[2],arr_minat[4]))

    max_S = max(S)

```

Program melakukan perhitungan defuzzifikasi. hasil dari perhitungan disimpan dalam variabel hasil.

```

#defuzzification

bagi = max_L + max_M + max_S

rs = ((max_S*20) + (max_M*40) + (max_L*60)) / bagi

hasil.append([id[x],rs])

```

Program menyimpan hasil perhitungan pada kolom “Kecocokan” .

```

#output

hasil_akhir = sorted(hasil, key = lambda x: x[1],
reverse=True)

hasil_xlsx = {'Kecocokan': hasil_akhir[:10]}

result_xlsx = pd.DataFrame(hasil_xlsx, columns =
['Kecocokan'])

result_xlsx.to_excel('luaran.xlsx')

```

	A	B	C	D	E
1	Kecocokan				
2	0	['C', 60.000000000000001]			
3	1	['J', 60.0]			
4	2	['N', 60.0]			
5	3	['P', 60.0]			
6	4	['V', 60.0]			
7	5	['AU', 60.0]			
8	6	['AY', 60.0]			
9	7	['BV', 60.0]			
10	8	['CG', 60.0]			
11	9	['CM', 60.0]			
12					
13					
14					
15					

Berdasarkan hasil perhitungan, program berhasil mendapatkan 10 list data terbaik dengan nilai 60.

BAB III

KESIMPULAN

Penggunaan *fuzzy* sebagai sistem rekomendasi pekerjaan berdasarkan minat dan bakat cocok diterapkan pada generasi muda zaman sekarang. karena dengan adanya pertimbangan minat dan bakat yang sesuai maka akan menghasilkan talenta yang memiliki performa lebih efisien.

DAFTAR PUSTAKA

- Elisawati. (2017). Sistem Deteksi Objek Dengan Menggunakan Sensor Ultrasonik Berbasis Fuzzy. Jurnal Informatika, Manajemen Dan Komputer, 9(1), 10–14. Retrieved From [Http://Ejournal.Stmikdumai.Ac.Id/Index.Ph p/Path/Article/View/58](http://ejournal.stmikdumai.ac.id/index.php/Path/Article/View/58).
- Setiadi. 2009. Himpunan & Logika Samar serta Aplikasinya, Yogyakarta: Graha Ilmu.
- Stewart, James. 2001. Kalkulus. Jakarta: Erlangga.