

**NAMA : AHMAT RAJALLI NASUTION**  
**NIM : 191011400961**

## **LOGIKA FUZZY**

### **Penerapan Metode Tsukamoto**

Jumlah variabel: 3

Sebutkan nama variabel: permintaan

Sebutkan nama variabel: persediaan

Sebutkan nama variabel: produksi

permintaan

naik : 5000

turun : 1000

persediaan

naik : 600

turun : 100

produksi

naik : 7000

turun : 2000

{'permintaan\_naik': 5000, 'permintaan\_turun': 1000, 'persediaan\_naik': 600,  
'persediaan\_turun': 100, 'produksi\_naik': 7000, 'produksi\_turun': 2000}

Jumlah variabel yang diketahui : 2

Nama variabel : permintaan

Nilai : 4000

Nama variabel : persediaan

Nilai : 300

{'permintaan': 4000, 'persediaan': 300}

Variabel yang ditanyakan : produksi

{'permintaan\_naik': 0.75, 'permintaan\_turun': 0.25, 'persediaan\_naik': 0.4, 'persediaan\_turun': 0.6}

Masukkan jumlah peraturan : 4

Kondisi 1(naik/turun): permintaan\_turun

Kondisi 2(naik/turun): persediaan\_naik

Kesimpulan(naik/turun): turun

Kondisi 1(naik/turun): permintaan\_turun

Kondisi 2(naik/turun): persediaan\_turun

Kesimpulan(naik/turun): turun

Kondisi 1(naik/turun): permintaan\_naik

Kondisi 2(naik/turun): persediaan\_naik

Kesimpulan(naik/turun): naik

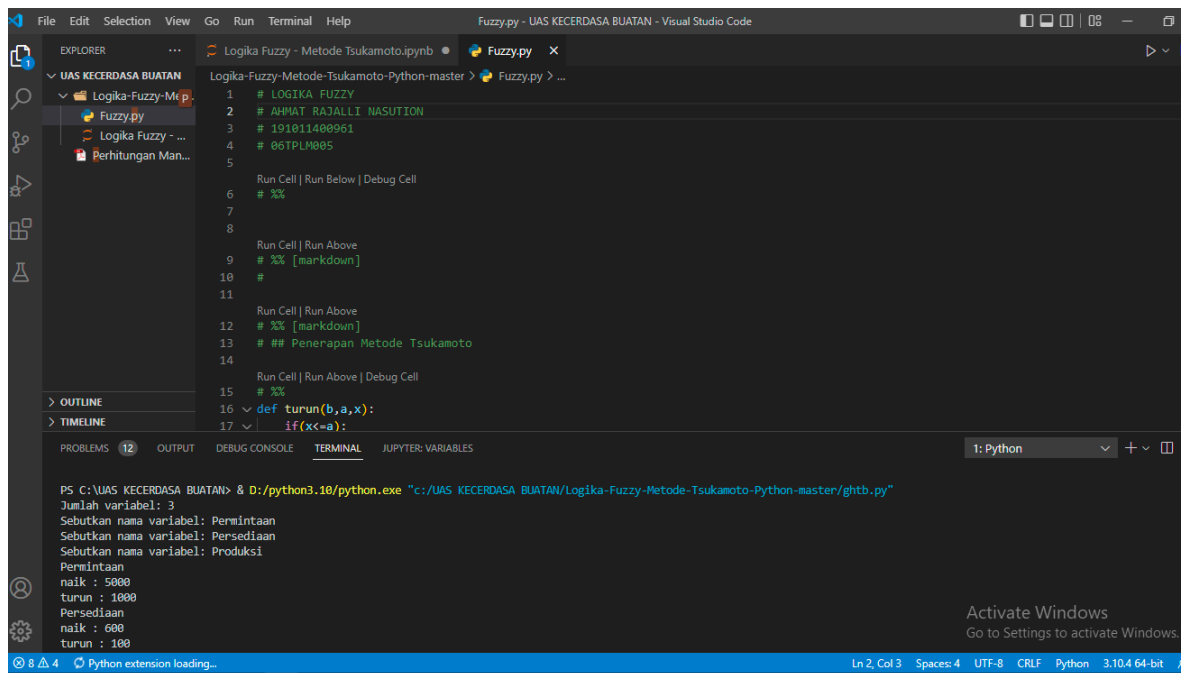
Kondisi 1(naik/turun): permintaan\_naik

Kondisi 2(naik/turun): persediaan\_turun

Kesimpulan(naik/turun): naik

[0.25, 0.25, 0.4, 0.6]

[5750.0, 5750.0, 4000.0, 5000.0]

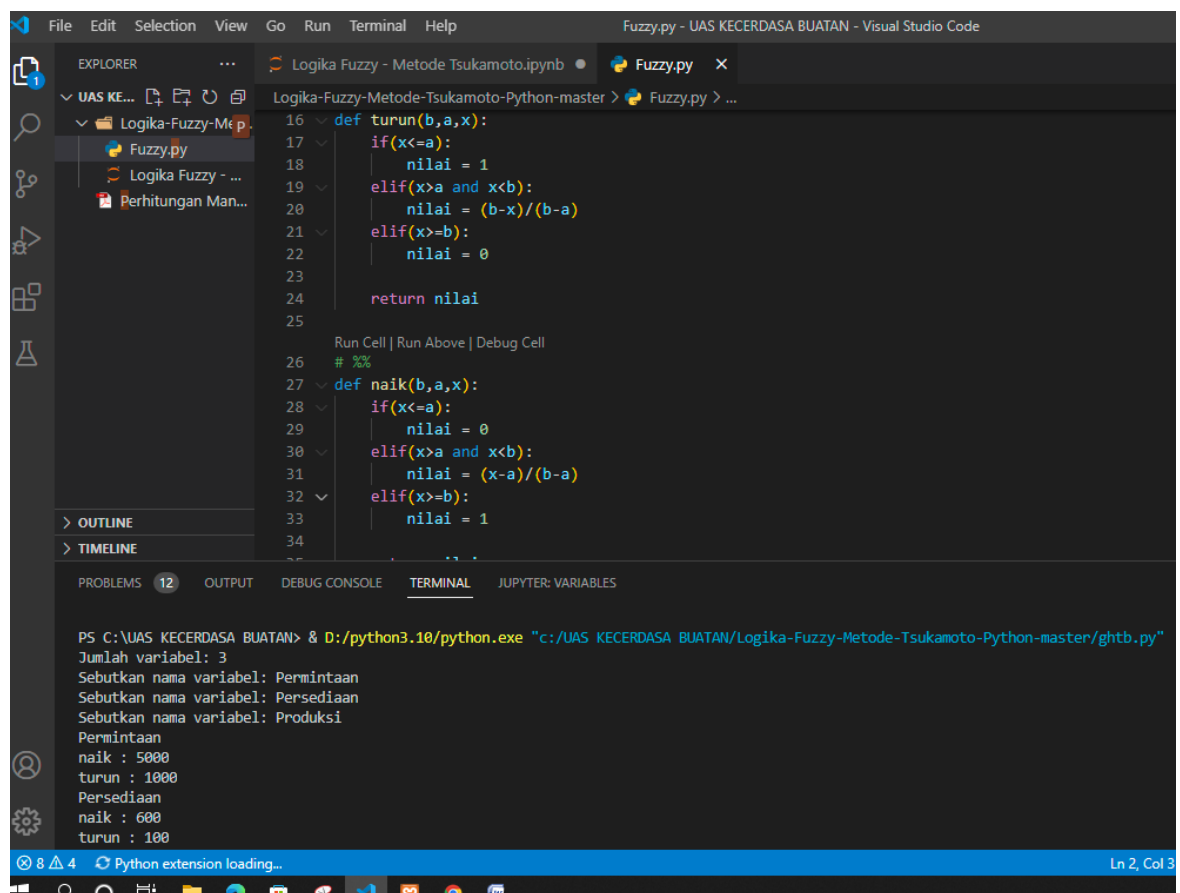


```
1 # LOGIKA FUZZY
2 # AHMAT RAJALLI NASUTION
3 # 191011400961
4 # 06TPLM005
5
6 Run Cell | Run Below | Debug Cell
7 # %%
8
9 Run Cell | Run Above
10 # %% [markdown]
11 #
12 Run Cell | Run Above
13 # %% [markdown]
14 # ## Penerapan Metode Tsukamoto
15
16 Run Cell | Run Above | Debug Cell
17 # %%
18 def turun(b,a,x):
19     if(x<=a):
```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER VARIABLES

PS C:\UAS KECERDASA BUATAN> & D:\python3.10\python.exe "c:\UAS KECERDASA BUATAN\Logika-Fuzzy-Metode-Tsukamoto-Python-master\ghbt.py"

Jumlah variabel: 3  
Sebutkan nama variabel: Permintaan  
Sebutkan nama variabel: Persediaan  
Sebutkan nama variabel: Produksi  
Permintaan  
naik : 5000  
turun : 1000  
Persediaan  
naik : 600  
turun : 100

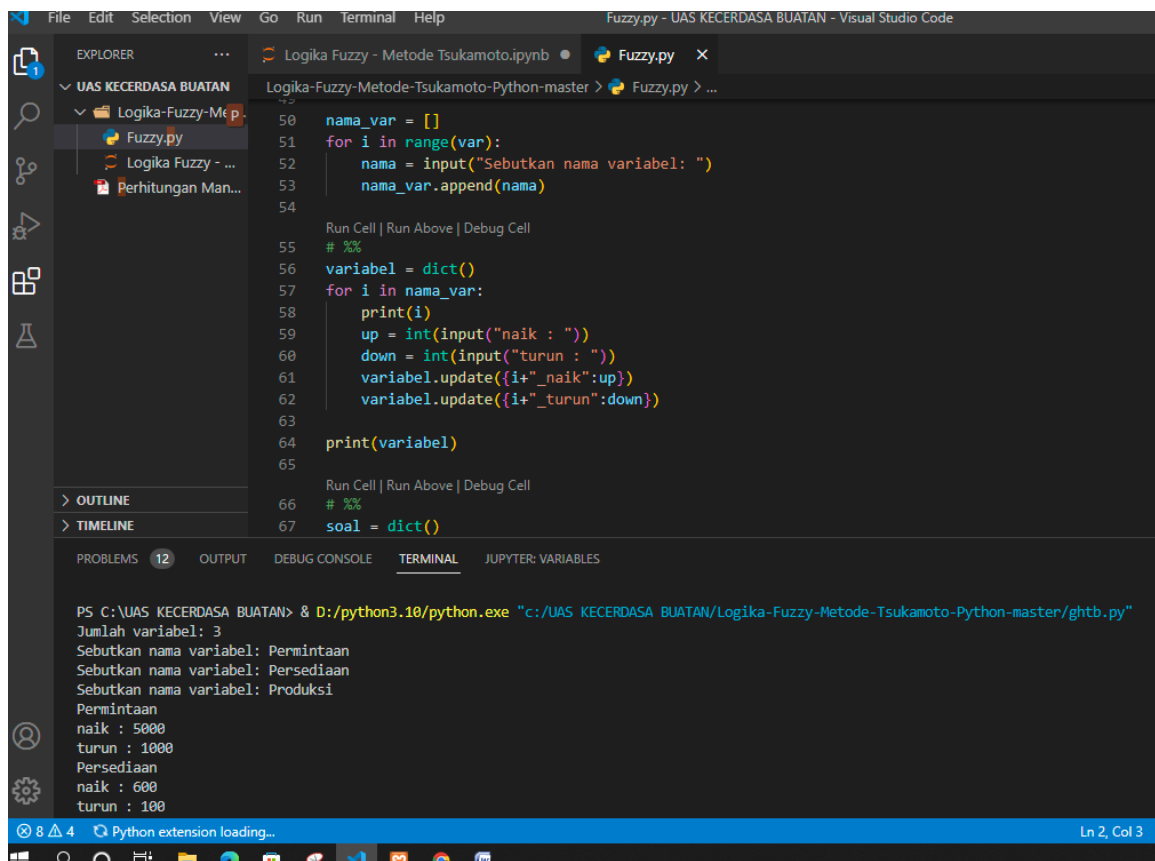


```
16 def turun(b,a,x):
17     if(x<=a):
18         nilai = 1
19     elif(x>a and x<b):
20         nilai = (b-x)/(b-a)
21     elif(x>=b):
22         nilai = 0
23
24     return nilai
25
26 # %%
27 def naik(b,a,x):
28     if(x<=a):
29         nilai = 0
30     elif(x>a and x<b):
31         nilai = (x-a)/(b-a)
32     elif(x>=b):
33         nilai = 1
34
```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER VARIABLES

PS C:\UAS KECERDASA BUATAN> & D:\python3.10\python.exe "c:\UAS KECERDASA BUATAN\Logika-Fuzzy-Metode-Tsukamoto-Python-master\ghbt.py"

Jumlah variabel: 3  
Sebutkan nama variabel: Permintaan  
Sebutkan nama variabel: Persediaan  
Sebutkan nama variabel: Produksi  
Permintaan  
naik : 5000  
turun : 1000  
Persediaan  
naik : 600  
turun : 100

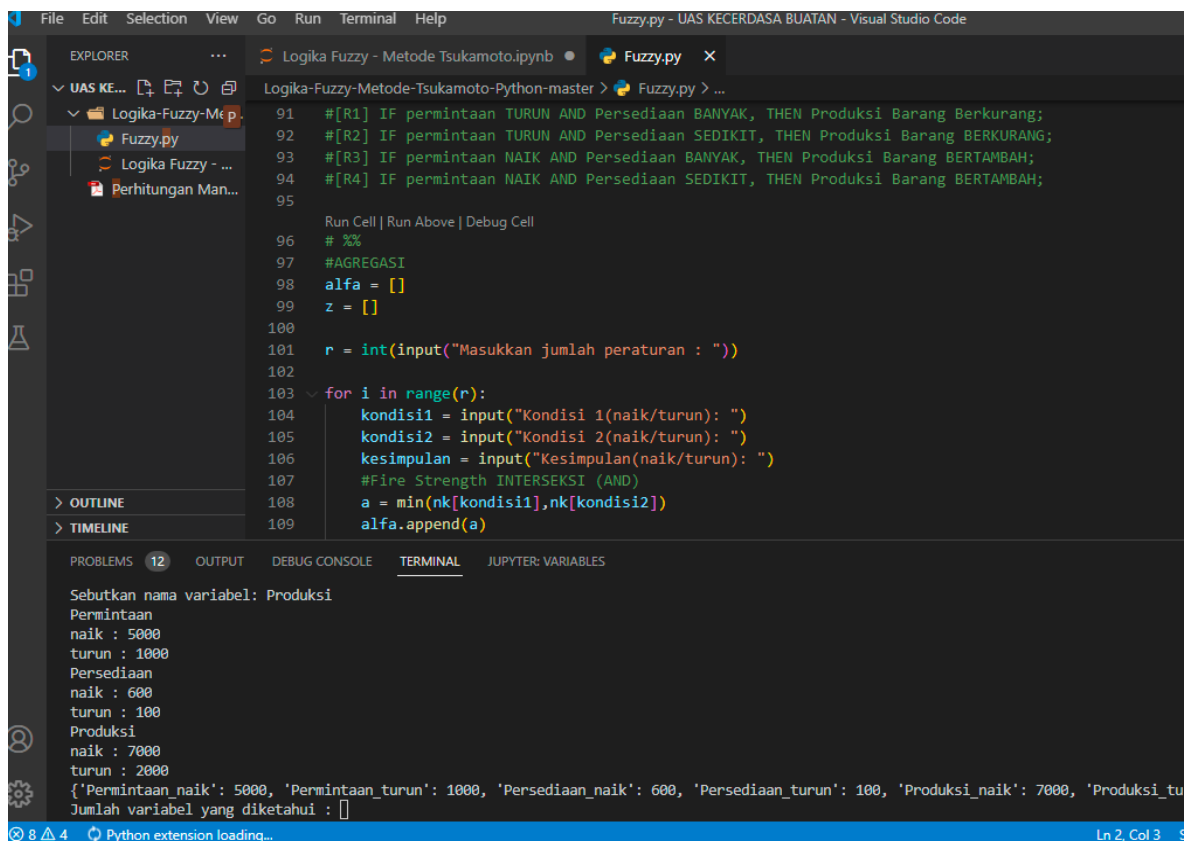


Visual Studio Code interface showing a Python script for fuzzy logic. The script defines a list of variables, takes user input for each, and updates a dictionary. The terminal shows the execution of the script with sample inputs.

```
50 nama_var = []
51 for i in range(var):
52     nama = input("Sebutkan nama variabel: ")
53     nama_var.append(nama)
54
55 # %%
56 variabel = dict()
57 for i in nama_var:
58     print(i)
59     up = int(input("naik : "))
60     down = int(input("turun : "))
61     variabel.update({i+"_naik":up})
62     variabel.update({i+"_turun":down})
63
64 print(variabel)
65
66 # %%
67 soal = dict()
```

Terminal output:

```
PS C:\UAS KECERDASA BUATAN> & D:/python3.10/python.exe "c:/UAS KECERDASA BUATAN/Logika-Fuzzy-Metode-Tsukamoto-Python-master/ghbt.py"
Jumlah variabel: 3
Sebutkan nama variabel: Permintaan
Sebutkan nama variabel: Persediaan
Sebutkan nama variabel: Produksi
Permintaan
naik : 5000
turun : 1000
Persediaan
naik : 600
turun : 100
```



Visual Studio Code interface showing a Python script for fuzzy logic. The script defines four fuzzy rules, takes user input for the number of rules and conditions, and calculates the minimum of the two conditions for each rule. The terminal shows the execution of the script with sample inputs.

```
91 #[R1] IF permintaan TURUN AND Persediaan BANYAK, THEN Produksi Barang Berkurang;
92 #[R2] IF permintaan TURUN AND Persediaan SEDIKIT, THEN Produksi Barang BERKURANG;
93 #[R3] IF permintaan NAIK AND Persediaan BANYAK, THEN Produksi Barang BERTAMBAH;
94 #[R4] IF permintaan NAIK AND Persediaan SEDIKIT, THEN Produksi Barang BERTAMBAH;
95
96 # %%
97 #AGREGASI
98 alfa = []
99 z = []
100
101 r = int(input("Masukkan jumlah peraturan : "))
102
103 for i in range(r):
104     kondisi1 = input("Kondisi 1(naik/turun): ")
105     kondisi2 = input("Kondisi 2(naik/turun): ")
106     kesimpulan = input("Kesimpulan(naik/turun): ")
107     #Fire Strength INTERSEKSI (AND)
108     a = min(nk[kondisi1],nk[kondisi2])
109     alfa.append(a)
```

Terminal output:

```
Sebutkan nama variabel: Produksi
Permintaan
naik : 5000
turun : 1000
Persediaan
naik : 600
turun : 100
Produksi
naik : 7000
turun : 2000
{'Permintaan_naik': 5000, 'Permintaan_turun': 1000, 'Persediaan_naik': 600, 'Persediaan_turun': 100, 'Produksi_naik': 7000, 'Produksi_turun': 2000}
Jumlah variabel yang diketahui : []
```



