

Nama : Tirta Agung Jati

NIM : 20106050001

Mata Kuliah : Logika Fuzzy

Tugas Fuzzy Inference System

Model Yang Saya buat ialah model Mamdani, dimana model disini terdiri dari Hanya bisa AND method dengan fungsi Min. Implicationnya menggunakan fungsi Min. Aggregation menggunakan fungsi Max. Untuk defuzzification menggunakan fungsi MOM.

Inisiasi Input Output

```
In [1]: var_input = []
        var_output = []

        def variabel_input():
            jumlah = int(input("Berapa Banyak variabel input yang anda miliki : "))
            print("\nSilahkan Masukkan nama nama variabel input anda\n")
            for i in range(jumlah):
                nilai = str(input(f"Variabel {i+1} : "))
                var_input.append(nilai)

        def variabel_output():
            nilai = str(input("Silahkan Masukkan nama variabel output anda : "))
            var_output.append(nilai)
```

```
In [2]: variabel_input()

Berapa Banyak variabel input yang anda miliki : 3

Silahkan Masukkan nama nama variabel input anda

Variabel 1 : umur
Variabel 2 : tinggi_badan
Variabel 3 : berat_badan
```

```
In [3]: variabel_output()

Silahkan Masukkan nama variabel output anda : status_gizi
```

Inisiasi himpunan variabel input dan output

```
In [4]: himpunan_var_input = []
        def himpunan_variabel_input():
            for i in var_input:
                jumlah = int(input(f"Ada Berapa Himpunan pada variabel {i} : "))
                for j in range(jumlah):
                    nilai = str(input(f"Himpunan variabel {i} ke-{j+1} : "))
                    himpunan_var_input.append([i, nilai])
```

```
In [5]: himpunan_variabel_input()

Ada Berapa Himpunan pada variabel umur : 3
```

```
Himpunan variabel umur ke-1 : remaja
Himpunan variabel umur ke-2 : dewasa
Himpunan variabel umur ke-3 : tua
Ada Berapa Himpunan pada variabel tinggi_badan : 3
Himpunan variabel tinggi_badan ke-1 : pendek
Himpunan variabel tinggi_badan ke-2 : sedang
Himpunan variabel tinggi_badan ke-3 : tinggi
Ada Berapa Himpunan pada variabel berat_badan : 3
Himpunan variabel berat_badan ke-1 : ringan
Himpunan variabel berat_badan ke-2 : normal
Himpunan variabel berat_badan ke-3 : berat
```

```
In [6]: print(himpunan_var_input)
```

```
[['umur', 'remaja'], ['umur', 'dewasa'], ['umur', 'tua'], ['tinggi_badan', 'pendek'],
 ['tinggi_badan', 'sedang'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'ringan'], ['berat_badan', 'normal'], ['berat_badan', 'berat']]
```

```
In [7]: himpunan_var_output = []
def himpunan_variabel_output():
    for i in var_output:
        jumlah = int(input(f"Ada Berapa Himpunan pada variabel {i} : "))
        for j in range(jumlah):
            nilai = str(input(f"Himpunan variabel {i} ke-{j+1} : "))
            himpunan_var_output.append([i, nilai])
```

```
In [8]: himpunan_variabel_output()
```

```
Ada Berapa Himpunan pada variabel status_gizi : 3
Himpunan variabel status_gizi ke-1 : kurang
Himpunan variabel status_gizi ke-2 : normal
Himpunan variabel status_gizi ke-3 : berlebihan
```

```
In [9]: print(himpunan_var_output)
```

```
[['status_gizi', 'kurang'], ['status_gizi', 'normal'], ['status_gizi', 'berlebihan']]
```

Inisiasi semesta pembicaraan variabel input dan output

```
In [10]: semesta_pembicaraan_var_input = []
def semesta_pembicaraan_variabel_input():
    for i in var_input:
        batas_bawah = int(input(f"Masukan nilai batas bawah semesta pembicaraan pada var {i} : "))
        batas_atas = int(input(f"Masukan nilai batas atas semesta pembicaraan pada variabel {i} : "))
        semesta_pembicaraan_var_input.append([i, [batas_bawah, batas_atas]])
```

```
In [11]: semesta_pembicaraan_variabel_input()
```

```
Masukan nilai batas bawah semesta pembicaraan pada variabel umur : 10
Masukan nilai batas atas semesta pembicaraan pada variabel umur : 40
Masukan nilai batas bawah semesta pembicaraan pada variabel tinggi_badan : 130
Masukan nilai batas atas semesta pembicaraan pada variabel tinggi_badan : 185
Masukan nilai batas bawah semesta pembicaraan pada variabel berat_badan : 30
Masukan nilai batas atas semesta pembicaraan pada variabel berat_badan : 90
```

```
In [12]: semesta_pembicaraan_var_input
```

```
Out[12]: [['umur', [10, 40]], ['tinggi_badan', [130, 185]], ['berat_badan', [30, 90]]]
```

```
In [13]: semesta_pembicaraan_var_output = []
def semesta_pembicaraan_variabel_output():
    for i in var_output:
        batas_bawah = int(input(f"Masukan nilai batas bawah semesta pembicaraan pada var {i} : "))
        batas_atas = int(input(f"Masukan nilai batas atas semesta pembicaraan pada variabel {i} : "))
        semesta_pembicaraan_var_output.append([i, [batas_bawah, batas_atas]])
```

```
batas_atas = int(input(f"Masukkan nilai batas atas semesta pembicaraan pada varia
semesta_pembicaraan_var_output.append([i,[batas_bawah,batas_atas]]))
```

```
In [14]: semesta_pembicaraan_variabel_output()
```

Masukan nilai batas bawah semesta pembicaraan pada variabel status_gizi : 10
Masukan nilai batas atas semesta pembicaraan pada variabel status_gizi : 40

Inisiasi Membership Function

```
In [15]: pemetaan_grafik_input = []
domain_var_input = []
def membership_function():
    print("Aplikasi ini hanya memiliki fungsi segitiga dan trapesium")
    for i in range(len(himpunan_var_input)):
        tugas = input(f"Untuk domain variabel {himpunan_var_input[i][0]} {himpunan_var_i
        "Jika anda ingin menggunakan fungsi segitiga ketikkan 1\n"+
        "Jika anda ingin menggunakan fungsi trapesium ketikkan 2\n"+
        "input(1/2) : ")
        if tugas == "1":
            a = float(input(f"masukkan nilai a : "))
            b = float(input(f"masukkan nilai b : "))
            c = float(input(f"masukkan nilai c : "))
            pemetaan_grafik_input.append([himpunan_var_input[i][0],himpunan_var_input[i]
            domain_var_input.insert(i,[himpunan_var_input[i][0],himpunan_var_input[i][1]
        if tugas == "2":
            a = float(input("masukkan nilai a : "))
            b = float(input("masukkan nilai b : "))
            c = float(input("masukkan nilai c : "))
            d = float(input("masukkan nilai d : "))
            domain_var_input.insert(i,[himpunan_var_input[i][0],himpunan_var_input[i][1]
            if a == b:
                pemetaan_grafik_input.append([himpunan_var_input[i][0],himpunan_var_inpu
            elif d == c:
                pemetaan_grafik_input.append([himpunan_var_input[i][0],himpunan_var_inpu
            else:
                pemetaan_grafik_input.append([himpunan_var_input[i][0],himpunan_var_inpu
```

```
In [16]: membership_function()
```

```
Aplikasi ini hanya memiliki fungsi segitiga dan trapesium
Untuk domain variabel umur remaja
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 10
masukkan nilai b : 10
masukkan nilai c : 15
masukkan nilai d : 18
Untuk domain variabel umur dewasa
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 16
masukkan nilai b : 20
masukkan nilai c : 24
masukkan nilai d : 27
Untuk domain variabel umur tua
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 26
masukkan nilai b : 32
masukkan nilai c : 40
```

```

masukkan nilai d : 40
Untuk domain variabel tinggi_badan pendek
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 130
masukkan nilai b : 130
masukkan nilai c : 140
masukkan nilai d : 150
Untuk domain variabel tinggi_badan sedang
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 145
masukkan nilai b : 155
masukkan nilai c : 160
masukkan nilai d : 170
Untuk domain variabel tinggi_badan tinggi
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 165
masukkan nilai b : 180
masukkan nilai c : 185
masukkan nilai d : 185
Untuk domain variabel berat_badan ringan
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 30
masukkan nilai b : 30
masukkan nilai c : 40
masukkan nilai d : 45
Untuk domain variabel berat_badan normal
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 43
masukkan nilai b : 50
masukkan nilai c : 55
masukkan nilai d : 60
Untuk domain variabel berat_badan berat
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 56
masukkan nilai b : 65
masukkan nilai c : 90
masukkan nilai d : 90

```

In [17]: pemetaan_grafik_input

```

Out[17]: [['umur', 'remaja', 'trapesium', [[10.0, 10.0, 15.0, 18.0], [1, 1, 1, 0]]],
          ['umur', 'dewasa', 'trapesium', [[16.0, 20.0, 24.0, 27.0], [0, 1, 1, 0]]],
          ['umur', 'tua', 'trapesium', [[26.0, 32.0, 40.0, 40.0], [0, 1, 1, 1]]],
          ['tinggi_badan',
           'pendek',
           'trapesium',
           [[130.0, 130.0, 140.0, 150.0], [1, 1, 1, 0]]],
          ['tinggi_badan',
           'sedang',
           'trapesium',
           [[145.0, 155.0, 160.0, 170.0], [0, 1, 1, 0]]],
          ['tinggi_badan',
           'tinggi',
           'trapesium',

```

```

[[165.0, 180.0, 185.0, 185.0], [0, 1, 1, 1]],
['berat_badan',
 'ringan',
 'trapesium',
 [[30.0, 30.0, 40.0, 45.0], [1, 1, 1, 0]],
['berat_badan',
 'normal',
 'trapesium',
 [[43.0, 50.0, 55.0, 60.0], [0, 1, 1, 0]],
['berat_badan',
 'berat',
 'trapesium',
 [[56.0, 65.0, 90.0, 90.0], [0, 1, 1, 1]]]]

```

In [18]: domain_var_input

Out[18]:

```

[['umur', 'remaja', 10.0, 18.0],
 ['umur', 'dewasa', 16.0, 27.0],
 ['umur', 'tua', 26.0, 40.0],
 ['tinggi_badan', 'pendek', 130.0, 150.0],
 ['tinggi_badan', 'sedang', 145.0, 170.0],
 ['tinggi_badan', 'tinggi', 165.0, 185.0],
 ['berat_badan', 'ringan', 30.0, 45.0],
 ['berat_badan', 'normal', 43.0, 60.0],
 ['berat_badan', 'berat', 56.0, 90.0]]

```

In [19]:

```

pemetaan_grafik_output = []
domain_var_output = []
def membership_function_output():
    print("Aplikasi ini hanya memiliki fungsi segitiga dan trapesium")
    for i in range(len(himpunan_var_output)):
        tugas = input(f"Untuk domain variabel {himpunan_var_output[i][0]} {himpunan_var_ou
        'Jika anda ingin menggunakan fungsi segitiga ketikkan 1\n"+
        'Jika anda ingin menggunakan fungsi trapesium ketikkan 2\n"+
        "input(1/2) : ")
        if tugas == "1":
            a = float(input(f"masukkan nilai a : "))
            b = float(input(f"masukkan nilai b : "))
            c = float(input(f"masukkan nilai c : "))
            pemetaan_grafik_output.append([himpunan_var_output[i][0],himpunan_var_ou
            domain_var_output.insert(i,[himpunan_var_output[i][0],himpunan_var_output[i]
        elif tugas == "2":
            a = float(input("masukkan nilai a : "))
            b = float(input("masukkan nilai b : "))
            c = float(input("masukkan nilai c : "))
            d = float(input("masukkan nilai d : "))
            domain_var_output.insert(i,[himpunan_var_output[i][0],himpunan_var_output[i]
            if a == b:
                pemetaan_grafik_output.append([himpunan_var_output[i][0],himpunan_var_ou
            elif d == c:
                pemetaan_grafik_output.append([himpunan_var_output[i][0],himpunan_var_ou
            else:
                pemetaan_grafik_output.append([himpunan_var_output[i][0],himpunan_var_ou

```

In [20]: membership_function_output()

```

Aplikasi ini hanya memiliki fungsi segitiga dan trapesium
Untuk domain variabel status_gizi kurang
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 10
masukkan nilai b : 10
masukkan nilai c : 14
masukkan nilai d : 18
Untuk domain variabel status_gizi normal

```

Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 18
masukkan nilai b : 20
masukkan nilai c : 21
masukkan nilai d : 24
Untuk domain variabel status_gizi berlebihan
Jika anda ingin menggunakan fungsi segitiga ketikkan 1
Jika anda ingin menggunakan fungsi trapesium ketikkan 2
input(1/2) : 2
masukkan nilai a : 23
masukkan nilai b : 27
masukkan nilai c : 40
masukkan nilai d : 40

In [21]: pemetaan_grafik_output

Out[21]:

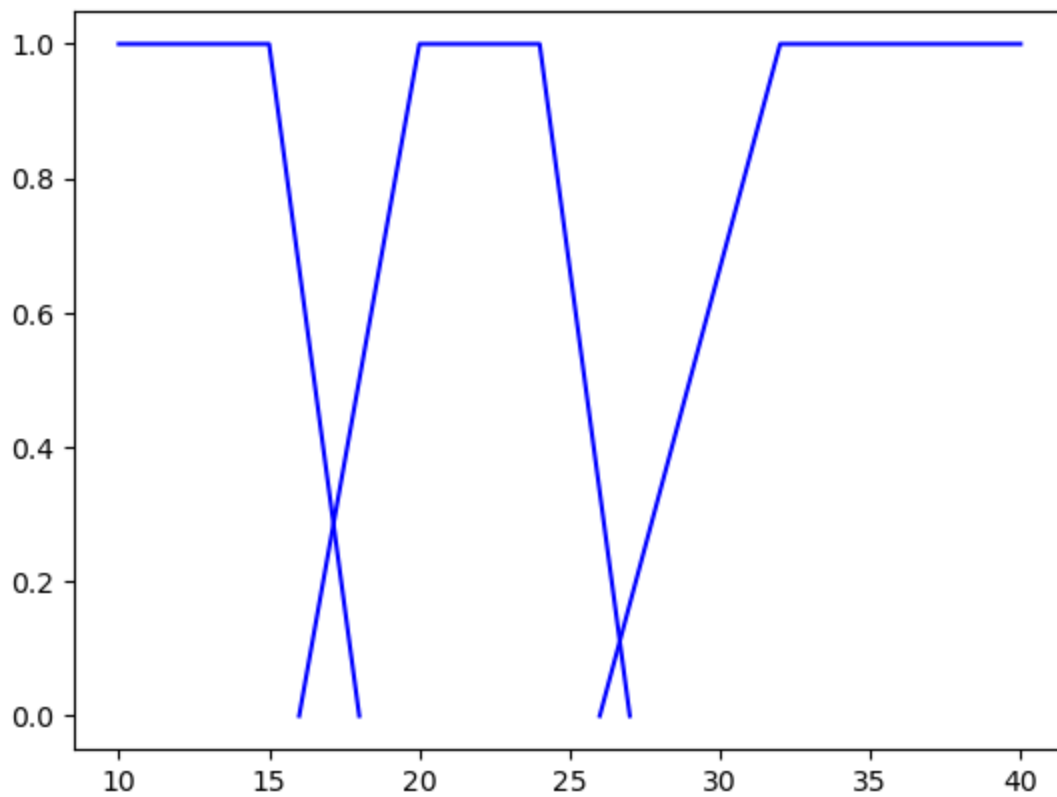
```
[['status_gizi',  
  'kurang',  
  'trapesium',  
  [[10.0, 10.0, 14.0, 18.0], [1, 1, 1, 0]]],  
 ['status_gizi',  
  'normal',  
  'trapesium',  
  [[18.0, 20.0, 21.0, 24.0], [0, 1, 1, 0]]],  
 ['status_gizi',  
  'berlebihan',  
  'trapesium',  
  [[23.0, 27.0, 40.0, 40.0], [0, 1, 1, 1]]]]
```

In [22]:

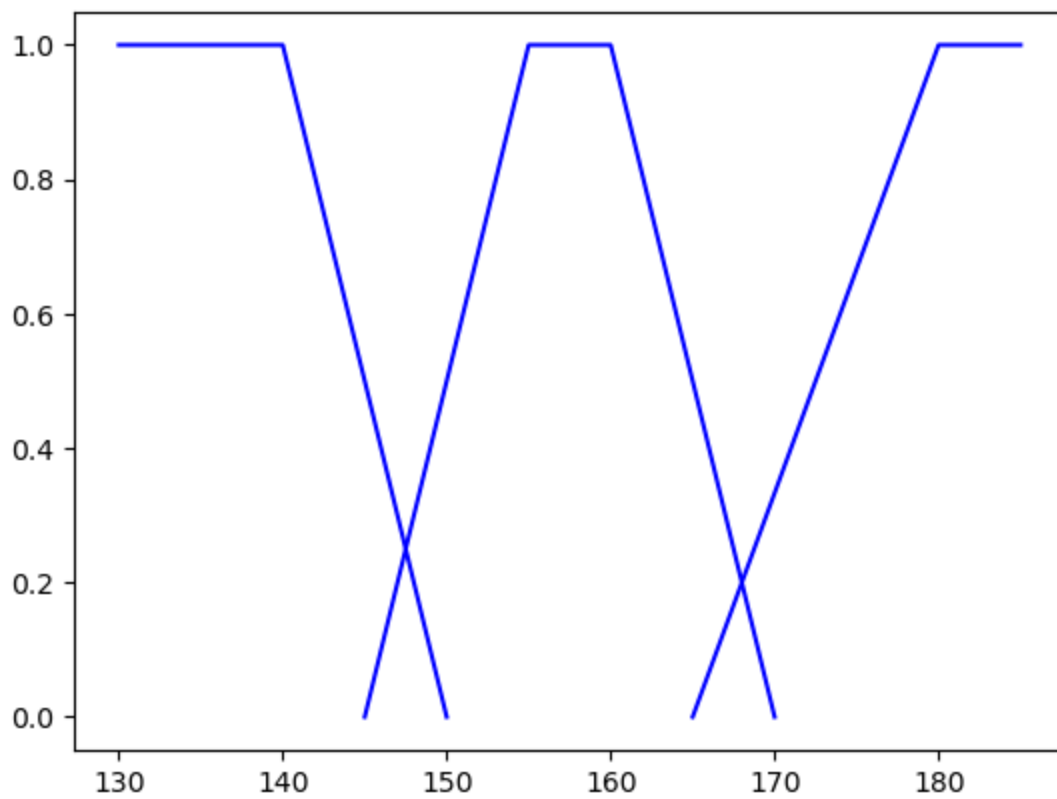
```
import matplotlib.pyplot as plt #Library  
  
def gambar_input():  
    for i in range(len(var_input)):  
        for j in range(len(pemetaan_grafik_input)):  
            if pemetaan_grafik_input[j][0] == var_input[i]:  
                plt.plot(pemetaan_grafik_input[j][3][0], pemetaan_grafik_input[j][3][1], c=  
                    pemetaan_grafik_input[j][2])  
                plt.suptitle(pemetaan_grafik_input[j][0])  
plt.show()
```

In [23]: gambar_input()

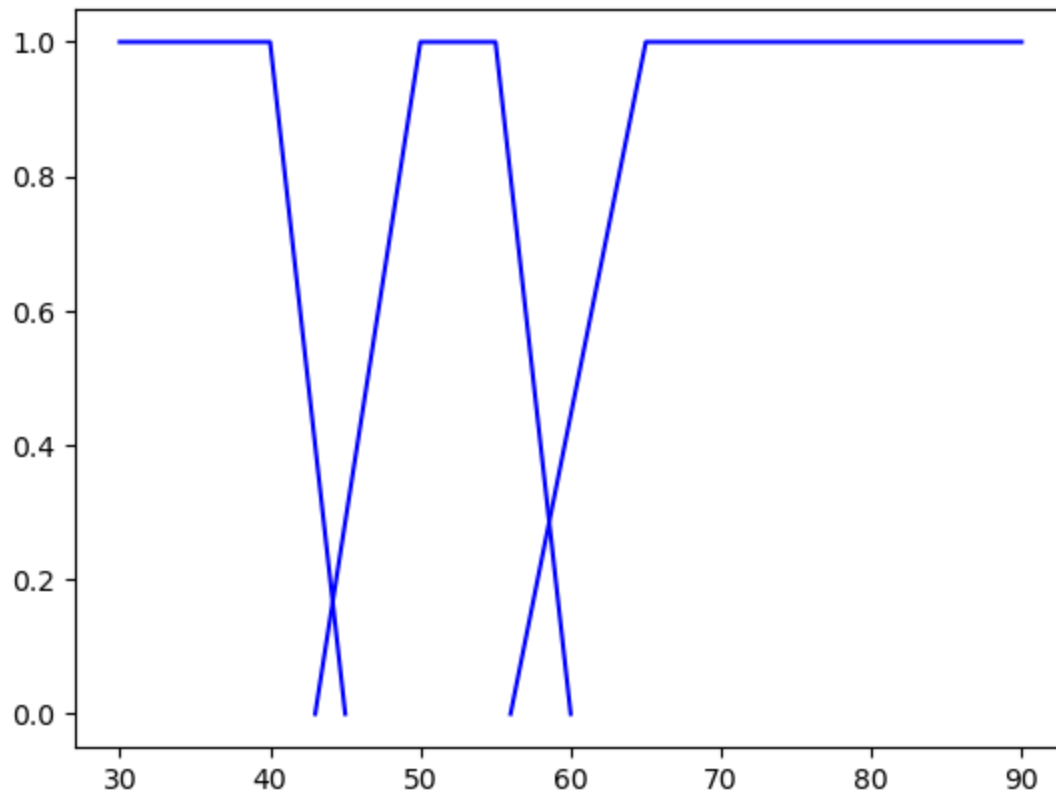
umur



tinggi_badan



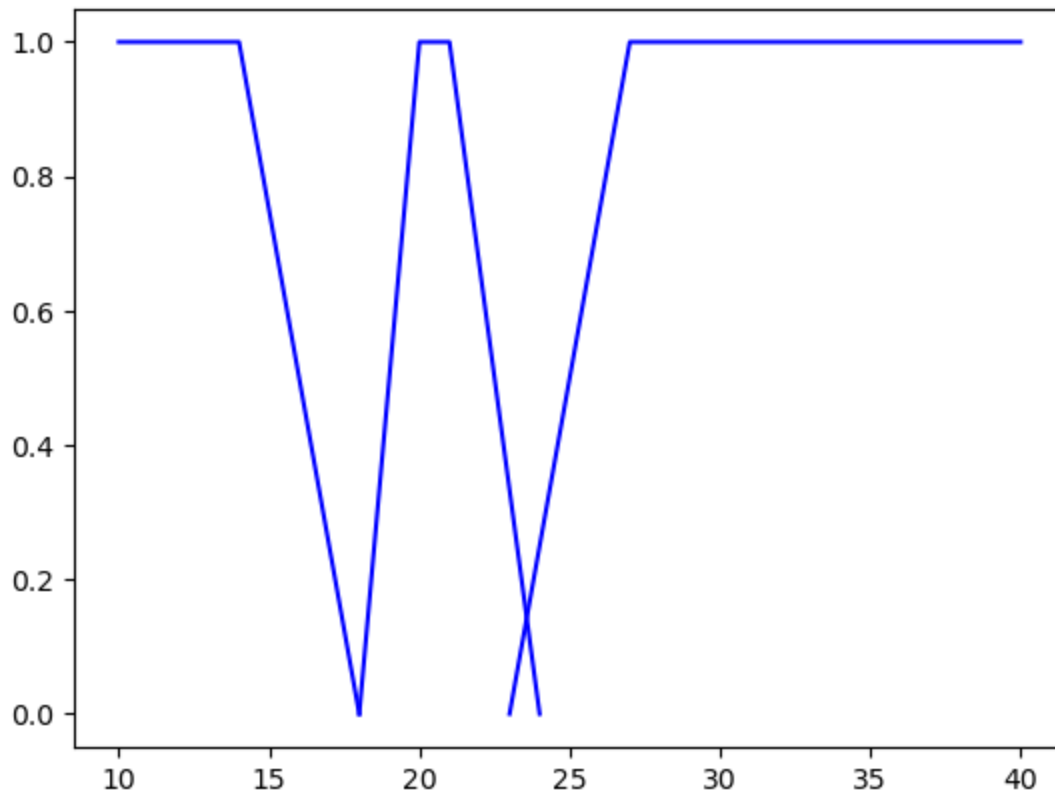
berat_badan



```
In [24]: def gambar_output():  
    for i in range(len(var_output)):  
        for j in range(len(pemetaan_grafik_output)):  
            if pemetaan_grafik_output[j][0] == var_output[i]:  
                plt.plot(pemetaan_grafik_output[j][3][0], pemetaan_grafik_output[j][3][1])  
                plt.suptitle(pemetaan_grafik_output[j][0])  
    plt.show()
```

```
In [25]: gambar_output()
```


status_gizi



Inisiasi rules

```
In [34]: rule = []
def rules():
    x = 0
    print("berikut contoh format penulisan rules : if ipk = buruk AND gaji = kecil THEN
    while x == 0:
        nilai = input("Masukkan rules anda : if ")
        rule.append(nilai)
        tugas = input("Apakah anda ingin memasukkan rules lagi ? \njika iya ketikkan 1 \
    if tugas == "1":
        x = 0
    elif tugas == "0":
        x = 1
```

```
In [35]: rules()
```

berikut contoh format penulisan rules : if ipk = buruk AND gaji = kecil THEN nilai_kelay akan = rendah

Masukkan rules anda : if umur = remaja AND tinggi_badan = pendek AND berat_badan = ringan THEN status_gizi = normal

Apakah anda ingin memasukkan rules lagi ?

jika iya ketikkan 1

jika tidak inputkan 0

input(1/0) : 1

Masukkan rules anda : if umur = remaja AND tinggi_badan = sedang AND berat_badan = ringan THEN status_gizi = kurang

Apakah anda ingin memasukkan rules lagi ?

jika iya ketikkan 1

jika tidak inputkan 0

input(1/0) : 1

Masukkan rules anda : if umur = remaja AND tinggi_badan = tinggi AND berat_badan = ringan THEN status_gizi = kurang

```

Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = remaja AND tinggi_badan = pendek AND berat_badan = norma
l THEN status_gizi = normal
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = remaja AND tinggi_badan = sedang AND berat_badan = norma
l THEN status_gizi = normal
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = remaja AND tinggi_badan = tinggi AND berat_badan = norma
l THEN status_gizi = kurang
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = remaja AND tinggi_badan = pendek AND berat_badan = berat
THEN status_gizi = berlebihan
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = remaja AND tinggi_badan = sedang AND berat_badan = berat
THEN status_gizi = berlebihan
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = remaja AND tinggi_badan = tinggi AND berat_badan = berat
THEN status_gizi = normal
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 0

```

In [36]: rule

```

Out[36]: ['umur = remaja AND tinggi_badan = pendek AND berat_badan = ringan THEN status_gizi = no
rmal',
'umur = remaja AND tinggi_badan = sedang AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = remaja AND tinggi_badan = tinggi AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = remaja AND tinggi_badan = pendek AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = remaja AND tinggi_badan = sedang AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = remaja AND tinggi_badan = tinggi AND berat_badan = normal THEN status_gizi = ku
rang',
'umur = remaja AND tinggi_badan = pendek AND berat_badan = berat THEN status_gizi = ber
lebihan',
'umur = remaja AND tinggi_badan = sedang AND berat_badan = berat THEN status_gizi = ber
lebihan',
'umur = remaja AND tinggi_badan = tinggi AND berat_badan = berat THEN status_gizi = nor
mal']

```

In [37]: rules()

```

berikut contoh format penulisan rules : if ipk = buruk AND gaji = kecil THEN nilai_kelay
akan= rendah

```

```

Masukkan rules anda : if umur = dewasa AND tinggi_badan = pendek AND berat_badan = ringa
n THEN status_gizi = normal
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = dewasa AND tinggi_badan = sedang AND berat_badan = ringa
n THEN status_gizi = kurang
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = dewasa AND tinggi_badan = tinggi AND berat_badan = ringa
n THEN status_gizi = kurang
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = dewasa AND tinggi_badan = pendek AND berat_badan = norma
l THEN status_gizi = normal
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = dewasa AND tinggi_badan = sedang AND berat_badan = norma
l THEN status_gizi = normal
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = dewasa AND tinggi_badan = tinggi AND berat_badan = norma
l THEN status_gizi = normal
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = dewasa AND tinggi_badan = pendek AND berat_badan = berat
THEN status_gizi = berlebihan
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = dewasa AND tinggi_badan = sedang AND berat_badan = berat
THEN status_gizi = berlebihan
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = dewasa AND tinggi_badan = tinggi AND berat_badan = berat
THEN status_gizi = berlebihan
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 0

```

In [38]: rule

```

Out[38]: ['umur = remaja AND tinggi_badan = pendek AND berat_badan = ringan THEN status_gizi = no
rmal',
'umur = remaja AND tinggi_badan = sedang AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = remaja AND tinggi_badan = tinggi AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = remaja AND tinggi_badan = pendek AND berat_badan = normal THEN status_gizi = no
rmal',

```

```
'umur = remaja AND tinggi_badan = sedang AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = remaja AND tinggi_badan = tinggi AND berat_badan = normal THEN status_gizi = ku
rang',
'umur = remaja AND tinggi_badan = pendek AND berat_badan = berat THEN status_gizi = ber
lebih',
'umur = remaja AND tinggi_badan = sedang AND berat_badan = berat THEN status_gizi = ber
lebih',
'umur = remaja AND tinggi_badan = tinggi AND berat_badan = berat THEN status_gizi = nor
mal',
'umur = dewasa AND tinggi_badan = pendek AND berat_badan = ringan THEN status_gizi = no
rmal',
'umur = dewasa AND tinggi_badan = sedang AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = dewasa AND tinggi_badan = tinggi AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = dewasa AND tinggi_badan = pendek AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = dewasa AND tinggi_badan = sedang AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = dewasa AND tinggi_badan = tinggi AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = dewasa AND tinggi_badan = pendek AND berat_badan = berat THEN status_gizi = ber
lebih',
'umur = dewasa AND tinggi_badan = sedang AND berat_badan = berat THEN status_gizi = ber
lebih',
'umur = dewasa AND tinggi_badan = tinggi AND berat_badan = berat THEN status_gizi = ber
lebih']
```

In [39]: `rules()`

berikut contoh format penulisan rules : if ipk = buruk AND gaji = kecil THEN nilai_kelay akan = rendah

Masukkan rules anda : if umur = tua AND tinggi_badan = pendek AND berat_badan = ringan T HEN status_gizi = normal

Apakah anda ingin memasukkan rules lagi ?

jika iya ketikkan 1

jika tidak inputkan 0

input(1/0) : 1

Masukkan rules anda : if umur = tua AND tinggi_badan = sedang AND berat_badan = ringan T HEN status_gizi = kurang

Apakah anda ingin memasukkan rules lagi ?

jika iya ketikkan 1

jika tidak inputkan 0

input(1/0) : 1

Masukkan rules anda : if umur = tua AND tinggi_badan = tinggi AND berat_badan = ringan T HEN status_gizi = kurang

Apakah anda ingin memasukkan rules lagi ?

jika iya ketikkan 1

jika tidak inputkan 0

input(1/0) : 1

Masukkan rules anda : if umur = tua AND tinggi_badan = pendek AND berat_badan = normal T HEN status_gizi = normal

Apakah anda ingin memasukkan rules lagi ?

jika iya ketikkan 1

jika tidak inputkan 0

input(1/0) : 1

Masukkan rules anda : if umur = tua AND tinggi_badan = sedang AND berat_badan = normal T HEN status_gizi = normal

Apakah anda ingin memasukkan rules lagi ?

jika iya ketikkan 1

jika tidak inputkan 0

input(1/0) : 1

Masukkan rules anda : if umur = tua AND tinggi_badan = tinggi AND berat_badan = normal T HEN status_gizi = normal

```

Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = tua AND tinggi_badan = pendek AND berat_badan = berat TH
EN status_gizi = berlebihan
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = tua AND tinggi_badan = sedang AND berat_badan = berat TH
EN status_gizi = berlebihan
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 1
Masukkan rules anda : if umur = tua AND tinggi_badan = tinggi AND berat_badan = berat TH
EN status_gizi = berlebihan
Apakah anda ingin memasukkan rules lagi ?
jika iya ketikkan 1
jika tidak inputkan 0
input(1/0) : 0

```

In [40]: rule

```

Out[40]: ['umur = remaja AND tinggi_badan = pendek AND berat_badan = ringan THEN status_gizi = no
rmal',
'umur = remaja AND tinggi_badan = sedang AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = remaja AND tinggi_badan = tinggi AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = remaja AND tinggi_badan = pendek AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = remaja AND tinggi_badan = sedang AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = remaja AND tinggi_badan = tinggi AND berat_badan = normal THEN status_gizi = ku
rang',
'umur = remaja AND tinggi_badan = pendek AND berat_badan = berat THEN status_gizi = ber
lebihan',
'umur = remaja AND tinggi_badan = sedang AND berat_badan = berat THEN status_gizi = ber
lebihan',
'umur = remaja AND tinggi_badan = tinggi AND berat_badan = berat THEN status_gizi = nor
mal',
'umur = dewasa AND tinggi_badan = pendek AND berat_badan = ringan THEN status_gizi = no
rmal',
'umur = dewasa AND tinggi_badan = sedang AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = dewasa AND tinggi_badan = tinggi AND berat_badan = ringan THEN status_gizi = ku
rang',
'umur = dewasa AND tinggi_badan = pendek AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = dewasa AND tinggi_badan = sedang AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = dewasa AND tinggi_badan = tinggi AND berat_badan = normal THEN status_gizi = no
rmal',
'umur = dewasa AND tinggi_badan = pendek AND berat_badan = berat THEN status_gizi = ber
lebihan',
'umur = dewasa AND tinggi_badan = sedang AND berat_badan = berat THEN status_gizi = ber
lebihan',
'umur = dewasa AND tinggi_badan = tinggi AND berat_badan = berat THEN status_gizi = ber
lebihan',
'umur = tua AND tinggi_badan = pendek AND berat_badan = ringan THEN status_gizi = norma
l',
'umur = tua AND tinggi_badan = sedang AND berat_badan = ringan THEN status_gizi = kuran
g',
'umur = tua AND tinggi_badan = tinggi AND berat_badan = ringan THEN status_gizi = kuran

```

```

g',
'umur = tua AND tinggi_badan = pendek AND berat_badan = normal THEN status_gizi = norma
l',
'umur = tua AND tinggi_badan = sedang AND berat_badan = normal THEN status_gizi = norma
l',
'umur = tua AND tinggi_badan = tinggi AND berat_badan = normal THEN status_gizi = norma
l',
'umur = tua AND tinggi_badan = pendek AND berat_badan = berat THEN status_gizi = berleb
ihan',
'umur = tua AND tinggi_badan = sedang AND berat_badan = berat THEN status_gizi = berleb
ihan',
'umur = tua AND tinggi_badan = tinggi AND berat_badan = berat THEN status_gizi = berleb
ihan']

```

```

In [41]: peraturan_kiri = []
peraturan_kanan = []
keseluruhan_peraturan = []
peraturan_kanan_berlaku = []
peraturan_kiri_berlaku = []

for i in rule:
    x = i.split(" THEN ")
    peraturan_kiri.append(x[0])
    peraturan_kanan.append(x[1])

peraturan_kanan2 = []
for i in peraturan_kanan:
    x = i.split(" = ")
    peraturan_kanan2.append([x[0],x[1]])
    if [x[0],x[1]] in peraturan_kanan_berlaku:
        continue
    else:
        peraturan_kanan_berlaku.append([x[0],x[1]])

haha = []
for i in peraturan_kiri:
    x = i.split(" AND ")
    kata = []
    for j in x:
        kata += [j]
    haha.append(kata)

peraturan_kiri2 = []

for i in range(len(haha)):
    nilai = []
    for j in range(len(haha[i])):
        y = haha[i][j].split(" = ")
        duar = []
        for k in y:
            duar += [k]
        nilai.append(duar)
        if duar in peraturan_kiri_berlaku:
            continue
        else:
            peraturan_kiri_berlaku.append(duar)
    peraturan_kiri2.append(nilai)

for i in range(len(peraturan_kanan2)):
    kata = []
    for j in range(len(peraturan_kiri2[i])):
        kata += [peraturan_kiri2[i][j]]

```

```
kata.append(peraturan_kanan2[i])
keseluruhan_peraturan.append(kata)
```

```
In [42]: peraturan_kiri2
```

```
Out[42]: [['umur', 'remaja'], ['tinggi_badan', 'pendek'], ['berat_badan', 'ringan']],
[['umur', 'remaja'], ['tinggi_badan', 'sedang'], ['berat_badan', 'ringan']],
[['umur', 'remaja'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'ringan']],
[['umur', 'remaja'], ['tinggi_badan', 'pendek'], ['berat_badan', 'normal']],
[['umur', 'remaja'], ['tinggi_badan', 'sedang'], ['berat_badan', 'normal']],
[['umur', 'remaja'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'normal']],
[['umur', 'remaja'], ['tinggi_badan', 'pendek'], ['berat_badan', 'berat']],
[['umur', 'remaja'], ['tinggi_badan', 'sedang'], ['berat_badan', 'berat']],
[['umur', 'remaja'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'berat']],
[['umur', 'dewasa'], ['tinggi_badan', 'pendek'], ['berat_badan', 'ringan']],
[['umur', 'dewasa'], ['tinggi_badan', 'sedang'], ['berat_badan', 'ringan']],
[['umur', 'dewasa'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'ringan']],
[['umur', 'dewasa'], ['tinggi_badan', 'pendek'], ['berat_badan', 'normal']],
[['umur', 'dewasa'], ['tinggi_badan', 'sedang'], ['berat_badan', 'normal']],
[['umur', 'dewasa'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'normal']],
[['umur', 'dewasa'], ['tinggi_badan', 'pendek'], ['berat_badan', 'berat']],
[['umur', 'dewasa'], ['tinggi_badan', 'sedang'], ['berat_badan', 'berat']],
[['umur', 'dewasa'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'berat']],
[['umur', 'tua'], ['tinggi_badan', 'pendek'], ['berat_badan', 'ringan']],
[['umur', 'tua'], ['tinggi_badan', 'sedang'], ['berat_badan', 'ringan']],
[['umur', 'tua'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'ringan']],
[['umur', 'tua'], ['tinggi_badan', 'pendek'], ['berat_badan', 'normal']],
[['umur', 'tua'], ['tinggi_badan', 'sedang'], ['berat_badan', 'normal']],
[['umur', 'tua'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'normal']],
[['umur', 'tua'], ['tinggi_badan', 'pendek'], ['berat_badan', 'berat']],
[['umur', 'tua'], ['tinggi_badan', 'sedang'], ['berat_badan', 'berat']],
[['umur', 'tua'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'berat']]]
```

```
In [43]: peraturan_kanan2
```

```
Out[43]: ['status_gizi', 'normal'],
['status_gizi', 'kurang'],
['status_gizi', 'kurang'],
['status_gizi', 'normal'],
['status_gizi', 'normal'],
['status_gizi', 'kurang'],
['status_gizi', 'berlebihan'],
['status_gizi', 'berlebihan'],
['status_gizi', 'normal'],
['status_gizi', 'normal'],
['status_gizi', 'kurang'],
['status_gizi', 'kurang'],
['status_gizi', 'normal'],
['status_gizi', 'normal'],
['status_gizi', 'normal'],
['status_gizi', 'berlebihan'],
['status_gizi', 'berlebihan'],
['status_gizi', 'berlebihan'],
['status_gizi', 'normal'],
['status_gizi', 'kurang'],
['status_gizi', 'kurang'],
['status_gizi', 'normal'],
['status_gizi', 'normal'],
['status_gizi', 'normal'],
['status_gizi', 'berlebihan'],
['status_gizi', 'berlebihan'],
['status_gizi', 'berlebihan']]
```

```
In [44]: keseluruhan_peraturan
```

```
[[['umur', 'remaja'],
```

```
Out[44]: [['tinggi_badan', 'pendek'],
['berat_badan', 'ringan'],
['status_gizi', 'normal']],
[['umur', 'remaja'],
['tinggi_badan', 'sedang'],
['berat_badan', 'ringan'],
['status_gizi', 'kurang']],
[['umur', 'remaja'],
['tinggi_badan', 'tinggi'],
['berat_badan', 'ringan'],
['status_gizi', 'kurang']],
[['umur', 'remaja'],
['tinggi_badan', 'pendek'],
['berat_badan', 'normal'],
['status_gizi', 'normal']],
[['umur', 'remaja'],
['tinggi_badan', 'sedang'],
['berat_badan', 'normal'],
['status_gizi', 'normal']],
[['umur', 'remaja'],
['tinggi_badan', 'tinggi'],
['berat_badan', 'normal'],
['status_gizi', 'kurang']],
[['umur', 'remaja'],
['tinggi_badan', 'pendek'],
['berat_badan', 'berat'],
['status_gizi', 'berlebihan']],
[['umur', 'remaja'],
['tinggi_badan', 'sedang'],
['berat_badan', 'berat'],
['status_gizi', 'berlebihan']],
[['umur', 'remaja'],
['tinggi_badan', 'tinggi'],
['berat_badan', 'berat'],
['status_gizi', 'normal']],
[['umur', 'dewasa'],
['tinggi_badan', 'pendek'],
['berat_badan', 'ringan'],
['status_gizi', 'normal']],
[['umur', 'dewasa'],
['tinggi_badan', 'sedang'],
['berat_badan', 'ringan'],
['status_gizi', 'kurang']],
[['umur', 'dewasa'],
['tinggi_badan', 'tinggi'],
['berat_badan', 'ringan'],
['status_gizi', 'kurang']],
[['umur', 'dewasa'],
['tinggi_badan', 'pendek'],
['berat_badan', 'normal'],
['status_gizi', 'normal']],
[['umur', 'dewasa'],
['tinggi_badan', 'sedang'],
['berat_badan', 'normal'],
['status_gizi', 'normal']],
[['umur', 'dewasa'],
['tinggi_badan', 'tinggi'],
['berat_badan', 'normal'],
['status_gizi', 'normal']],
[['umur', 'dewasa'],
['tinggi_badan', 'pendek'],
['berat_badan', 'berat'],
['status_gizi', 'berlebihan']],
[['umur', 'dewasa'],
['tinggi_badan', 'sedang'],
['berat_badan', 'berat'],
```



```

['status_gizi', 'berlebihan']],
[['umur', 'dewasa'],
 ['tinggi_badan', 'tinggi'],
 ['berat_badan', 'berat'],
 ['status_gizi', 'berlebihan']],
[['umur', 'tua'],
 ['tinggi_badan', 'pendek'],
 ['berat_badan', 'ringan'],
 ['status_gizi', 'normal']],
[['umur', 'tua'],
 ['tinggi_badan', 'sedang'],
 ['berat_badan', 'ringan'],
 ['status_gizi', 'kurang']],
[['umur', 'tua'],
 ['tinggi_badan', 'tinggi'],
 ['berat_badan', 'ringan'],
 ['status_gizi', 'kurang']],
[['umur', 'tua'],
 ['tinggi_badan', 'pendek'],
 ['berat_badan', 'normal'],
 ['status_gizi', 'normal']],
[['umur', 'tua'],
 ['tinggi_badan', 'sedang'],
 ['berat_badan', 'normal'],
 ['status_gizi', 'normal']],
[['umur', 'tua'],
 ['tinggi_badan', 'tinggi'],
 ['berat_badan', 'normal'],
 ['status_gizi', 'normal']],
[['umur', 'tua'],
 ['tinggi_badan', 'pendek'],
 ['berat_badan', 'berat'],
 ['status_gizi', 'berlebihan']],
[['umur', 'tua'],
 ['tinggi_badan', 'sedang'],
 ['berat_badan', 'berat'],
 ['status_gizi', 'berlebihan']],
[['umur', 'tua'],
 ['tinggi_badan', 'tinggi'],
 ['berat_badan', 'berat'],
 ['status_gizi', 'berlebihan']]

```

```
In [45]: print(peraturan_kanan_berlaku)
```

```

[['status_gizi', 'normal'], ['status_gizi', 'kurang'], ['status_gizi', 'berlebihan']]

```

```
In [46]: print(peraturan_kiri_berlaku)
```

```

[['umur', 'remaja'], ['tinggi_badan', 'pendek'], ['berat_badan', 'ringan'], ['tinggi_badan', 'sedang'], ['tinggi_badan', 'tinggi'], ['berat_badan', 'normal'], ['berat_badan', 'berat'], ['umur', 'dewasa'], ['umur', 'tua']]

```

Melakukan Pencarian Hasil yang di inginkan

```
In [63]: pemasukan = []
def masukan():
    for i in var_input:
        nilai = float(input(f"nilai {i} : "))
        pemasukan.append([i, nilai])
```

```
In [64]: masukan()
```

```

nilai umur : 19
nilai tinggi_badan : 173
nilai berat_badan : 73

```

```
In [65]: pemasukan
```

```
Out[65]: [['umur', 19.0], ['tinggi_badan', 173.0], ['berat_badan', 73.0]]
```

Mencari fire strength atau perhitungan variabel input dengan rules yang ada

```
In [66]: pengecekan = []
fire_strength = []
def perhitunganaturanfuzzy():
    for i in range(len(pemasukan)):
        for j in range(len(domain_var_input)):
            if pemasukan[i][0] == domain_var_input[j][0] and float(pemasukan[i][1]) >= d:
                if [domain_var_input[j][0], domain_var_input[j][1]] in pengecekan:
                    continue
                else:
                    pengecekan.append([domain_var_input[j][0], domain_var_input[j][1]])

    for i in range(len(pengecekan)):
        for j in range(len(peraturan_kiri_berlaku)):
            if pengecekan[i] == peraturan_kiri_berlaku[j]:
                for l in range(len(pemasukan)):
                    for k in range(len(pemetaan_grafik_input)):
                        if pengecekan[i] == [pemetaan_grafik_input[k][0], pemetaan_grafik_input[k][2]] and pemetaan_grafik_input[k][2] == "trapesium":
                            x = float(pemasukan[l][1])
                            a = float(pemetaan_grafik_input[k][3][0][0])
                            b = float(pemetaan_grafik_input[k][3][0][1])
                            c = float(pemetaan_grafik_input[k][3][0][2])
                            d = float(pemetaan_grafik_input[k][3][0][3])
                            if a == b and b == x:
                                nilaiderajatkeanggotaan = 1
                            elif b <= x <= c:
                                nilaiderajatkeanggotaan = 1
                            elif d == c and c == x:
                                nilaiderajatkeanggotaan = 1
                            elif a == b and c < x < d:
                                if (d-c) == 0:
                                    nilaiderajatkeanggotaan = 0
                                else:
                                    nilaiderajatkeanggotaan = (d-x)/(d-c)
                            elif c == d and a < x < b:
                                if (b-a) == 0:
                                    nilaiderajatkeanggotaan = 0
                                else:
                                    nilaiderajatkeanggotaan = (x-a)/(b-a)
                            elif a == x:
                                nilaiderajatkeanggotaan = 0
                            else:
                                if (b-a) == 0 and (d-c) == 0:
                                    angka = 0
                                    nilaiderajatkeanggotaan = max(min(min(angka, 1), a), min(min(angka, 1), b))
                                elif (b-a) == 0:
                                    angka = 0
                                    nilaiderajatkeanggotaan = max(min(min(angka, 1), (d-x)/(d-c)), min(min(angka, 1), b))
                                elif (d-c) == 0:
                                    angka = 0
                                    nilaiderajatkeanggotaan = max(min(min((x-a)/(b-a), 1), a), min(min(angka, 1), b))
                                else:
                                    nilaiderajatkeanggotaan = max(min(min((x-a)/(b-a), 1), a), min(min(angka, 1), b))
                            if [[pengecekan[i][0], pengecekan[i][1]], nilaiderajatkeanggotaan] in fire_strength:
                                continue
                            else:
                                fire_strength.append([pengecekan[i][0], pengecekan[i][1], nilaiderajatkeanggotaan])
```

```

        else :
            fire_strength.append([[pengecekan[i][0], pengecekan[i]
elif pemetaan_grafik_input[k][2] == "segitiga":
    x = float(pemasukan[l][1])
    a = float(pemetaan_grafik_input[k][3][0][0])
    b = float(pemetaan_grafik_input[k][3][0][1])
    c = float(pemetaan_grafik_input[k][3][0][2])
    nilaiderajatkeanggotaan = max(min((x-a)/(b-a), (c-x)/(c-b
if [[pengecekan[i][0], pengecekan[i][1]], nilaiderajatkean
        continue
    else :
        fire_strength.append([[pengecekan[i][0], pengecekan[i]

```

In [67]: perhitunganaturanfuzzy()

In [68]: fire_strength

Out[68]:

```

[[['umur', 'dewasa'], 0.75],
 [['tinggi_badan', 'tinggi'], 0.5333333333333333],
 [['berat_badan', 'berat'], 1]]

```

Melakukan Implication dengan fungsi Min karena Operatornya AND pada RULES nya

In [69]:

```

fire_strength_hasil = []
def implikasi():
    for i in range(len(keseluruhan_peraturan)):
        for j in range(len(fire_strength)):
            for k in range(len(fire_strength)):
                for l in range(len(fire_strength)):
                    if [fire_strength[j][0], fire_strength[k][0], fire_strength[l][0]] ==
                        a = fire_strength[j][1]
                        b = fire_strength[k][1]
                        c = fire_strength[l][1]
                        nilai = min(a,b,c)
                        fire_strength_hasil.append([keseluruhan_peraturan[i][len(var_inp

```

In [70]: implikasi()

In [71]: fire_strength_hasil

Out[71]:

```

[[['status_gizi', 'berlebihan'], 0.5333333333333333]]

```

Melakukan Aggregation dengan fungsi MAX

In [72]:

```

fire_strength_hasil_akhir = []
hasil_yang_ada = []
def aggregation():
    for i in range(len(fire_strength_hasil)):
        if fire_strength_hasil[i][0] in hasil_yang_ada:
            continue
        else:
            hasil_yang_ada.append(fire_strength_hasil[i][0])
    for i in hasil_yang_ada:
        nilai = max([el for el in fire_strength_hasil if el[0] == i], key=lambda x: x[1])
        fire_strength_hasil_akhir.append(nilai)

```

In [73]: aggregation()

In [74]: fire_strength_hasil_akhir

Out[74]: [[['status_gizi', 'berlebihan'], 0.5333333333333333]]

Melakukan defuzzification menggunakan fungsi MOM (Mean of Maximum)

```
In [75]: pemetaan_grafik_hasil_akhir = []
atas = []
bawah = []
def defuzzification_mamdani():
    print("Disini Menggunakan metode MOM (Mean of Maximum)")
    for long in range(len(fire_strength_hasil_akhir)):
        for i in range(len(pemetaan_grafik_output)):
            if fire_strength_hasil_akhir[long][0][0] and fire_strength_hasil_akhir[long]
                if 'trapesium' in pemetaan_grafik_output[i]:
                    a = pemetaan_grafik_output[i][3][0][0]
                    b = pemetaan_grafik_output[i][3][0][1]
                    c = pemetaan_grafik_output[i][3][0][2]
                    d = pemetaan_grafik_output[i][3][0][3]
                    puncak = fire_strength_hasil_akhir[long][1]
                    if a == b:
                        titik_y_fire_strength = d - (puncak*(d-c))
                        y = (titik_y_fire_strength + a) / 2
                        atas.append(y)
                        bawah.append(puncak)
                        pemetaan_grafik_hasil_akhir.append([fire_strength_hasil_akhir[lo
                    elif d == c:
                        titik_y_fire_strength = a + (puncak*(b-a))
                        y = (titik_y_fire_strength + d) / 2
                        atas.append(y)
                        bawah.append(puncak)
                        pemetaan_grafik_hasil_akhir.append([fire_strength_hasil_akhir[lo
                    else:
                        titik_y_fire_strength1 = a + (puncak*(b-a))
                        titik_y_fire_strength2 = d - (puncak*(d-c))
                        y = (titik_y_fire_strength1 + titik_y_fire_strength2) / 2
                        atas.append(y)
                        bawah.append(puncak)
                        pemetaan_grafik_hasil_akhir.append([fire_strength_hasil_akhir[lo

                elif 'segitiga' in pemetaan_grafik_output[i]:
                    a = pemetaan_grafik_output[i][3][0][0]
                    b = pemetaan_grafik_output[i][3][0][1]
                    c = pemetaan_grafik_output[i][3][0][2]
                    y = b
                    titik_y_fire_strength1 = a + (puncak*(b-a))
                    titik_y_fire_strength2 = c - (puncak*(c-b))
                    atas.append(y)
                    bawah.append(puncak)
                    pemetaan_grafik_hasil_akhir.append([fire_strength_hasil_akhir[long][

        y = atas[bawah.index(max(bawah))]
        print(fire_strength_hasil_akhir[long][0][0], " dari ", pemasukan, " adalah ", y)
```

```
In [76]: defuzzification_mamdani()

Disini Menggunakan metode MOM (Mean of Maximum)
status_gizi dari [['umur', 19.0], ['tinggi_badan', 173.0], ['berat_badan', 73.0]] ada
lah 32.566666666666666
```

Dibawah ini hanya gambaran dari perhitungannya

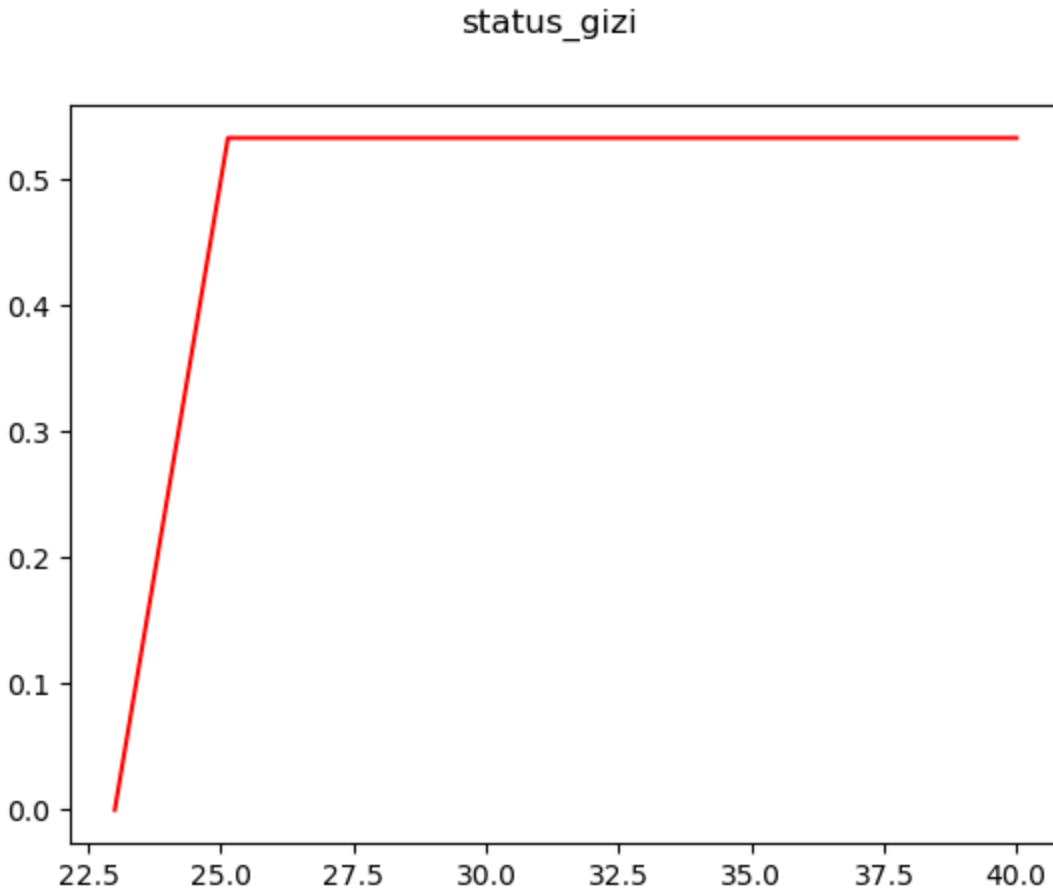
```
In [77]: def gambar_output_akhir():
```

```

for i in range(len(var_output)):
    for j in range(len(pemetaan_grafik_hasil_akhir)):
        if pemetaan_grafik_hasil_akhir[j][0] == var_output[i]:
            plt.plot(pemetaan_grafik_hasil_akhir[j][3][0], pemetaan_grafik_hasil_akhir[j][3][1])
            plt.suptitle(pemetaan_grafik_hasil_akhir[j][0])
plt.show()

```

In [78]: gambar_output_akhir()



berikut hasil hasil percobaan

```

In [79]: pemasukan = []
masukkan()
pengecekan = []
fire_strength = []
perhitunganaturanfuzzy()
fire_strength_hasil = []
implikasi()
fire_strength_hasil_akhir = []
hasil_yang_ada = []
aggregation()
pemetaan_grafik_hasil_akhir = []
atas = []
bawah = []
defuzzification_mamdani()

nilai umur : 21
nilai tinggi_badan : 181
nilai berat_badan : 80
Disini Menggunakan metode MOM (Mean of Maximum)
status_gizi dari [['umur', 21.0], ['tinggi_badan', 181.0], ['berat_badan', 80.0]] ada
lah 33.5

```

In [80]: pemasukan = []

```

masukkan()
pengecekan = []
fire_strength = []
perhitunganaturanfuzzy()
fire_strength_hasil = []
implikasi()
fire_strength_hasil_akhir = []
hasil_yang_ada = []
aggregation()
pemetaan_grafik_hasil_akhir = []
atas = []
bawah = []
defuzzification_mamdani()

```

```

nilai umur : 20
nilai tinggi_badan : 155
nilai berat_badan : 60
Disini Menggunakan metode MOM (Mean of Maximum)
status_gizi dari [['umur', 20.0], ['tinggi_badan', 155.0], ['berat_badan', 60.0]] ada
lah 32.388888888888886

```

```

In [81]: pemasukan = []
masukkan()
pengecekan = []
fire_strength = []
perhitunganaturanfuzzy()
fire_strength_hasil = []
implikasi()
fire_strength_hasil_akhir = []
hasil_yang_ada = []
aggregation()
pemetaan_grafik_hasil_akhir = []
atas = []
bawah = []
defuzzification_mamdani()

```

```

nilai umur : 14
nilai tinggi_badan : 154
nilai berat_badan : 56
Disini Menggunakan metode MOM (Mean of Maximum)
status_gizi dari [['umur', 14.0], ['tinggi_badan', 154.0], ['berat_badan', 56.0]] ada
lah 20.6

```

```

In [82]: pemasukan = []
masukkan()
pengecekan = []
fire_strength = []
perhitunganaturanfuzzy()
fire_strength_hasil = []
implikasi()
fire_strength_hasil_akhir = []
hasil_yang_ada = []
aggregation()
pemetaan_grafik_hasil_akhir = []
atas = []
bawah = []
defuzzification_mamdani()

```

```

nilai umur : 17
nilai tinggi_badan : 154
nilai berat_badan : 56
Disini Menggunakan metode MOM (Mean of Maximum)
status_gizi dari [['umur', 17.0], ['tinggi_badan', 154.0], ['berat_badan', 56.0]] ada
lah 20.833333333333336

```

```

In [83]: pemasukan = []

```

```
masukkan()
pengecekan = []
fire_strength = []
perhitunganaturanfuzzy()
fire_strength_hasil = []
implikasi()
fire_strength_hasil_akhir = []
hasil_yang_ada = []
aggregation()
pemetaan_grafik_hasil_akhir = []
atas = []
bawah = []
defuzzification_mamdani()
```

```
nilai umur : 30
nilai tinggi_badan : 154
nilai berat_badan : 66
Disini Menggunakan metode MOM (Mean of Maximum)
status_gizi dari [['umur', 30.0], ['tinggi_badan', 154.0], ['berat_badan', 66.0]] ada
lah 32.833333333333336
```

In []: