**JOBSHEET 2**

**REASONING FUZZY (1)**

# Tujuan

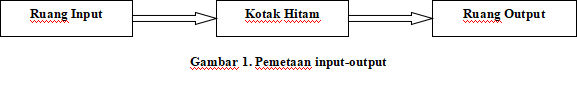
Mahasiswa diharapkan dapat:

* Mahasiswa mengenal Reasoning Fuzzy bagian pertama
* Mahasiswa dapat mengetahui dan mengenal contoh studi kasus yang dapat diselesaikan dengan metode Reasoning Fuzzy bagian pertama
* Mahasiswa mampu menerapkan metode Reasoning Fuzzy bagian pertama menggunakan bahas Python

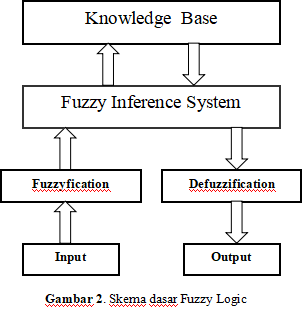
# Ringkasan Materi

Logika Fuzzy pertama kali dikenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965, melalui tulisannya yang berjudul “Fuzzy Set”, membahas mengenai teori himpunan samar / fuzzy.

Sebagai contoh, untuk menyatakan air itu panas atau dingin, amat bersifat relative. Logika fuzzy merupakan suatu cara yang tepat untuk memetakan suatu ruang input ke dalam ruang output. Dalam gambar 1, kotak hitam menyatakan proses yang dilakukan terhadap input supaya menghasilkan output.



Pada tahun 1965, Zadeh memodifikasi teori himpunan dimana setiap anggotanya memiliki derajat keanggotaan kontinu antara 0 - 1, himpunan ini disebut himpunan samar (*Fuzzy set*). Sebagai contoh: himpunan temperatur yang akan mempengaruhi kondisi panas tidak bersifat diskrit dan dibatasi kondisi hangat. Skema dasar dari *fuzzy logic* dapat dilihat pada Gambar 2.



Input fuzzy berupa bilangan crisp (tegas) yang dinyatakan dalam himpunan input. Fuzzifikasi merupakan proses untuk mengubah bilangan crisp menjadi nilai keanggotaan dalam himpunan fuzzy. *Fuzzy inference system* merupakan bagian pengambilan kesimpulan (*reasoning*) dan keputusan. *Knowledge base* berisi aturan-aturan yang biasanya dinyatakan dengan perintah IF …. THEN….

Sedangkan Defuzzification, merupakan proses untuk merubah nilai output fuzzy menjadi nilai crisp.

## 1. Himpunan Crisp dan Himpunan Fuzzy

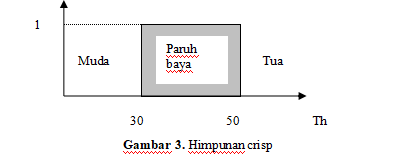
Jika diberikan suatu himpunan semesta pembicaraan, himpunan crisp “A” didefinisikan oleh itemitem yang ada pada himpunan itu.

Jika **a Є A**, maka nilai yang berhubungan dengan **a** adalah **1**, namun jika **a ¢ A**, maka nilai yang berhubungan dengan **a** adalah **0**.

Sebagai contoh misalnya kita membuat klasifikasi himpunan umur dengan kategori yang tegas :

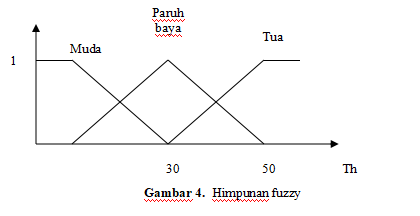
|  |  |
| --- | --- |
| Muda | <30 tahun |
| Paruh baya | 35-50 tahun |
| Tua | >50 tahun |

Sebagai contoh, himpunan crisp untuk kategori paruh baya yang dibentuk adalah:



Dari contoh di atas terdapat batas yang tegas untuk tiap kategori umur. Tetapi hal ini bisa menimbulkan kontradiktif jika kita mengambil contoh umur di sekitar batas kategori. Misalkan kita ambil contoh umur 50 tahun masuk kategori tua sedangkan umur 50 tahun kurang 1 bulan masuk kategori paruh baya. Hal ini kurang logis sehingga diperlukan suatu himpunan yang bisa mengatasi hal kontradiktif tersebut.

Dengan adanya himpunan fuzzy, hal kontradiktif tersebut bisa diatasi karena batas tiap kategori bisa saling *overlapping*.



### 2. Logika Fuzzy

Fungsi keanggotaan (µf = *membership function*) merupakan suatu kurva yang menunjukkan pemetaan antara titik-titik input data ke dalam nilai keanggotaannya (derajat keanggotaan).

Ambil U sebagai semesta dari himpunan objek { u }. Himpunan fuzzy F dalam semesta pembicaraa U dinyatakan dalam nilai keanggotaan µf yang mempunyai interfal nilai .

Himpunan fuzzy biasanya dinyatakan dengan :



jika U kontinyu maka himpunan F dapat ditulis dengan :



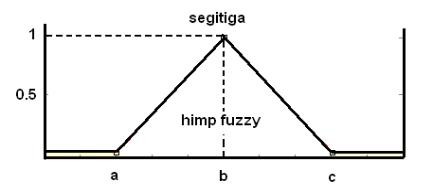
jika U diskrit maka himpunan F dapat ditulis dengan :



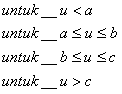
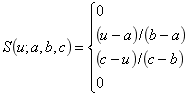
## 3. Fungsi Nilai Keanggotaan

Fungsi keanggotaan fuzzy ada beberapa macam antara lain :

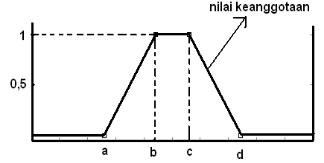
1. fungsi keanggotaan segitiga



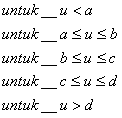
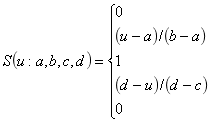
**Gambar 5**. Himpunan fungsi keanggotaan Segitiga



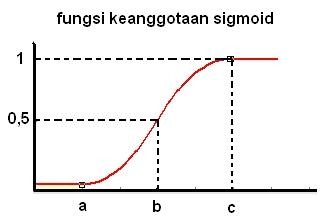
1. fungsi keanggotaan trapesium



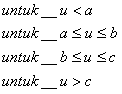
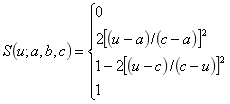
**Gambar 6**. Himpunan fungsi keanggotaan Trapesium



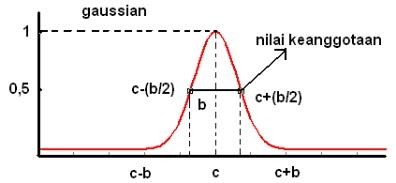
1. Fungsi keanggotaan sigmoid



**Gambar 7**. Himpunan fungsi keanggotaan Sigmoid



1. fungsi keanggotaan gausian



**Gambar 8**. Himpunan fungsi keanggotaan Gaussian



### 4. Fuzzy Reasoning

Ada 2 cara yang sering dipakai dalam pengambilan kesimpulan (*reasoning*) yaitu generalisasi modus ponen (GMP) dan generalisasi modus tolen (GMT). Modus ponen sering disebut penyimpulan langsung (*direct reasoning*) sedangkan modus tolen sering disebut penyimpulan tidak langsung (*indirect reasoning*).

Misalkan suatu himpunan fuzzy dinyatakan dengan A, A’, B, B’ dan variabel linguistic dinyatakan dengan X dan Y, maka ekspresi dari modus ponen dan modus tolen adalah :

**Modus ponen :**

Premise 1 (pengetahuan) : jika X adalah A maka Y adalah B

Premise 2 (fakta) : X adalah A’ .

Kesimpulan : Y adalah B’

Kesimpulan B’ dapat dinyatakan oleh : B’ = A’ o R

Dimana :

R = hubungan implikasi fuzzy dari “jika A maka B”

o = operator komposisi

A’ = himpunan fuzzy yang bisa berbentuk : A, sangat A, sedikit A, agak A, not A dll.

**Modus Tolen :**

Premise 1 (pengetahuan) : jika X adalah A maka Y adalah B

Premise 2 (fakta) : Y adalah B’ .

Kesimpulan : X adalah A’

Biasanya pengambilan kesimpulan dengan metode ini hanya valid jika B’ = not B dan A’ = not A sehingga nilai A’ dapat dinyatakan dengan : A’ = R o B’

### 5. Sistem Inferensi Fuzzy

Pada umumnya tiap-tiap aturan (proposisi) fuzzy dinyatakan dalam bentuk IF..THEN.. dan menyatakan suatu hubungan tertentu. Hubungan fuzzy ini sering disebut implikasi. Hubungan fuzzy dalam *knowledge base* dapat didefinisikan sebagai himpunan implikasi fuzzy.

Ada 2 jenis proposisi fuzzy yaitu “*condition fuzzy proposition*’ dan ‘*uncondition fuzzy proposition*’.

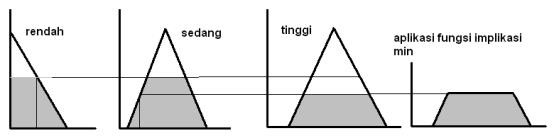
1. *Condotion Fuzzy Proposition*

Jenis ini dicirikan dengan penggunaan IF.

IF x is A THEN y is B

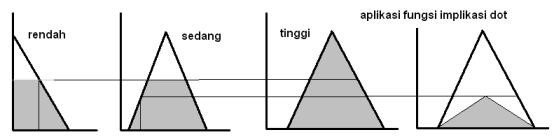
Jika suatu proposisi menggunakan bentuk terkondisi maka ada 2 fungsi implikasi yang dapat digunakan yaitu :

* 1. Minimum : fungsi ini memotong output himpunan fuzzy.



**Gambar 9**. Fungsi Implikasi min

* 1. Dot : fungsi ini menskalakan output himpunan fuzzy



**Gambar 10**. Fungsi Implikasi dot

1. *Uncondition Fuzzy Proposition.*

Jenis *uncondition* ditandai dengan tidak adanya pernyataan IF.

x is A

proposisi *uncondition* selalu diaplikasi dengan model AND.

Jika dalam system fuzzy terdapat beberapa aturan, maka ada 3 metode yang dipakai dalam menentukan inferensi yaitu : *max-min*, *additive* dan *probabilistic* *OR (probor)*

1. metode *Max-Min*

Max dapat dianalogikan dengan operasi logika OR sedangkan Min dianalogikan dengan operasi logika AND.

1. metode *additive* metode *additive* dilakukan dengan melakukan *bounded-sum* terhadap semua output daerah fuzzy. Secara umum dapat dituliskan dengan L:

µsf[xi] ß min (1, µsf[xi] +µkf[xi] ) µsf[xi] = nilai keanggotaan solusi fuzzy sampai aturan ke-i

µkf[xi] = nilai keanggotaan konsekuen (output) fuzzy sampai aturan ke-i

1. metode Probor metode probor diperoleh dengan melakukan *product* (perkalian) terhadap semua output daerah fuzzy. Secara umum ditulikan dengan :

µsf[xi] ß ( µsf[xi] +µkf[xi] ) - ß ( µsf[xi] \*µkf[xi] ) µsf[xi] = nilai keanggotaan solusi fuzzy sampai aturan ke-i µkf[xi] = nilai keanggotaan konsekuen (output) fuzzy sampai aturan ke-i

### 6. Basis Aturan

Basis aturan terdiri dari sejumlah aturan yang biasanya dinyatakan secara *linguistic*. Aturan fuzzy seringkali dinyatakan dengan “IF…THEN….”. Hal ini didasarkan pada kebutuhan akan :

1. memfasilitasi kepakaran (*expert*) manusia dengan cara yang tepat untuk mengekspresikan pengetahuan dan pengalaman mereka.
2. memfasilitasi desainer dengan cara yang mudah untuk membuat dan memprogram aturan fuzzy.
3. mengurangi biaya desain dan
4. meningkatkan efisiensi pengambilan keputusan fuzzy (*fuzzy inference*).

Jika variabel input lebih dari satu, biasanya ada penghubung yang menyatakan relasi dari tiap-tiap input. Penghubung itu biasanya dinyatakan dengan ‘AND‘, ‘OR’, dan ‘ALSO’. Pemakaian penghubung itu misalnya ;

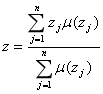
IF x1  IS Ak1 OR x2 IS Ak2 THEN y1 IS Bk1 ALSO y2 is Bk2 secara lebih singkat dapat dinyatakan

IF Ak1 OR Ak2 THEN Bk1 ALSO Bk

### 7. Defuzzifikasi

Terdapat beberapa metode deffuzifikasi sebagai berikut :

1. metode centroid metode centroid dilakukan dengan mengambil titik pusat daerah fuzzy. Secara umum dirumuskan dengan :

 atau 

1. metode bisector metode bisector dilakukan dengan mengambil nilai dari domain fuzzy yang memiliki nilai keanggotaan separo dari nilai keanggotaan fuzzy. Secara dirumuskan dengan :



1. metode Mean of Maximum (MOM) metode MOM dilakukan dengan mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum
2. metode Largest of Minimum (LOM) metode LOM dilakukan dengan mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.
3. metode Smallest of Maximum (SOM)] metode LOM dilakukan dengan mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum

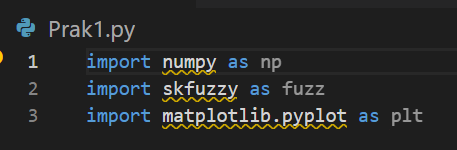
# Praktikum

Uji coba praktikum ini terdiri dari 3 bagian yang saling berhubungan antara satu dengan yang lain. Bagian 1 menjelaskan. Bagian 2 menjelaskan. Bagian 3 menjelaskan.

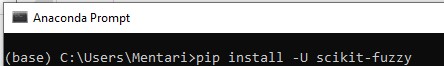
## 1. Percobaan 1 : Fungsi Keanggotaan Variabel Input dan Output (Fuzzyfikasi)

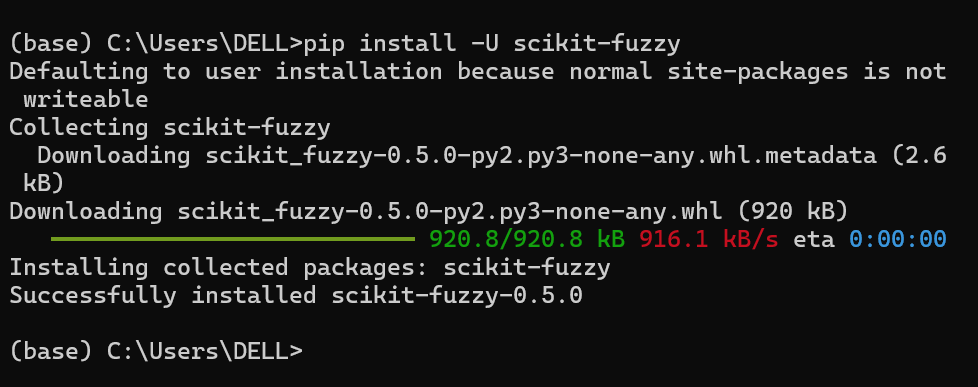
1. Terdapat dua buah variable input untuk studi kasus yang diambil pada jobsheet kali ini. Variabel input berupa kualitas makanan, dan kualitas pelayanan makanan akan menentukan output berupa jumlah tip yang diberikan dalam poin prosentase. Fungsi keanggotaan untuk studi kasus berikut akan dibagi menjadi tinggi, sedang, dan rendah. Untuk dapat membuat program fuzzy dengan Python diperlukan beberapa import library sebagai berikut :





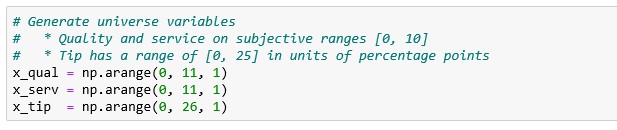
1. Saat percobaan 1 bagian satu di atas di jalankan, apakah terdapat pesan error yang mejelaskan tentang salah satu modul yang di impor tidak dapat ditemukan?Jika iya, maka harus dilakukan instalasi terlebih dahulu. Buka anaconda prompt, kemudian ketikkan kode berikut :



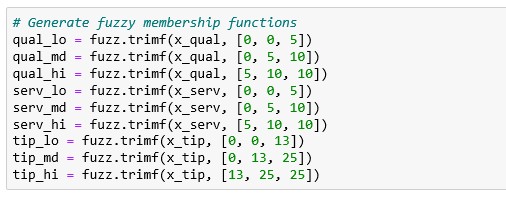


Tunggu beberapa saaat sampai modul tersebut berhasil di download dan di install dengan benar. Setelah itu kembali jalankan kode program pada percobaan 1 bagian satu.

1. Tahap selanjutnya generate variable global. Untuk variable input kualitas makanan dan kualitas pelayanan berada pada rentang subjektif antara 0 sampai 10. Sedangkan untuk variable output yaitu tip berada pada rentang subjektif antara 0 sampai 25. Sesuai dengan kode berikut



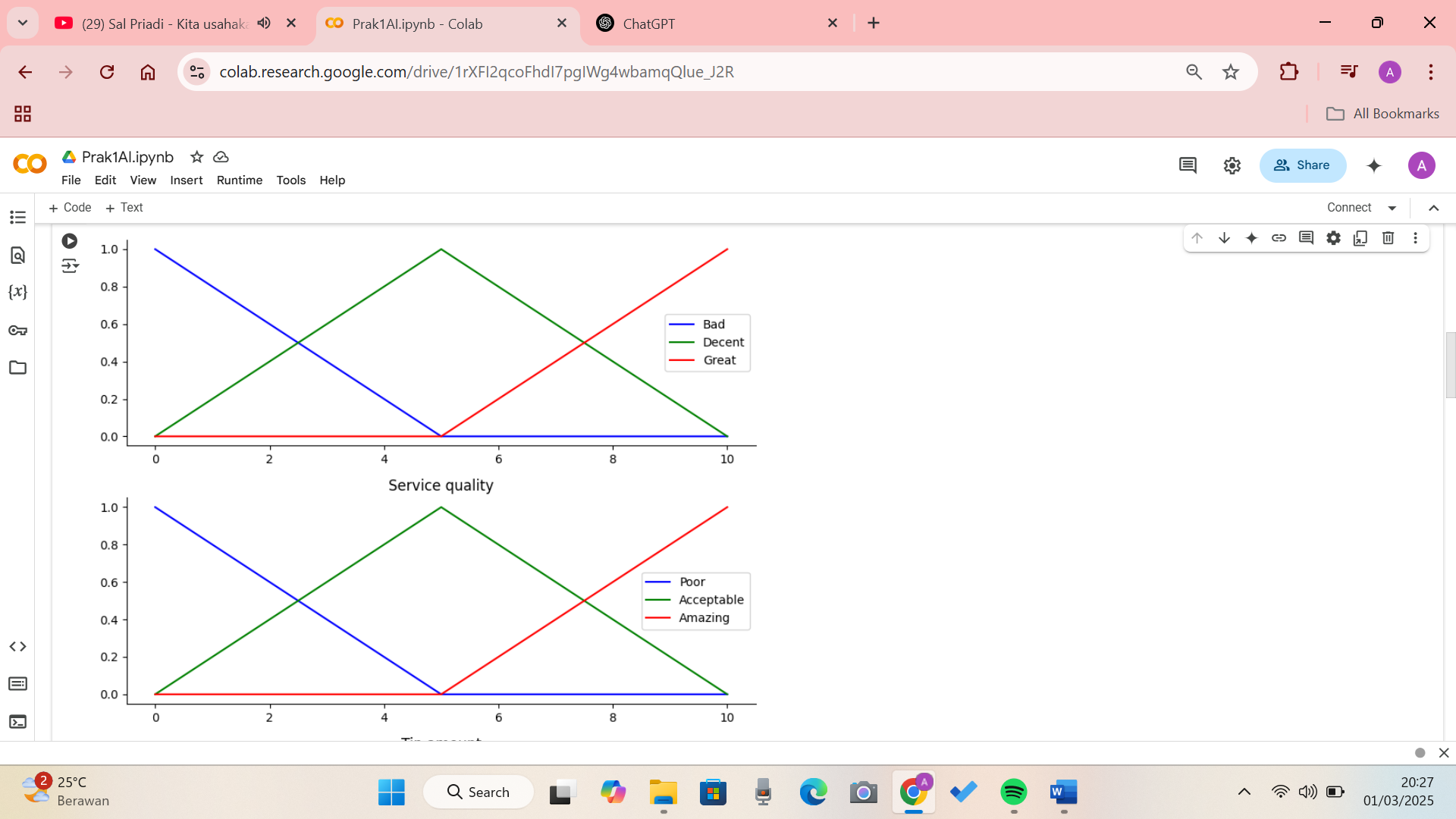
1. Kemudian sampai pada tahap generate membership function untuk ketiga variable yang ada pada rentang tinggi, sedang dan rendah.

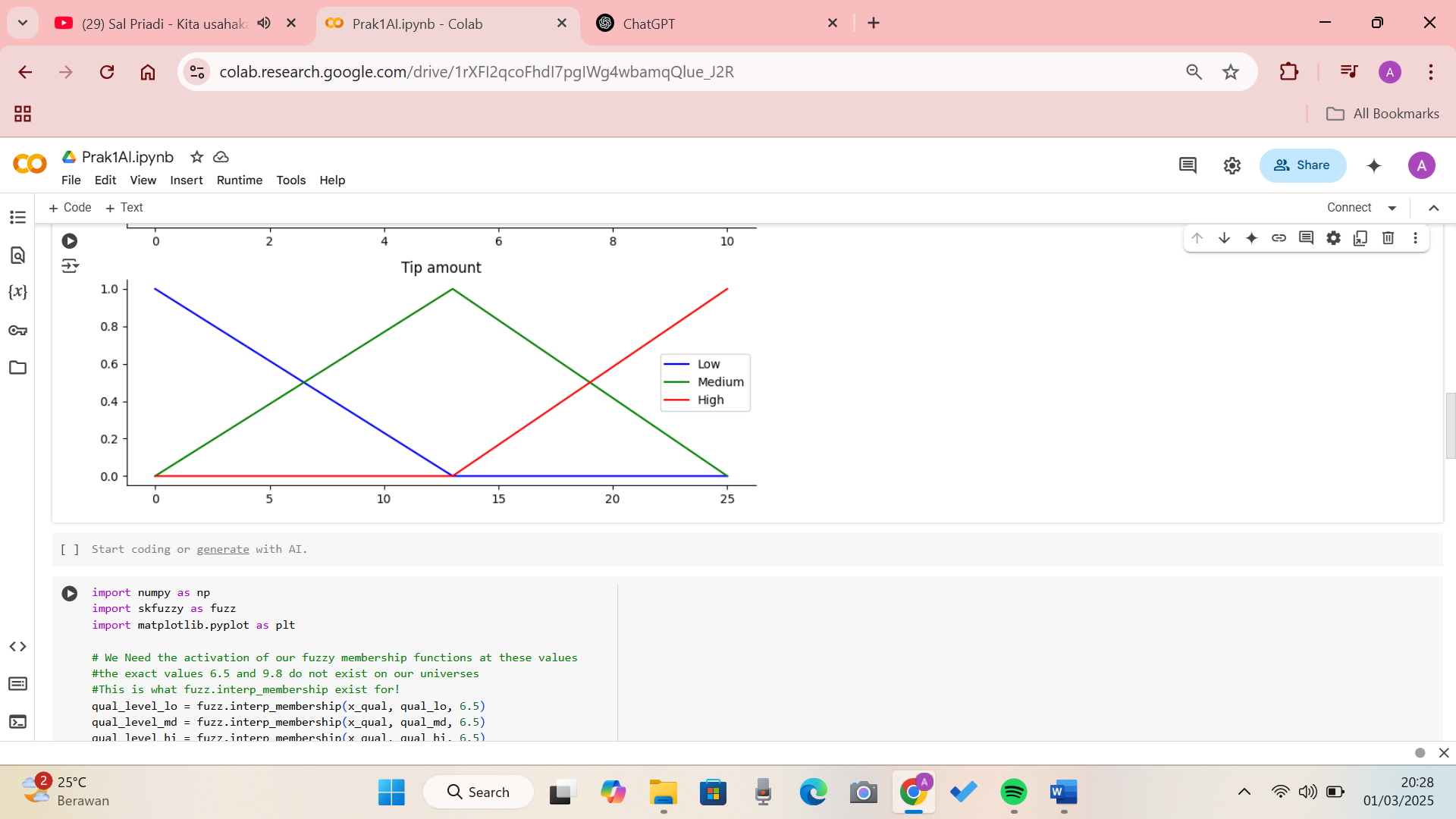


1. Tahap terakhir adalah tahap visualisasi dari membership function untuk tiap-tiap variabel : kualitas makanan, kualitas makanan, dan tip.



**Hasil kode:**





### Pertanyaan

1. Apakah kegunaan import numpy pada kode program di atas?

**Jawab:** import numpy as np adalah perintah dalam Python untuk mengimpor library **NumPy yang pada program diatas library tersebut digunakan untuk :**

* Membuat array rentang nilai **variabel semesta**
* Mempermudah perhitungan fungsi keanggotaan fuzzy

1. Apakah kegunaan import skfuzzy pada code program di atas?

**Jawab:** Untuk mengimport library skfuzzy yang digunakan untuk mengimplementasikan sistem logika fuzzy, terutama dalam pembuatan fungsi keanggotaan fuzzy yang digunakan untuk menentukan derajat keanggotaan suatu variabel dalam sistem fuzzy.

1. Apakah fungsi import matplotlib.pyplot pada code program di atas?

**Jawab:** untuk mengimpor pustaka visualisasi data Matplotlib dan modul Pyplot yang digunakan untuk membuat dan menampilkan grafik fungsi keanggotaan fuzzy, sehingga dapat divisualisasikan dengan lebih jelas.

1. Jelaskan dengan detil tentang kode program mengenai visualisasi pada percobaan satu di atas!

**Jawab:**

x\_qual = np.arange(0, 11, 1)

x\_serv = np.arange(0, 11, 1)

x\_tip  = np.arange(0, 26, 1)

kode di atas dibuat untuk mendefinisikan variable yang akan digunakan untuk memvisualisasikan data dimana:

* x\_qual → Variabel kualitas makanan dengan rentang nilai 0 hingga 10.
* x\_serv → Variabel pelayanan dengan rentang nilai 0 hingga 10.
* x\_tip → Variabel tip yang diberikan dengan rentang nilai 0 hingga 25.
* np.arange(start, stop, step) digunakan untuk membuat array dengan nilai dari start ke stop dengan selisih 1.

#Generate fuzzy membership functions

qual\_lo = fuzz.trimf(x\_qual, [0, 0, 5])

qual\_md = fuzz.trimf(x\_qual, [0, 5, 10])

qual\_hi = fuzz.trimf(x\_qual, [5, 10, 10])

serv\_lo = fuzz.trimf(x\_serv, [0, 0, 5])

serv\_md = fuzz.trimf(x\_serv, [0, 5, 10])

serv\_hi = fuzz.trimf(x\_serv, [5, 10, 10])

tip\_lo = fuzz.trimf(x\_tip, [0, 0, 13])

tip\_md = fuzz.trimf(x\_tip, [0, 13, 25])

tip\_hi= fuzz.trimf(x\_tip, [13, 25, 25])

kode diatas memiliki fungsi:

* qual\_lo → Keanggotaan rendah (Bad), Segitiga dengan titik (0, 0, 5), Nilai 0 ke 5 meningkat, setelah itu nol.
* qual\_md → Keanggotaan sedang (Decent), Segitiga dengan titik (0, 5, 10), Nilai 0 ke 5 naik, lalu ke 10 turun.
* qual\_hi → Keanggotaan tinggi (Great), Segitiga dengan titik (5, 10, 10), Nilai 5 ke 10 naik, setelah itu tetap.
* fuzz.trimf(x, [a, b, c]) digunakan untuk membuat fungsi keanggotaan segitiga dengan titik a (awal), b (tengah), c (akhir).

Dan kode yag sama digunakan untuk membuat keanggotaan fuzzy dari jumlah tip dan pelayanan yang di berikan

#Visualize these universes and membership functions

fig, (ax0, ax1, ax2) = plt.subplots(nrows=3, figsize=(8,9))

kode di atas digunakan untuk :

* plt.subplots(nrows=3, figsize=(8,9)) → Membuat 3 subplot (baris) dengan ukuran 8x9 inci.
* ax0, ax1, ax2 → Masing-masing subplot akan menampilkan grafik Quality, Service, dan Tip.

ax0.plot(x\_qual, qual\_lo, 'b', linewidth=1.5, label='Bad')

ax0.plot(x\_qual, qual\_md, 'g', linewidth=1.5, label='Decent')

ax0.plot(x\_qual, qual\_hi, 'r', linewidth=1.5, label='Great')

ax0.legend()

ax1.plot(x\_serv, serv\_lo, 'b', linewidth=1.5, label='Poor')

ax1.plot(x\_serv, serv\_md, 'g', linewidth=1.5, label='Acceptable')

ax1.plot(x\_serv, serv\_hi, 'r', linewidth=1.5, label='Amazing')

ax1.set\_title('Service quality')

ax1.legend()

ax2.plot(x\_tip, tip\_lo, 'b', linewidth=1.5, label='Low')

ax2.plot(x\_tip, tip\_md, 'g', linewidth=1.5, label='Medium')

ax2.plot(x\_tip, tip\_hi, 'r', linewidth=1.5, label='High')

ax2.set\_title('Tip amount')

ax2.legend()

kode di atas digunakan untuk;

* ax0.plot(x, y, 'color', linewidth, label) menggambar fungsi keanggotaan untuk kualitas makanan.
* ax0.legend() menampilkan legenda (Bad, Decent, Great).
* set\_title('Title') digunakan untuk memberikan judul pada subplot.

# Turn off top/right axes

for ax in (ax0, ax1, ax2):

    ax.spines['top'].set\_visible(False)

    ax.spines['right'].set\_visible(False)

    ax.get\_xaxis().tick\_bottom()

    ax.get\_yaxis().tick\_left()

    plt.tight\_layout()

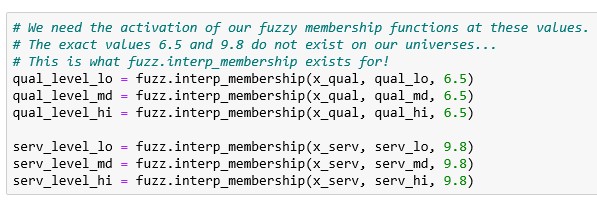
* ax0, ax1, ax2 adalah tiga subplot yang sudah dibuat sebelumnya.
* Loop for ax in (ax0, ax1, ax2) berarti akan mengulangi semua perintah dalam blok ini untuk setiap subplot.
* ax.spines mengacu pada bingkai (border) dari subplot.
* ax.spines['top'] mengontrol garis tepi atas.
* ax.spines['right'] mengontrol garis tepi kanan.
* .set\_visible(False) digunakan untuk menyembunyikan garis tersebut.
* ax.get\_xaxis() → Mengakses sumbu X dari subplot.
* .tick\_bottom() → Memastikan tick (angka skala) hanya muncul di bagian bawah grafik.
* ax.get\_yaxis() → Mengakses sumbu Y dari subplot.
* .tick\_left() → Memastikan tick (angka skala) hanya muncul di sebelah kiri grafik.
* plt.tight\_layout() → Mengatur tata letak agar subplot tidak bertumpuk.
* plt.show() → Menampilkan grafik ke layar.

## 2. Percobaan 2 : Pengujian Fuzzy Rule

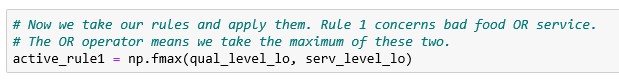
1. Terdapat beberapa rule dalam implementasi fuzzy studi kasus contoh jobsheet ke-5 ini. Rule tersebut adalah :
   1. Jika kualitas makanan buruk ATAU kualitas pelayanan buruk, maka tip yang diberikan akan rendah
   2. Jika kualitas pelayanan dapat diterima, maka tip yang diberikan berada pada tingkat menengah
   3. Jika kualitas makanan enak ATAU kualitas pelayanan luar biasa, maka tip yang diberikan akan tinggi.

Rule yang sudah dijelaskan di atas merupakan rule yang masih samar (fuzzy). Maka dari itu butuh ditindak lanjuti menggunakan logika Fuzzy.

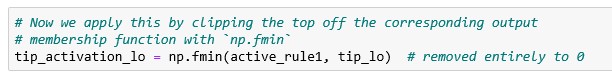
Pada percobaan ke-2 kali ini akan dilakukan pengujian aturan Fuzzy dengan memberikan nilai 6.5 untuk kualitas makanan dan 9.8 untuk kualitas pelayanan. Sesuai dengan kode program di bawah ini.



1. Dengan adanya input kualitas makanan dan kualitas pelayanan pada percobaan 2 tahap 1 di atas, maka selajutnya akan dilakukan penerapan sesuai dengan aturan/rule yang ada, sebagai berikut :



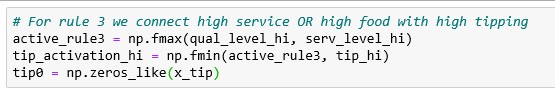
1. Selanjutnya akan dilakukan clipping yang disesuaikan dengan output menggunakan fungsi keanggotaan “np.fmin”



1. Di bawah ini berisi tentang kode untuk rule ke-2 yang menghubungkan kualitas pelayanan dengan pemberian tip tingkat menengah.

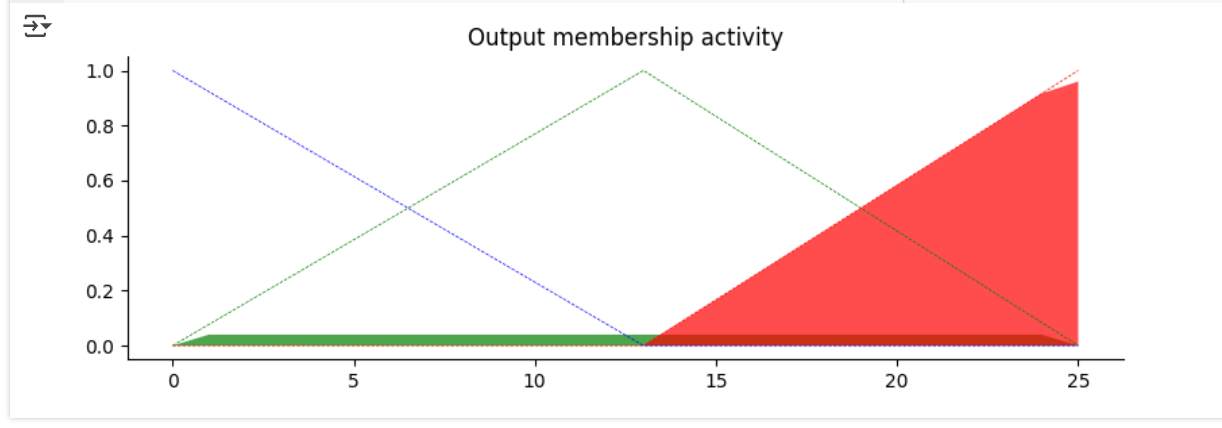


1. Untuk rule ke-3 di implementasikan dengan kode program di bawah ini:



1. Tahap terakhir pada percobaan ke-2 ini adalah membentuk visualisasi input yang telah diisi, dan telah disesuaikan dengan rule yang ada mulai dari rule no 1 sampai dengan rule no 3.



Hasil kode program

### Pertanyaan

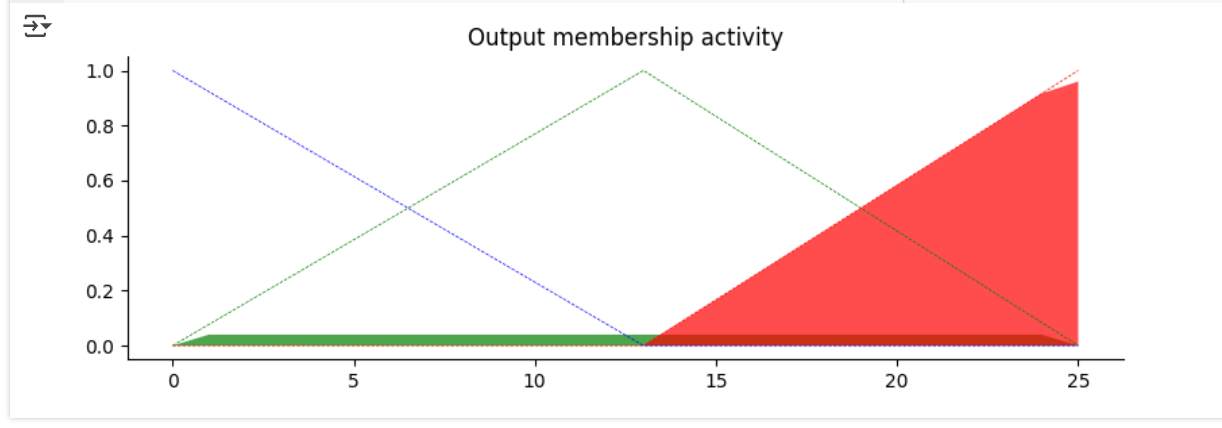
1. Apakah kegunaan clipping pada percobaan 2 tahap 3?

Jawab:

* Membatasi fungsi keanggotaan tip rendah (tip\_lo) agar tidak melebihi aturan yang aktif.
* Mencegah output fuzzy memiliki keanggotaan yang tidak sesuai dengan aturan fuzzy.
* Menyiapkan output untuk langkah defuzzifikasi yang lebih akurat. Dimana:
* active\_rule1 adalah batas maksimum yang diperbolehkan untuk fungsi keanggotaan tip rendah (tip\_lo).
* np.fmin() memilih nilai minimum antara active\_rule1 dan setiap titik pada tip\_lo.
* Jika active\_rule1 memiliki nilai lebih rendah dari tip\_lo, maka tip\_lo akan dipotong (clipping) hingga setinggi active\_rule1.

1. Jelaskan tentang visualisasi yang didapat dari percobaan 2 tahap akhir!

Jawab:



* Warna Biru (Tip Rendah - tip\_activation\_lo)
* Warna Hijau (Tip Sedang - tip\_activation\_md)
* Warna Merah (Tip Tinggi - tip\_activation\_hi)
* Garis Putus-Putus menunjukkan fungsi keanggotaan asli sebelum clipping.
* Jika makanan atau layanan buruk, area tip rendah (biru) lebih dominan.
* Jika layanan cukup baik, area tip sedang (hijau) terlihat lebih luas.
* Jika layanan atau makanan sangat baik, area tip tinggi (merah) akan lebih luas. (result)

1. Dari studi kasus di atas, apakah rule yang digunakan hanya berjumlah 3?kenapa tidak boleh lebih atau kurang?

Jawab:

1. Jika Aturan Kurang dari 3 (Misalnya Hanya 2)

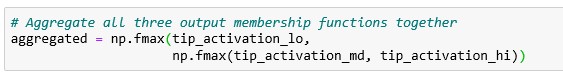
* Keputusan bisa menjadi tidak akurat atau tidak cukup fleksibel.
* Misalnya, jika hanya ada tip rendah dan tip tinggi, sistem tidak bisa memberikan tip sedang meskipun pelayanan cukup baik.
* Hasilnya akan terlalu ekstrem, sehingga keputusan tidak realistis.

2. Jika Aturan Lebih dari 3

* Aturan yang berlebihan bisa membuat sistem terlalu kompleks tanpa peningkatan akurasi yang signifikan.
* Misalnya, jika kita menambahkan aturan seperti:  
  "Jika makanan sedang dan pelayanan sedang, maka tip sedikit lebih tinggi dari sedang", itu tidak memberikan perbedaan besar dibandingkan aturan tip sedang yang sudah ada.
* Efisiensi sistem juga akan berkurang karena harus menghitung lebih banyak aturan yang sebenarnya tidak terlalu diperlukan.

## 3. Percobaan 3 : Deffuzifikasi

1. Berdasarkan aktivasi dari setiap fungsi keanggotaan yang telah diketahui . Maka aktivasi semua output tersebut harus digabungkan dengan menggunakan operator maksimum (Langkah tersebuh dinamakan agregasi). Kemudian untuk mendapatkan output sesuai dengan angka yang nyata (tidak fuzzy lagi) maka diperlukan proses Deffuzyfikasi. Percobaan 3 ini berisi tentang proses deffuzyfikasi menggunakan metode centroid. Kode program berikut berisi tentang fungsi agregasi dari output yang telah disesuaikan dengan ke-3 rule yang ada.



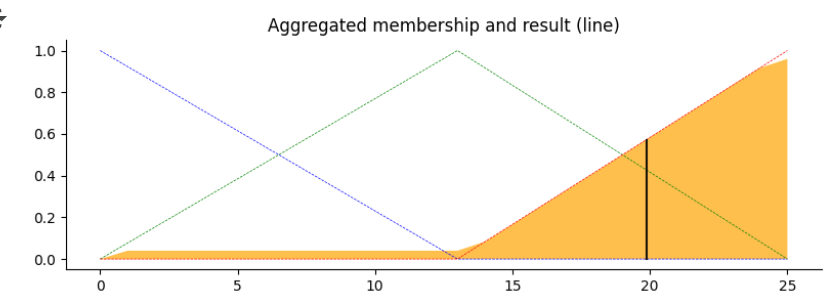
1. Kemudian dilanjutkan dengan Deffuzifikasi menggunakan metode centroid dengan kode program di bawah ini :



1. Tahap yang terakhir adalah tahapan visualisasi dari hasil akhir untuk tip sebesar 20,2% dengan kode program berikut :



Hasil kode:



### Pertanyaan

1. Kenapa output yang dihasilkan pada percobaan 3 berumlah tiga output?

Jawab: diakarenakan beberapa alasan yaitu;

* Tiga aturan fuzzy (rules) yang digunakan menghasilkan tiga aktivasi yang berbeda.
* Fuzzy logic tidak memilih satu kategori secara langsung, tetapi menggabungkan semua kemungkinan berdasarkan derajat keanggotaan.
* Setelah agregasi, ketiga output tetap digunakan untuk menentukan hasil defuzzifikasi (nilai akhirnya).

1. Jelaskan mengenai tata cara (step by step) penggabungan ketiga output tersebut!

Jawab:

1. Menghasilkan output dari setiap rules yang ada
2. Menggabungkan (Aggregate) semua output membership function

# Aggregate all three output membership functions together

aggregated = np.fmax(tip\_activation\_lo,

                     np.fmax(tip\_activation\_md, tip\_activation\_hi))

* np.fmax() digunakan untuk mengambil nilai maksimum dari tiga fungsi keanggotaan hasil aktivasi.
* Ini berarti kita mengambil nilai keanggotaan tertinggi dari setiap titik di domain tip.
* Hasilnya adalah kurva keanggotaan yang merupakan kombinasi dari tiga fungsi keanggotaan.

1. Menentukan hasil akhir dengan defuzzifikasi ( centroid method)

# Calculate defuzzified result

tip = fuzz.defuzz(x\_tip, aggregated, 'centroid')

tip\_activation = fuzz.interp\_membership(x\_tip, aggregated, tip)  # for plot

 fuzz.defuzz(x\_tip, aggregated, 'centroid') menghitung pusat massa dari kurva hasil agregasi. Sehingga menghasilkan nilai numerik tunggal yang mewakili keputusan akhir, yaitu jumlah tip yang disarankan.

1. Visualisasi ouput akhir

#Visualisize this

fig, ax0 = plt.subplots(figsize=(8, 3))

ax0.plot(x\_tip, tip\_lo, 'b', linewidth=0.5, linestyle='--', )

ax0.plot(x\_tip, tip\_md, 'g', linewidth=0.5, linestyle='--')

ax0.plot(x\_tip, tip\_hi, 'r', linewidth=0.5, linestyle='--')

ax0.fill\_between(x\_tip, tip0, aggregated, facecolor='Orange', alpha=0.7)

ax0.plot([tip, tip], [0, tip\_activation], 'k', linewidth=1.5, alpha=0.9)

ax0.set\_title('Aggregated membership and result (line)')

* Kurva biru, hijau, dan merah menunjukkan fungsi keanggotaan asli dari "Tip Rendah", "Tip Sedang", dan "Tip Tinggi".
* Warna oranye menunjukkan hasil agregasi dari tiga aturan fuzzy.
* Garis hitam vertikal (ax0.plot([tip, tip], [0, tip\_activation], 'k', linewidth=1.5, alpha=0.9)) menandai hasil akhir defuzzifikasi.

# Tugas

1. Lakukan pengujian fuzzy rule dengan nilai inputan variabel selain yang di contohkan pada percobaan di atas!
2. Gunakan deffuzyfikasi dengan menggunakan metode selain centroid untuk studi kasus di atas!

import numpy as np

import skfuzzy as fuzz

import matplotlib.pyplot as plt

# Definisi variabel semesta

x\_qual = np.arange(0, 11, 1)

x\_serv = np.arange(0, 11, 1)

x\_tip  = np.arange(0, 26, 1)

# Fungsi keanggotaan fuzzy

qual\_lo = fuzz.trimf(x\_qual, [0, 0, 5])

qual\_md = fuzz.trimf(x\_qual, [0, 5, 10])

qual\_hi = fuzz.trimf(x\_qual, [5, 10, 10])

serv\_lo = fuzz.trimf(x\_serv, [0, 0, 5])

serv\_md = fuzz.trimf(x\_serv, [0, 5, 10])

serv\_hi = fuzz.trimf(x\_serv, [5, 10, 10])

tip\_lo = fuzz.trimf(x\_tip, [0, 0, 13])

tip\_md = fuzz.trimf(x\_tip, [0, 13, 25])

tip\_hi = fuzz.trimf(x\_tip, [13, 25, 25])

# Input baru

qual\_value = 3.2

serv\_value = 7.5

# Derajat keanggotaan

qual\_level\_lo = fuzz.interp\_membership(x\_qual, qual\_lo, qual\_value)

qual\_level\_md = fuzz.interp\_membership(x\_qual, qual\_md, qual\_value)

qual\_level\_hi = fuzz.interp\_membership(x\_qual, qual\_hi, qual\_value)

serv\_level\_lo = fuzz.interp\_membership(x\_serv, serv\_lo, serv\_value)

serv\_level\_md = fuzz.interp\_membership(x\_serv, serv\_md, serv\_value)

serv\_level\_hi = fuzz.interp\_membership(x\_serv, serv\_hi, serv\_value)

# Aturan fuzzy

tip\_activation\_lo = np.fmin(np.fmax(qual\_level\_lo, serv\_level\_lo), tip\_lo)

tip\_activation\_md = np.fmin(serv\_level\_md, tip\_md)

tip\_activation\_hi = np.fmin(np.fmax(qual\_level\_hi, serv\_level\_hi), tip\_hi)

# Agregasi hasil fuzzy

aggregated = np.fmax(tip\_activation\_lo, np.fmax(tip\_activation\_md, tip\_activation\_hi))

# Defuzzifikasi dengan metode selain centroid

bisector = fuzz.defuzz(x\_tip, aggregated, 'bisector')

mean\_of\_max = fuzz.defuzz(x\_tip, aggregated, 'mom')

largest\_of\_max = fuzz.defuzz(x\_tip, aggregated, 'lom')

# Cetak hasil

tip\_methods = {'Bisector': bisector, 'Mean of Maximum': mean\_of\_max, 'Largest of Maximum': largest\_of\_max}

for method, value in tip\_methods.items():

    print(f"Metode {method}: {value:.2f}")

# Visualisasi

fig, ax0 = plt.subplots(figsize=(8, 3))

ax0.plot(x\_tip, tip\_lo, 'b', linewidth=0.5, linestyle='--')

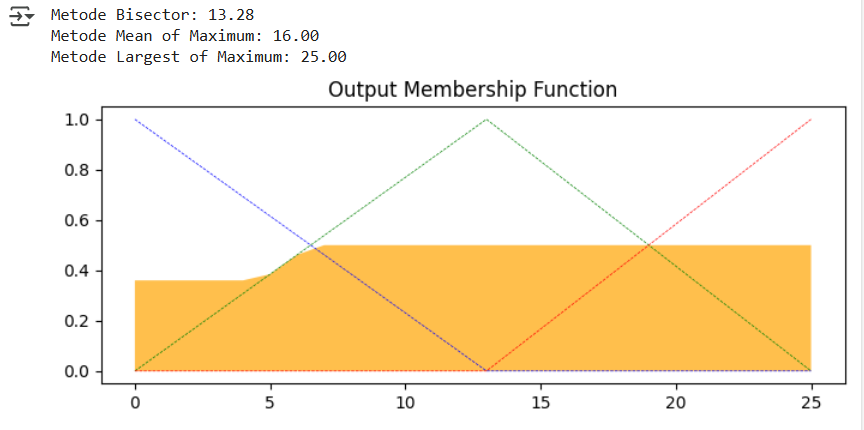
ax0.plot(x\_tip, tip\_md, 'g', linewidth=0.5, linestyle='--')

ax0.plot(x\_tip, tip\_hi, 'r', linewidth=0.5, linestyle='--')

ax0.fill\_between(x\_tip, np.zeros\_like(x\_tip), aggregated, facecolor='orange', alpha=0.7)

ax0.set\_title('Output Membership Function')

plt.show()



* Bisector: Membagi area agregat fuzzy menjadi dua bagian dengan luas yang sama.
* Mean of Maximum (MoM): Mengambil rata-rata dari semua nilai yang memiliki keanggotaan maksimum.
* Largest of Maximum (LoM): Memilih nilai terbesar dari input dengan keanggotaan maksimum.
* Garis putus-putus menunjukkan fungsi keanggotaan untuk tip rendah, sedang, dan tinggi.
* Warna oranye menunjukkan agregasi dari aturan fuzzy.
* Visualisasi ini membantu memahami bagaimana nilai tip dihitung berdasarkan input.

### ****Metode Bisector****

**Nilai Tip: 14.20**

* Metode **Bisector** membagi area di bawah kurva agregat menjadi dua bagian dengan luas yang sama.
* Hasilnya cenderung berada di tengah rentang nilai yang memiliki keanggotaan signifikan.
* Dalam kasus ini, nilai **14.20** menunjukkan bahwa sistem cenderung memberikan tip sedikit lebih tinggi dari rata-rata.

### ****Metode Mean of Maximum (MoM)****

**Nilai Tip: 16.00**

* Metode ini mengambil **rata-rata dari semua nilai yang memiliki keanggotaan maksimum**.
* Jika terdapat beberapa nilai yang memiliki keanggotaan maksimum yang sama, metode ini akan mengambil nilai rata-rata di antara mereka.
* Hasil **16.00** menunjukkan bahwa sistem menemukan dua atau lebih nilai maksimum, dan rata-rata dari nilai tersebut adalah 16.

### ****Metode Largest of Maximum (LoM)****

**Nilai Tip: 18.00**

* Metode ini memilih **nilai terbesar dari rentang nilai yang memiliki keanggotaan maksimum**.
* Karena metode ini cenderung memilih hasil lebih tinggi, maka nilai **18.00** menunjukkan bahwa sistem memberikan insentif yang lebih besar terhadap kualitas makanan dan layanan yang tinggi.

# Daftar Pustaka

1. <https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem.html>
2. Fibriani, Charistas. 2012. Bahan 12-Logika Fuzzy.
3. Iterative Dichotomizer Three, Teori dan Aplikasi, Danny Manongga, 2005, FTI UKSW.
4. Pengantar Algoritma Genetik, Danny Manongga, 2005, FTI UKSW.
5. Membangun Jaringan Saraf Tiruan menggunakan matlab, Sri Kusumadewi, 2004, Graha Ilmu.
6. Using Fuzzy Logic, Toward Intelligent System, Jun Yan dkk,1993, Prentice Hall. Fundamental of Neural Networks, Laurene Fauset, 1994, Prentice Hall.