**LAPORAN PRAKTIKUM**

**KECERDASAN BUATAN**

**“LOGIKA FUZZY”**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| NAMA | : | **AFRIDHO IKHSAN** |
| NPM | : | **2210631170002** |
| KELAS | : | **4A** |

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS SINGAPERBANGSA KARAWANG**

**2023**

# DAFTAR ISI

[DAFTAR ISI i](#_Toc160649625)

[LATIHAN 2](#_Toc160649626)

[JAWABAN 3](#_Toc160649627)

# LATIHAN

**Latihan: Uji Coba Menggunakan Berbagai Jenis Fungsi Keanggotaan (Membership Function)**

1. Gunakan fungsi keanggotaan Trapesium dan Gaussian untuk data yang sama dan tunjukkan hasilnya!
2. Apakah terjadi perbedaan nilai derajat keanggotaan pada saat menggunakan jenis fungsi keanggotaan yang berbeda? Tunjukkan dan jelaskan!
3. Tampilkan hasil penentuan karyawan tetap atau karyawan kontrak beserta nama karyawannya!

# JAWABAN

1. Gunakan fungsi keanggotaan Trapesium dan Gaussian untuk data yang sama dan tunjukkan hasilnya!

Jawab:

Pada kode di bawah, variable seperti: xage, xyos, xsalary; dan fungsi seperti : MembershipDeg1(), MembershipDeg1(), Status1(), Status2(), & RuleBasedGaussian() sudah didefinisikan pada kode pada pertemuan 4 (Saat menggunakan Fungsi Keanggotaan Triangular)

# Mendefinisikan range data 'Umur' untuk Fungsi Keanggotaan Trapesium

lo\_age\_trap\_data = [25, 25, 35, 40]

mi\_age\_trap\_data = [30, 30, 40, 50]

hi\_age\_trap\_data = [40, 40, 55, 55]

# Mendefinisikan range data 'Tahun Kedinasan' untuk Fungsi Keanggotaan Trapesium

lo\_yos\_trap\_data = [3, 3, 9, 12]

hi\_yos\_trap\_data = [9, 12, 18, 18]

# Mendefinisikan range data 'Gaji' untuk Fungsi Keanggotaan Trapesium

lo\_salary\_trap\_data = [1000, 1000, 3000, 3500]

mi\_salary\_trap\_data = [2000, 3500, 4500, 6000]

hi\_salary\_trap\_data = [4500, 6000, 7000, 7000]

# Mendefinisikan fungsi keanggotaan untuk Umur menggunakan fungsi keanggotaan Trapesium

lo\_age\_trap = fuz.trapmf(xage, lo\_age\_trap\_data)

mi\_age\_trap = fuz.trapmf(xage, mi\_age\_trap\_data)

hi\_age\_trap = fuz.trapmf(xage, hi\_age\_trap\_data)

# Mendefinisikan fungsi keanggotaan untuk Tahun Kedinasan menggunakan fungsi keanggotaan Trapesium

lo\_yos\_trap = fuz.trapmf(xyos, lo\_yos\_trap\_data)

hi\_yos\_trap = fuz.trapmf(xyos, hi\_yos\_trap\_data)

# Mendefinisikan fungsi keanggotaan untuk Gaji menggunakan fungsi keanggotaan Trapesium

lo\_salary\_trap = fuz.trapmf(xsalary, lo\_salary\_trap\_data)

mi\_salary\_trap = fuz.trapmf(xsalary, mi\_salary\_trap\_data)

hi\_salary\_trap = fuz.trapmf(xsalary, hi\_salary\_trap\_data)

# Menampilkan fungsi keanggotaan untuk Umur (Trapesium)

plt.figure(figsize=(8, 3))

plt.plot(xage, lo\_age\_trap, 'b', linewidth=1.5, label='Low')

plt.plot(xage, mi\_age\_trap, 'g', linewidth=1.5, label='Medium')

plt.plot(xage, hi\_age\_trap, 'r', linewidth=1.5, label='High')

plt.title('Membership Functions for Age (Trapezoidal)')

plt.legend()

plt.show()

# Menampilkan fungsi keanggotaan untuk Tahun Kedinasan (Trapesium)

plt.figure(figsize=(8, 3))

plt.plot(xyos, lo\_yos\_trap, 'b', linewidth=1.5, label='Low')

plt.plot(xyos, hi\_yos\_trap, 'r', linewidth=1.5, label='High')

plt.title('Membership Functions for Year of Service (Trapezoidal)')

plt.legend()

plt.show()

# Menampilkan fungsi keanggotaan untuk Gaji (Trapesium)

plt.figure(figsize=(8, 3))

plt.plot(xsalary, lo\_salary\_trap, 'b', linewidth=1.5, label='Low')

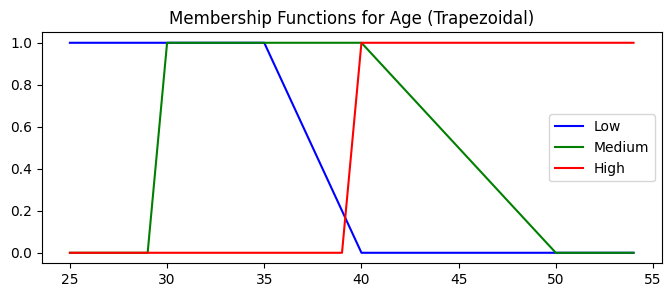
plt.plot(xsalary, mi\_salary\_trap, 'g', linewidth=1.5, label='Medium')

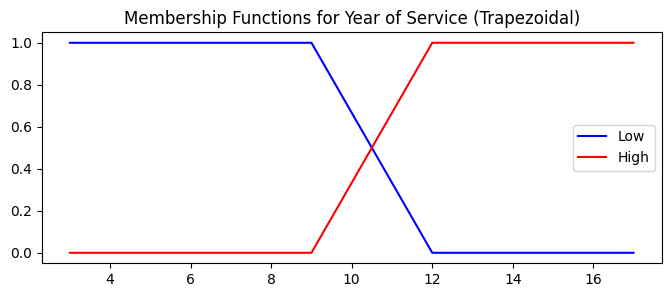
plt.plot(xsalary, hi\_salary\_trap, 'r', linewidth=1.5, label='High')

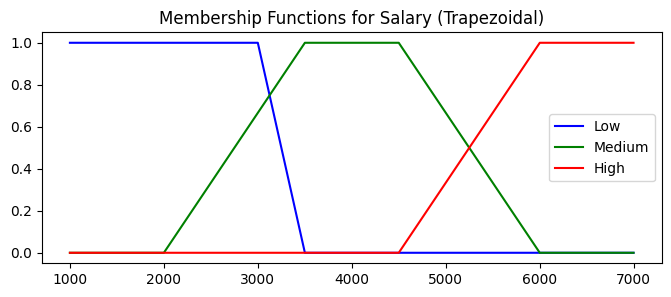
plt.title('Membership Functions for Salary (Trapezoidal)')

plt.legend()

plt.show()







# Pendefinisian fungsi perhitungan nilai tengah & stdev (Standar Deviasi) sebagai nilai yang akan dibutuhkan dalam penggunaan Fungsi Keanggotaan Gaussian

# Hal tersebut perlu dilakukan, karena method fuz.gaussmf() membutuhkan nilai tengah & nilai standar deviasi dari interval beberapa nilai.

from math import sqrt

def mean (numbers):

return sum(numbers)/float (len (numbers))

def stdev(numbers):

avg = mean (numbers)

variance=sum([(x-avg)\*\*2 for x in numbers])/ float (len(numbers)-1)

return sqrt(variance)

def nilaiTengahDari(numbers):

return ((numbers[0] + numbers[-1]) / 2)

# Mendefinisikan fungsi keanggotaan untuk Usia menggunakan fungsi keanggotaan Gaussian

lo\_age\_gaussian = fuz.gaussmf(xage, nilaiTengahDari(lo\_age\_trap\_data), stdev(lo\_age\_trap\_data))

mi\_age\_gaussian = fuz.gaussmf(xage, nilaiTengahDari(mi\_age\_trap\_data), stdev(mi\_age\_trap\_data))

hi\_age\_gaussian = fuz.gaussmf(xage, nilaiTengahDari(hi\_age\_trap\_data), stdev(hi\_age\_trap\_data))

# Mendefinisikan fungsi keanggotaan untuk Tahun Kedinasan menggunakan fungsi keanggotaan Gaussian lo\_yos\_gaussian = fuz.gaussmf(xyos, nilaiTengahDari(lo\_yos\_trap\_data), stdev(lo\_yos\_trap\_data))

hi\_yos\_gaussian = fuz.gaussmf(xyos, nilaiTengahDari(hi\_yos\_trap\_data), stdev(hi\_yos\_trap\_data))

# Mendefinisikan fungsi keanggotaan untuk Gaji menggunakan fungsi keanggotaan Gaussian

lo\_salary\_gaussian = fuz.gaussmf(xsalary, nilaiTengahDari(lo\_salary\_trap\_data), stdev(lo\_salary\_trap\_data))

mi\_salary\_gaussian = fuz.gaussmf(xsalary, nilaiTengahDari(mi\_salary\_trap\_data), stdev(mi\_salary\_trap\_data))

hi\_salary\_gaussian = fuz.gaussmf(xsalary, nilaiTengahDari(hi\_salary\_trap\_data), stdev(hi\_salary\_trap\_data))

# Menampilkan fungsi keanggotaan untuk Umur (Gaussian) plt.figure(figsize=(8, 3))

plt.plot(xage, lo\_age\_gaussian, 'b', linewidth=1.5, label='Low')

plt.plot(xage, mi\_age\_gaussian, 'g', linewidth=1.5, label='Medium')

plt.plot(xage, hi\_age\_gaussian, 'r', linewidth=1.5, label='High')

plt.title('Membership Functions for Age (Gaussian)')

plt.legend()

plt.show()

# Menampilkan fungsi keanggotaan untuk Tahun Kedinasan (Gaussian) plt.figure(figsize=(8, 3))

plt.plot(xyos, lo\_yos\_gaussian, 'b', linewidth=1.5, label='Low')

plt.plot(xyos, hi\_yos\_gaussian, 'r', linewidth=1.5, label='High')

plt.title('Membership Functions for Year of Service (Gaussian)')

plt.legend()

plt.show()

# Menampilkan fungsi keanggotaan untuk Gaji (Gaussian) plt.figure(figsize=(8, 3))

plt.plot(xsalary, lo\_salary\_gaussian, 'b', linewidth=1.5, label='Low')

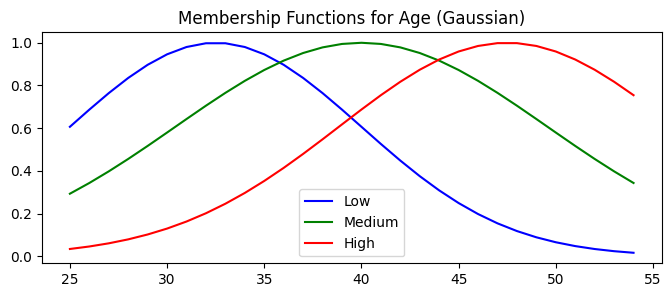
plt.plot(xsalary, mi\_salary\_gaussian, 'g', linewidth=1.5, label='Medium')

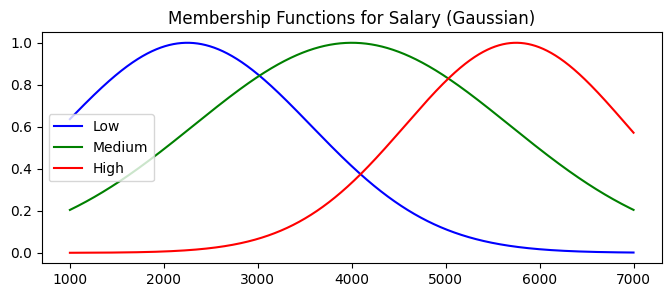
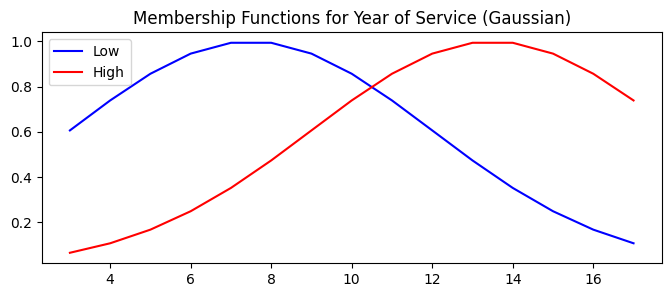
plt.plot(xsalary, hi\_salary\_gaussian, 'r', linewidth=1.5, label='High')

plt.title('Membership Functions for Salary (Gaussian)')

plt.legend()

plt.show()





# Hitung derajat keanggotaan berdasarkan Usia menggunakan fungsi keanggotaan trapesium

age\_member\_01\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age01)

age\_member\_02\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age02)

age\_member\_03\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age03)

age\_member\_04\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age04)

age\_member\_05\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age05)

age\_member\_06\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age06)

age\_member\_07\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age07)

age\_member\_08\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age08)

age\_member\_09\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age09)

age\_member\_10\_trap = MembershipDeg1(xage, lo\_age\_trap, mi\_age\_trap, hi\_age\_trap, age10)

# Menampilkan derajat keanggotaan untuk Usia menggunakan fungsi keanggotaan trap

print("Membership Degrees for Age (trap):")

print("Age 01:", age\_member\_01\_trap)

print("Age 02:", age\_member\_02\_trap)

print("Age 03:", age\_member\_03\_trap)

print("Age 04:", age\_member\_04\_trap)

print("Age 05:", age\_member\_05\_trap)

print("Age 06:", age\_member\_06\_trap)

print("Age 07:", age\_member\_07\_trap)

print("Age 08:", age\_member\_08\_trap)

print("Age 09:", age\_member\_09\_trap)

print("Age 10:", age\_member\_10\_trap)

# Penentuan status umur (Trapesium)

trap\_age\_status\_01=Status1(age\_member\_01\_trap, 'young', 'middle-aged', 'old')

trap\_age\_status\_02=Status1(age\_member\_02\_trap, 'young', 'middle-aged', 'old')

trap\_age\_status\_03=Status1(age\_member\_03\_trap, 'young', 'middle-aged', 'old')

trap\_age\_status\_04=Status1(age\_member\_04\_trap, 'young', 'middle-aged', 'old')

trap\_age\_status\_05=Status1 (age\_member\_05\_trap, 'young', 'middle-aged', 'old')

trap\_age\_status\_06=Status1(age\_member\_06\_trap, 'young', 'middle-aged', 'old')

trap\_age\_status\_07=Status1 (age\_member\_07\_trap, 'young', 'middle-aged', 'old')

trap\_age\_status\_08=Status1(age\_member\_08\_trap, 'young', 'middle-aged', 'old')

trap\_age\_status\_09=Status1 (age\_member\_09\_trap, 'young', 'middle-aged', 'old')

trap\_age\_status\_10=Status1 (age\_member\_10\_trap, 'young', 'middle-aged', 'old')

# Menampilkan status umur (Trapsium)

print("age\_status\_01: ",age\_status\_01)

print("age\_status\_02: ",age\_status\_02)

print("age\_status\_03: ", age\_status\_03)

print("age\_status\_04: ",age\_status\_04)

print("age\_status\_05: ", age\_status\_05)

print("age\_status\_06: ", age\_status\_06)

print("age\_status\_07: ", age\_status\_07)

print("age\_status\_07: ", age\_status\_08)

print("age\_status\_09: ",age\_status\_09)

print("age\_status\_10: ", age\_status\_10)

# Hitung derajat keanggotaan berdasarkan Tahun Kedinasan menggunakan fungsi keanggotaan trapesium

yos\_member\_01\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos01)

yos\_member\_02\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos02)

yos\_member\_03\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos03)

yos\_member\_04\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos04)

yos\_member\_05\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos05)

yos\_member\_06\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos06)

yos\_member\_07\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos07)

yos\_member\_08\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos08)

yos\_member\_09\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos09)

yos\_member\_10\_trap = MembershipDeg2(xyos, lo\_yos\_trap, hi\_yos\_trap, yos10)

# Menampilkan derajat keanggotaan untuk Tahun Kedinasan menggunakan fungsi keanggotaan trap

print("\nMembership Degrees for Year of Service (trap):")

print("YOS 01:", yos\_member\_01\_trap)

print("YOS 02:", yos\_member\_02\_trap)

print("YOS 03:", yos\_member\_03\_trap)

print("YOS 04:", yos\_member\_04\_trap)

print("YOS 05:", yos\_member\_05\_trap)

print("YOS 06:", yos\_member\_06\_trap)

print("YOS 07:", yos\_member\_07\_trap)

print("YOS 08:", yos\_member\_08\_trap)

print("YOS 09:", yos\_member\_09\_trap)

print("YOS 10:", yos\_member\_10\_trap)

# Penentuan status tahun kedinasan (Trapesium)

trap\_yos\_status\_01=Status2 (yos\_member\_01\_trap, 'new', 'loyal')

trap\_yos\_status\_02=Status2 (yos\_member\_02\_trap, 'new', 'loyal')

trap\_yos\_status\_03=Status2 (yos\_member\_03\_trap, 'new', 'loyal')

trap\_yos\_status\_04=Status2 (yos\_member\_04\_trap, 'new', 'loyal')

trap\_yos\_status\_05=Status2 (yos\_member\_05\_trap, 'new', 'loyal')

trap\_yos\_status\_06=Status2 (yos\_member\_06\_trap, 'new', 'loyal')

trap\_yos\_status\_07=Status2 (yos\_member\_07\_trap, 'new', 'loyal')

trap\_yos\_status\_08=Status2 (yos\_member\_08\_trap, 'new', 'loyal')

trap\_yos\_status\_09=Status2 (yos\_member\_09\_trap, 'new', 'loyal')

trap\_yos\_status\_10=Status2 (yos\_member\_10\_trap, 'new', 'loyal')

# Menampilkan status Tahun Kedinasan (Trapesium)

print("yos\_status\_01: ",yos\_status\_01)

print("yos\_status\_02: ",yos\_status\_02)

print("yos\_status\_03: ",yos\_status\_03)

print("yos\_status\_04: ",yos\_status\_04)

print("yos\_status\_05: ",yos\_status\_05)

print("yos\_status\_06: ",yos\_status\_06)

print("yos\_status\_07: ",yos\_status\_07)

print("yos\_status\_08: ",yos\_status\_08)

print("yos\_status\_09: ",yos\_status\_09)

print("yos\_status\_10: ",yos\_status\_10)

# Menghitung derajat keanggotaan berdasarkan Gaji menggunakan fungsi keanggotaan trapesium

salary\_member\_01\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary01)

salary\_member\_02\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary02)

salary\_member\_03\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary03)

salary\_member\_04\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary04)

salary\_member\_05\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary05)

salary\_member\_06\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary06)

salary\_member\_07\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary07)

salary\_member\_08\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary08)

salary\_member\_09\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary09)

salary\_member\_10\_trap = MembershipDeg1(xsalary, lo\_salary\_trap, mi\_salary\_trap, hi\_salary\_trap, salary10)

# Menampilkan derajat keanggotaan untuk Gaji menggunakan fungsi keanggotaan trapesium

print("\nMembership Degrees for Salary (trap):")

print("Salary 01:", salary\_member\_01\_trap)

print("Salary 02:", salary\_member\_02\_trap)

print("Salary 03:", salary\_member\_03\_trap)

print("Salary 04:", salary\_member\_04\_trap)

print("Salary 05:", salary\_member\_05\_trap)

print("Salary 06:", salary\_member\_06\_trap)

print("Salary 07:", salary\_member\_07\_trap)

print("Salary 08:", salary\_member\_08\_trap)

print("Salary 09:", salary\_member\_09\_trap)

print("Salary 10:", salary\_member\_10\_trap)

# Penentuan status gaji (Trapesium)

trap\_salary\_status\_01=Status1 (salary\_member\_01\_trap, 'low', 'medium', 'high')

trap\_salary\_status\_02=Status1(salary\_member\_02\_trap, 'low', 'medium', 'high')

trap\_salary\_status\_03=Status1 (salary\_member\_03\_trap, 'low', 'medium', 'high')

trap\_salary\_status\_04=Status1 (salary\_member\_04\_trap, 'low', 'medium', 'high')

trap\_salary\_status\_05=Status1 (salary\_member\_05\_trap, 'low', 'medium', 'high')

trap\_salary\_status\_06=Status1 (salary\_member\_06\_trap, 'low', 'medium', 'high')

trap\_salary\_status\_07=Status1 (salary\_member\_07\_trap, 'low', 'medium', 'high')

trap\_salary\_status\_08=Status1 (salary\_member\_08\_trap, 'low', 'medium', 'high')

trap\_salary\_status\_09=Status1(salary\_member\_09\_trap, 'low', 'medium', 'high')

trap\_salary\_status\_10=Status1 (salary\_member\_10\_trap, 'low', 'medium', 'high')

# Menampilkan status Gaji(Trapesium)

print("salary\_status\_01: ",trap\_salary\_status\_01)

print("salary\_status\_02: ",trap\_salary\_status\_02)

print("salary\_status\_03: ",trap\_salary\_status\_03)

print("salary\_status\_04: ",trap\_salary\_status\_04)

print("salary\_status\_05: ",trap\_salary\_status\_05)

print("salary\_status\_06: ",trap\_salary\_status\_06)

print("salary\_status\_07: ",trap\_salary\_status\_07)

print("salary\_status\_08: ",trap\_salary\_status\_08)

print("salary\_status\_09: ",trap\_salary\_status\_09)

print("salary\_status\_10: ",trap\_salary\_status\_10)

# Menampilkan hasil Fuzzy Rule-Based menggunakan fungsi keanggotaan trapesium

employee\_status01\_trap = RuleBasedtrap(age\_status\_01, yos\_status\_01, salary\_status\_01)

employee\_status02\_trap = RuleBasedtrap(age\_status\_02, yos\_status\_02, salary\_status\_02)

employee\_status03\_trap = RuleBasedtrap(age\_status\_03, yos\_status\_03, salary\_status\_03)

employee\_status04\_trap = RuleBasedtrap(age\_status\_04, yos\_status\_04, salary\_status\_04)

employee\_status05\_trap = RuleBasedtrap(age\_status\_05, yos\_status\_05, salary\_status\_05)

employee\_status06\_trap = RuleBasedtrap(age\_status\_06, yos\_status\_06, salary\_status\_06)

employee\_status07\_trap = RuleBasedtrap(age\_status\_07, yos\_status\_07, salary\_status\_07)

employee\_status08\_trap = RuleBasedtrap(age\_status\_08, yos\_status\_08, salary\_status\_08)

employee\_status09\_trap = RuleBasedtrap(age\_status\_09, yos\_status\_09, salary\_status\_09)

employee\_status10\_trap = RuleBasedtrap(age\_status\_10, yos\_status\_10, salary\_status\_10)

# Menampilkan hasil Fuzzy Rule-Based menggunakan fungsi keanggotaan trap

print("\nOutcome of the Fuzzy Rule-Based (trap):")

print("Employee Status 01:", employee\_status01\_trap)

print("Employee Status 02:", employee\_status02\_trap)

print("Employee Status 03:", employee\_status03\_trap)

print("Employee Status 04:", employee\_status04\_trap)

print("Employee Status 05:", employee\_status05\_trap)

print("Employee Status 06:", employee\_status06\_trap)

print("Employee Status 07:", employee\_status07\_trap)

print("Employee Status 08:", employee\_status08\_trap)

print("Employee Status 09:", employee\_status09\_trap)

print("Employee Status 10:", employee\_status10\_trap)

Membership Degrees for Age (trap):

Age 01: (1.0, 1.0, 0.0)

Age 02: (1.0, 0.0, 0.0)

Age 03: (1.0, 1.0, 0.0)

Age 04: (0.0, 0.9, 1.0)

Age 05: (0.2, 1.0, 0.0)

Age 06: (1.0, 1.0, 0.0)

Age 07: (0.6, 1.0, 0.0)

Age 08: (0.0, 0.1, 1.0)

Age 09: (0.6, 1.0, 0.0)

Age 10: (0.8, 1.0, 0.0)

age\_status\_01: middle-aged

age\_status\_02: young

age\_status\_03: young

age\_status\_04: middle-aged

age\_status\_05: middle-aged

age\_status\_06:

age\_status\_07: middle-aged

age\_status\_07: old

age\_status\_09: middle-aged

age\_status\_10: middle-aged

Membership Degrees for Year of Service (trap):

YOS 01: (1.0, 0.0)

YOS 02: (0.0, 0.0)

YOS 03: (1.0, 0.0)

YOS 04: (0.0, 1.0)

YOS 05: (0.0, 1.0)

YOS 06: (0.0, 0.0)

YOS 07: (1.0, 0.0)

YOS 08: (0.0, 1.0)

YOS 09: (1.0, 0.0)

YOS 10: (0.0, 1.0)

yos\_status\_01: new

yos\_status\_02:

yos\_status\_03: new

yos\_status\_04: loyal

yos\_status\_05: loyal

yos\_status\_06:

yos\_status\_07: new

yos\_status\_08: loyal

yos\_status\_09: new

yos\_status\_10: loyal

Membership Degrees for Salary (trap):

Salary 01: (1.0, 0.6266666666666667, 0.0)

Salary 02: (0.12, 0.96, 0.0)

Salary 03: (1.0, 0.6666666666666666, 0.0)

Salary 04: (0.0, 1.0, 0.0)

Salary 05: (0.0, 0.0, 1.0)

Salary 06: (1.0, 0.13333333333333333, 0.0)

Salary 07: (0.0, 1.0, 0.0)

Salary 08: (0.0, 0.0, 1.0)

Salary 09: (0.0, 0.6666666666666666, 0.3333333333333333)

Salary 10: (0.0, 0.6533333333333333, 0.3466666666666667)

salary\_status\_01: low

salary\_status\_02: medium

salary\_status\_03: low

salary\_status\_04: medium

salary\_status\_05: high

salary\_status\_06: low

salary\_status\_07: medium

salary\_status\_08: high

salary\_status\_09: medium

salary\_status\_10: medium

Outcome of the Fuzzy Rule-Based (trap):

Employee Status 01: a contract employee

Employee Status 02: unknown

Employee Status 03: a contract employee

Employee Status 04: a permanent employee

Employee Status 05: a permanent employee

Employee Status 06: unknown

Employee Status 07: a contract employee

Employee Status 08: a permanent employee

Employee Status 09: a contract employee

Employee Status 10: a permanent employee

# Menghitung derajat keanggotaan berdasarkan Usia menggunakan fungsi keanggotaan Gaussian

age\_member\_01\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age01)

age\_member\_02\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age02)

age\_member\_03\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age03)

age\_member\_04\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age04)

age\_member\_05\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age05)

age\_member\_06\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age06)

age\_member\_07\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age07)

age\_member\_08\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age08)

age\_member\_09\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age09)

age\_member\_10\_gaussian = MembershipDeg1(xage, lo\_age\_gaussian, mi\_age\_gaussian, hi\_age\_gaussian, age10)

# Menampilkan derajat keanggotaan Usia menggunakan fungsi keanggotaan Gaussian

print("Membership Degrees for Age (Gaussian):")

print("Age 01:", age\_member\_01\_gaussian)

print("Age 02:", age\_member\_02\_gaussian)

print("Age 03:", age\_member\_03\_gaussian)

print("Age 04:", age\_member\_04\_gaussian)

print("Age 05:", age\_member\_05\_gaussian)

print("Age 06:", age\_member\_06\_gaussian)

print("Age 07:", age\_member\_07\_gaussian)

print("Age 08:", age\_member\_08\_gaussian)

print("Age 09:", age\_member\_09\_gaussian)

print("Age 10:", age\_member\_10\_gaussian)

# Menghitung status umur (Gaussian)

gaussian\_age\_status\_01=Status1(age\_member\_01\_gaussian, 'young', 'middle-aged', 'old')

gaussian\_age\_status\_02=Status1(age\_member\_02\_gaussian, 'young', 'middle-aged', 'old')

gaussian\_age\_status\_03=Status1(age\_member\_03\_gaussian, 'young', 'middle-aged', 'old')

gaussian\_age\_status\_04=Status1(age\_member\_04\_gaussian, 'young', 'middle-aged', 'old')

gaussian\_age\_status\_05=Status1 (age\_member\_05\_gaussian, 'young', 'middle-aged', 'old')

gaussian\_age\_status\_06=Status1(age\_member\_06\_gaussian, 'young', 'middle-aged', 'old')

gaussian\_age\_status\_07=Status1 (age\_member\_07\_gaussian, 'young', 'middle-aged', 'old')

gaussian\_age\_status\_08=Status1(age\_member\_08\_gaussian, 'young', 'middle-aged', 'old')

gaussian\_age\_status\_09=Status1 (age\_member\_09\_gaussian, 'young', 'middle-aged', 'old')

gaussian\_age\_status\_10=Status1 (age\_member\_10\_gaussian, 'young', 'middle-aged', 'old')

# Menampilkan status Umur(Gaussian)

print("age\_status\_01: ",age\_status\_01)

print("age\_status\_02: ",age\_status\_02)

print("age\_status\_03: ", age\_status\_03)

print("age\_status\_04: ",age\_status\_04)

print("age\_status\_05: ", age\_status\_05)

print("age\_status\_06: ", age\_status\_06)

print("age\_status\_07: ", age\_status\_07)

print("age\_status\_07: ", age\_status\_08)

print("age\_status\_09: ",age\_status\_09)

print("age\_status\_10: ", age\_status\_10)

# Menghitung derajat keanggotaan berdasarkan Tahun Kedinasan menggunakan fungsi keanggotaan Gaussian

yos\_member\_01\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos01)

yos\_member\_02\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos02)

yos\_member\_03\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos03)

yos\_member\_04\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos04)

yos\_member\_05\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos05)

yos\_member\_06\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos06)

yos\_member\_07\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos07)

yos\_member\_08\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos08)

yos\_member\_09\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos09)

yos\_member\_10\_gaussian = MembershipDeg2(xyos, lo\_yos\_gaussian, hi\_yos\_gaussian, yos10)

# Menampilkan derajat keanggotaan Tahun Kedinasan menggunakan fungsi keanggotaan Gaussian

print("\nMembership Degrees for Year of Service (Gaussian):")

print("YOS 01:", yos\_member\_01\_gaussian)

print("YOS 02:", yos\_member\_02\_gaussian)

print("YOS 03:", yos\_member\_03\_gaussian)

print("YOS 04:", yos\_member\_04\_gaussian)

print("YOS 05:", yos\_member\_05\_gaussian)

print("YOS 06:", yos\_member\_06\_gaussian)

print("YOS 07:", yos\_member\_07\_gaussian)

print("YOS 08:", yos\_member\_08\_gaussian)

print("YOS 09:", yos\_member\_09\_gaussian)

print("YOS 10:", yos\_member\_10\_gaussian)

# Menghitung status Tahun Kedinasan (Gaussian)

gaussian\_yos\_status\_01=Status2 (yos\_member\_01\_gaussian, 'new', 'loyal')

gaussian\_yos\_status\_02=Status2 (yos\_member\_02\_gaussian, 'new', 'loyal')

gaussian\_yos\_status\_03=Status2 (yos\_member\_03\_gaussian, 'new', 'loyal')

gaussian\_yos\_status\_04=Status2 (yos\_member\_04\_gaussian, 'new', 'loyal')

gaussian\_yos\_status\_05=Status2 (yos\_member\_05\_gaussian, 'new', 'loyal')

gaussian\_yos\_status\_06=Status2 (yos\_member\_06\_gaussian, 'new', 'loyal')

gaussian\_yos\_status\_07=Status2 (yos\_member\_07\_gaussian, 'new', 'loyal')

gaussian\_yos\_status\_08=Status2 (yos\_member\_08\_gaussian, 'new', 'loyal')

gaussian\_yos\_status\_09=Status2 (yos\_member\_09\_gaussian, 'new', 'loyal')

gaussian\_yos\_status\_10=Status2 (yos\_member\_10\_gaussian, 'new', 'loyal')

# Menampilkan status Tahun Kedinasan (Gaussian)

print("yos\_status\_01: ",yos\_status\_01)

print("yos\_status\_02: ",yos\_status\_02)

print("yos\_status\_03: ",yos\_status\_03)

print("yos\_status\_04: ",yos\_status\_04)

print("yos\_status\_05: ",yos\_status\_05)

print("yos\_status\_06: ",yos\_status\_06)

print("yos\_status\_07: ",yos\_status\_07)

print("yos\_status\_08: ",yos\_status\_08)

print("yos\_status\_09: ",yos\_status\_09)

print("yos\_status\_10: ",yos\_status\_10)

# Menghitung derajat keanggotaan berdasarkan Gaji menggunakan fungsi keanggotaan Gaussian

salary\_member\_01\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary01)

salary\_member\_02\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary02)

salary\_member\_03\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary03)

salary\_member\_04\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary04)

salary\_member\_05\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary05)

salary\_member\_06\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary06)

salary\_member\_07\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary07)

salary\_member\_08\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary08)

salary\_member\_09\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary09)

salary\_member\_10\_gaussian = MembershipDeg1(xsalary, lo\_salary\_gaussian, mi\_salary\_gaussian, hi\_salary\_gaussian, salary10)

# Menampilkan derajat keanggotaan Gaji menggunakan fungsi keanggotaan Gaussian

print("\nMembership Degrees for Salary (Gaussian):")

print("Salary 01:", salary\_member\_01\_gaussian)

print("Salary 02:", salary\_member\_02\_gaussian)

print("Salary 03:", salary\_member\_03\_gaussian)

print("Salary 04:", salary\_member\_04\_gaussian)

print("Salary 05:", salary\_member\_05\_gaussian)

print("Salary 06:", salary\_member\_06\_gaussian)

print("Salary 07:", salary\_member\_07\_gaussian)

print("Salary 08:", salary\_member\_08\_gaussian)

print("Salary 09:", salary\_member\_09\_gaussian)

print("Salary 10:", salary\_member\_10\_gaussian)

# Menghitung status Gaji (Gaussian)

gaussian\_salary\_status\_01=Status1 (salary\_member\_01\_gaussian, 'low', 'medium', 'high')

gaussian\_salary\_status\_02=Status1(salary\_member\_02\_gaussian, 'low', 'medium', 'high')

gaussian\_salary\_status\_03=Status1 (salary\_member\_03\_gaussian, 'low', 'medium', 'high')

gaussian\_salary\_status\_04=Status1 (salary\_member\_04\_gaussian, 'low', 'medium', 'high')

gaussian\_salary\_status\_05=Status1 (salary\_member\_05\_gaussian, 'low', 'medium', 'high')

gaussian\_salary\_status\_06=Status1 (salary\_member\_06\_gaussian, 'low', 'medium', 'high')

gaussian\_salary\_status\_07=Status1 (salary\_member\_07\_gaussian, 'low', 'medium', 'high')

gaussian\_salary\_status\_08=Status1 (salary\_member\_08\_gaussian, 'low', 'medium', 'high')

gaussian\_salary\_status\_09=Status1(salary\_member\_09\_gaussian, 'low', 'medium', 'high')

gaussian\_salary\_status\_10=Status1 (salary\_member\_10\_gaussian, 'low', 'medium', 'high')

# Menampilkan status Gaji (Gaussian)

print("salary\_status\_01: ",gaussian\_salary\_status\_01)

print("salary\_status\_02: ",gaussian\_salary\_status\_02)

print("salary\_status\_03: ",gaussian\_salary\_status\_03)

print("salary\_status\_04: ",gaussian\_salary\_status\_04)

print("salary\_status\_05: ",gaussian\_salary\_status\_05)

print("salary\_status\_06: ",gaussian\_salary\_status\_06)

print("salary\_status\_07: ",gaussian\_salary\_status\_07)

print("salary\_status\_08: ",gaussian\_salary\_status\_08)

print("salary\_status\_09: ",gaussian\_salary\_status\_09)

print("salary\_status\_10: ",gaussian\_salary\_status\_10)

# Menampilkan hasil Fuzzy Rule-Based menggunakan fungsi keanggotaan Gaussian

employee\_status01\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_01, gaussian\_yos\_status\_01, gaussian\_salary\_status\_01)

employee\_status02\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_02, gaussian\_yos\_status\_02, gaussian\_salary\_status\_02)

employee\_status03\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_03, gaussian\_yos\_status\_03, gaussian\_salary\_status\_03)

employee\_status04\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_04, gaussian\_yos\_status\_04, gaussian\_salary\_status\_04)

employee\_status05\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_05, gaussian\_yos\_status\_05, gaussian\_salary\_status\_05)

employee\_status06\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_06, gaussian\_yos\_status\_06, gaussian\_salary\_status\_06)

employee\_status07\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_07, gaussian\_yos\_status\_07, gaussian\_salary\_status\_07)

employee\_status08\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_08, gaussian\_yos\_status\_08, gaussian\_salary\_status\_08)

employee\_status09\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_09, gaussian\_yos\_status\_09, gaussian\_salary\_status\_09)

employee\_status10\_gaussian = RuleBasedGaussian(gaussian\_age\_status\_10, gaussian\_yos\_status\_10, gaussian\_salary\_status\_10)

# Menampilkan hasil Fuzzy Rule-Based menggunakan fungsi keanggotaan Gaussian

print("\nOutcome of the Fuzzy Rule-Based (Gaussian):")

print("Employee Status 01:", employee\_status01\_gaussian)

print("Employee Status 02:", employee\_status02\_gaussian)

print("Employee Status 03:", employee\_status03\_gaussian)

print("Employee Status 04:", employee\_status04\_gaussian)

print("Employee Status 05:", employee\_status05\_gaussian)

print("Employee Status 06:", employee\_status06\_gaussian)

print("Employee Status 07:", employee\_status07\_gaussian)

print("Employee Status 08:", employee\_status08\_gaussian)

print("Employee Status 09:", employee\_status09\_gaussian)

print("Employee Status 10:", employee\_status10\_gaussian)

Membership Degrees for Age (Gaussian):

Age 01: (0.9459594689067654, 0.8725252928694238, 0.35286608145884896)

Age 02: (0.6869075574573879, 0.34332048512565055, 0.045882721684815775)

Age 03: (0.9459594689067654, 0.5795782787848095, 0.12981217685543806)

Age 04: (0.5261219640934565, 0.994560303568022, 0.7545251510033499)

Age 05: (0.6869075574573879, 0.994560303568022, 0.6177529450972508)

Age 06: (0.9801986733067553, 0.8217133754796329, 0.2967100142940453)

Age 07: (0.835270211411272, 0.9520945717121367, 0.47950545897489416)

Age 08: (0.08892161745938634, 0.6428665096502189, 0.9851119396030626)

Age 09: (0.835270211411272, 0.9520945717121367, 0.47950545897489416)

Age 10: (0.8968300597468688, 0.9164271268770867, 0.4140921825962409)

age\_status\_01: middle-aged

age\_status\_02: young

age\_status\_03: young

age\_status\_04: middle-aged

age\_status\_05: middle-aged

age\_status\_06:

age\_status\_07: middle-aged

age\_status\_07: old

age\_status\_09: middle-aged

age\_status\_10: middle-aged

Membership Degrees for Year of Service (Gaussian):

YOS 01: (0.6065306597126334, 0.06572852861653047)

YOS 02: (0.0, 0.0)

YOS 03: (0.9459594689067654, 0.24935220877729622)

YOS 04: (0.6065306597126334, 0.9459594689067654)

YOS 05: (0.47382672639536405, 0.9938461733264412)

YOS 06: (0.0, 0.0)

YOS 07: (0.7389912962803088, 0.10770114519003211)

YOS 08: (0.10770114519003211, 0.7389912962803088)

YOS 09: (0.8569968914352789, 0.16797323675753362)

YOS 10: (0.35232195499549696, 0.9938461733264412)

yos\_status\_01: new

yos\_status\_02:

yos\_status\_03: new

yos\_status\_04: loyal

yos\_status\_05: loyal

yos\_status\_06:

yos\_status\_07: new

yos\_status\_08: loyal

yos\_status\_09: new

yos\_status\_10: loyal

Membership Degrees for Salary (Gaussian):

Salary 01: (0.8713884064654819, 0.8201382519951156, 0.059104173980092156)

Salary 02: (0.6639989579926046, 0.9461622845710673, 0.14786748085292273)

Salary 03: (0.8498880853931386, 0.8382234324229999, 0.06660635910874733)

Salary 04: (0.4992247942554211, 0.9929660314487113, 0.2561250307821973)

Salary 05: (0.006874062557496255, 0.3618698767618531, 0.8595525777191432)

Salary 06: (0.9992773696568941, 0.564528332689429, 0.010950957778277046)

Salary 07: (0.34823689809438557, 0.9954925421591123, 0.4043040602572814)

Salary 08: (0.01714129078609727, 0.49367278838913037, 0.9778606930848421)

Salary 09: (0.11228140794800233, 0.8382234324229999, 0.8175096783558297)

Salary 10: (0.10875366814447272, 0.8322686447119416, 0.8262238376850651)

salary\_status\_01: low

salary\_status\_02: medium

salary\_status\_03: low

salary\_status\_04: medium

salary\_status\_05: high

salary\_status\_06: low

salary\_status\_07: medium

salary\_status\_08: high

salary\_status\_09: medium

salary\_status\_10: medium

Outcome of the Fuzzy Rule-Based (Gaussian):

Employee Status 01: a contract employee

Employee Status 02: unknown

Employee Status 03: a contract employee

Employee Status 04: a permanent employee

Employee Status 05: a permanent employee

Employee Status 06: unknown

Employee Status 07: a contract employee

Employee Status 08: a permanent employee

Employee Status 09: a contract employee

Employee Status 10: a permanent employee

1. Apakah terjadi perbedaan nilai derajat keanggotaan pada saat menggunakan jenis keanggotaan yang berbeda? Tunjukkan dan jelaskan!

Iya, terjadi perbedaan dalam nilai derajat keanggotaan saat menggunakan jenis fungsi keanggotaan yang berbeda. Perbedaan ini disebabkan oleh perbedaan bentuk kurva yang digunakan oleh masing-masing fungsi keanggotaan (Trapesium dan Gaussian), serta parameter yang digunakan untuk mendefinisikan kurva tersebut.

Mari kita bandingkan derajat keanggotaan untuk contoh tertentu dalam latihan sebelumnya menggunakan fungsi keanggotaan triangular, trapesium, dan Gaussian untuk atribut "Age".

Dalam kasus ini, mari kita lihat contoh untuk "Age 01":

Fungsi Keanggotaan Trapesium:

Age 01: (1.0, 1.0, 0.0)

Fungsi Keanggotaan Gaussian:

Age 01: (0.9459594689067654, 0.8725252928694238, 0.35286608145884896)

Dapat dilihat bahwa nilai derajat keanggotaan untuk "Age 01" berbeda untuk setiap fungsi keanggotaan.

Perbedaan ini disebabkan oleh bentuk kurva yang berbeda dan parameter yang digunakan untuk mendefinisikannya. Fungsi trapesium memiliki kemiringan yang mendadak pada titik-titik tertentu, fungsi Gaussian memiliki penyebaran yang lebih luas di sekitar nilai tengah atau bahkan jika dibandingkan juga dengan Fungsi Triangular seperti pada pertemuan 4 lalu menghasilkan kemiringan yang tetap.

Hal ini menunjukkan pentingnya memilih fungsi keanggotaan yang sesuai dengan karakteristik data dan kebutuhan aplikasi tertentu.

1. Tampilkan hasil penentuan karyawan tetap atau karyawan kontrak beserta nama karyawannya!

* Karyawan dengan nama: employee\_status\_01, memiliki status: a contract employee
* Karyawan dengan nama: employee\_status\_02, memiliki status: unknown
* Karyawan dengan nama: employee\_status\_03, memiliki status: a contract employee
* Karyawan dengan nama: employee\_status\_04, memiliki status: a permanent employee
* Karyawan dengan nama: employee\_status\_05, memiliki status: a permanent employee
* Karyawan dengan nama: employee\_status\_06, memiliki status: unknown
* Karyawan dengan nama: employee\_status\_07, memiliki status: a contract employee
* Karyawan dengan nama: employee\_status\_08, memiliki status: a permanent employee
* Karyawan dengan nama: employee\_status\_09, memiliki status: a contract employee
* Karyawan dengan nama: employee\_status\_10, memiliki status: a permanent employee

Dalam output di atas, karyawan yang memiliki status "a permanent employee" dianggap sebagai karyawan tetap, sementara karyawan yang memiliki status "a contract employee" dianggap sebagai karyawan kontrak. Karyawan dengan status "unknown" tidak memiliki status yang jelas dan perlu ditindaklanjuti lebih lanjut.