

NAMA : RIKI RAMADAN

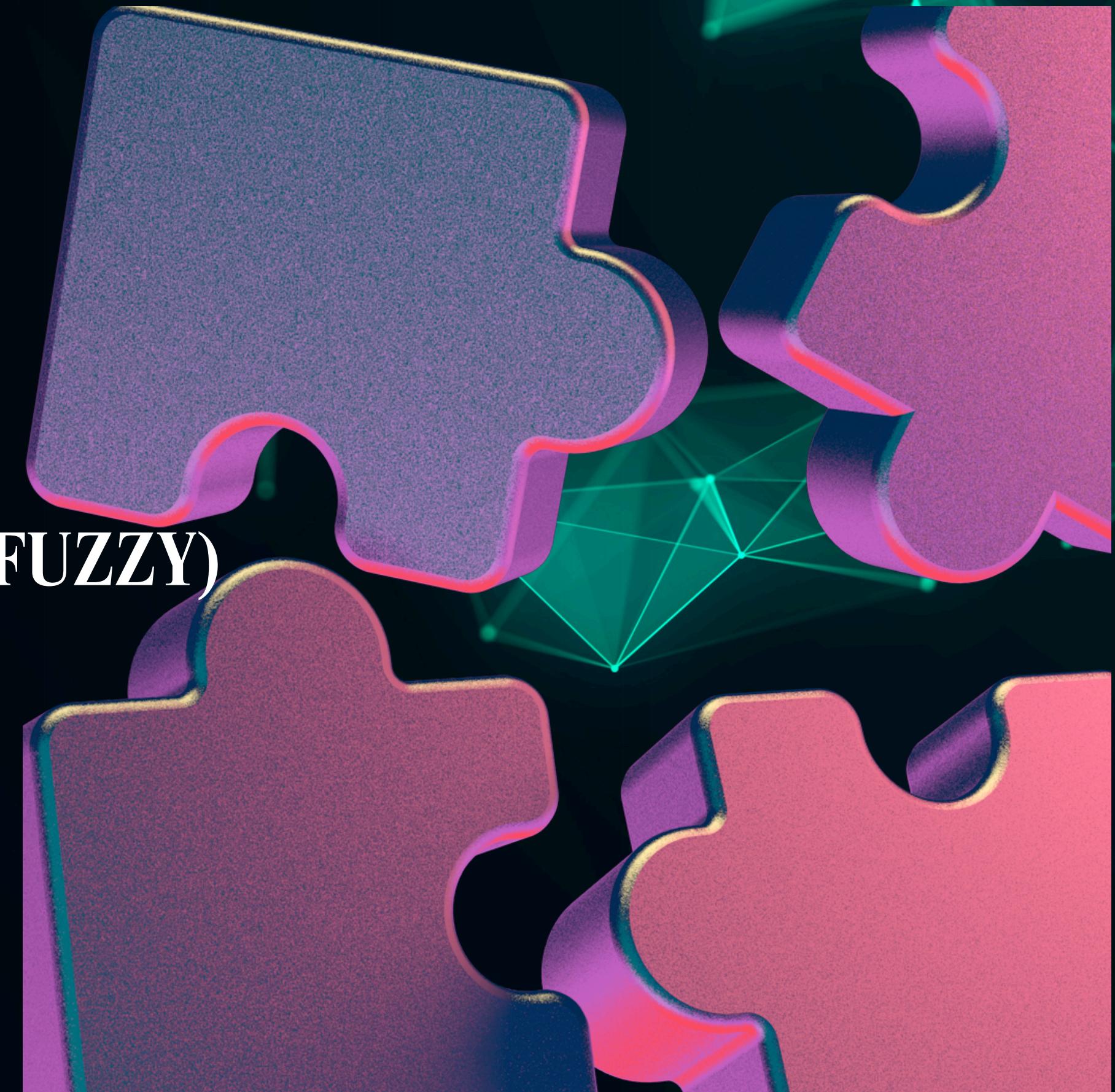
NIM : 221011401392

KELAS : 05TPLM007

TUGAS : UAS KECERDASAN BUATAN (FUZZY)

LINK GITHUB :

<https://github.com/RikiRamadan12/Riki-Ramadan-Tugas-UAS-Fuzzy-221011401392-05TPLM007>



FUNGSI KEANGGOTAAN

Judul: Fungsi Keanggotaan

Kode: Gambar 1 (Bagian MembershipFunction)

Penjelasan:

1. MembershipFunction Class: Digunakan untuk mendefinisikan fungsi keanggotaan (segitiga dan trapesium).

2. Metode:

- calculate: Menghitung derajat keanggotaan untuk input tertentu berdasarkan parameter fungsi keanggotaan.
- plot: Membuat grafik fungsi keanggotaan.

3. Penggunaan:

- Mendefinisikan input seperti suhu atau kelembapan dengan bentuk segitiga atau trapesium.

```
t numpy as np
t matplotlib.pyplot as plt

gsi Keanggotaan
MembershipFunction:
ef __init__(self, name, points):
    self.name = name
    self.points = points

ef calculate(self, x):
    if len(self.points) == 3: # Segitiga
        a, b, c = self.points
        if a <= x < b:
            return (x - a) / (b - a)
        elif b <= x < c:
            return (c - x) / (c - b)
        else:
            return 0.0
    elif len(self.points) == 4: # Trapesium
        a, b, c, d = self.points
        if a <= x < b:
            return (x - a) / (b - a)
        elif b <= x <= c:
            return 1.0
        elif c < x <= d:
            return (d - x) / (d - c)
        else:
            return 0.0

ef plot(self, x_range):
    y_values = [self.calculate(x) for x in x_range]
    plt.plot(x_range, y_values, label=self.name)
```

FUZZIFIKASI & EVALUASI

- Fuzzification Class

Penjelasan:

1. Fuzzification Class: Menerjemahkan input nilai crisp (misalnya suhu atau kelembapan aktual) menjadi nilai fuzzy.

2. Metode:

- fuzzify: Menghitung derajat keanggotaan untuk setiap fungsi keanggotaan pada input tertentu.

3. Output: Dictionary dengan nama fungsi keanggotaan sebagai kunci dan nilai keanggotaan sebagai nilai.

- Evaluasi Aturan Fuzzy

penjelasan:

1. RuleEvaluation Class: Mengevaluasi aturan fuzzy berdasarkan nilai fuzzy dari input.

2. Metode:

- evaluate: Menggunakan logika min untuk menghitung kekuatan aturan fuzzy.

3. Output: Daftar aturan dengan kekuatan aturan dan output yang dihasilkan.

```
# Fuzzifikasi
class Fuzzification:
    def __init__(self, membership_functions):
        self.membership_functions = membership_functions

    def fuzzify(self, value):
        fuzzy_values = {}
        for mf in self.membership_functions:
            fuzzy_values[mf.name] = mf.calculate(value)
        return fuzzy_values

# Evaluasi Aturan Fuzzy
class RuleEvaluation:
    def __init__(self, rules):
        self.rules = rules

    def evaluate(self, temp_fuzzy, hum_fuzzy):
        rule_results = []
        for rule in self.rules:
            temp_level, hum_level, output = rule
            rule_strength = min(temp_fuzzy[temp_level], hum_fuzzy[hum_level])
            rule_results.append((rule_strength, output))
        return rule_results
```

DEFUZZIFIKASI

Penjelasan:

Defuzzification Class: Mengubah hasil fuzzy menjadi nilai crisp.

Metode:

defuzzify: Menggunakan metode rata-rata berbobot (weighted average) untuk menghasilkan output crisp.

Output: Nilai akhir (misalnya suhu AC yang harus diatur).

```
[5]: # Defuzzifikasi
class Defuzzification:
    def __init__(self, output_ranges):
        self.output_ranges = output_ranges

    def defuzzify(self, rules):
        numerator = 0.0
        denominator = 0.0
        for strength, output_name in rules:
            crisp_value = self.output_ranges[output_name]
            numerator += strength * crisp_value
            denominator += strength
        return numerator / denominator if denominator != 0 else 0.0
```

```

# Sistem Inferensi Fuzzy
class FuzzySystem:
    def __init__(self, temp_mfs, hum_mfs, rules, output_ranges):
        self.temp_fuzz = Fuzzification(temp_mfs)
        self.hum_fuzz = Fuzzification(hum_mfs)
        self.rule_eval = RuleEvaluation(rules)
        self.defuzz = Defuzzification(output_ranges)

    def calculate(self, temp_value, hum_value):
        temp_fuzzy = self.temp_fuzz.fuzzify(temp_value)
        hum_fuzzy = self.hum_fuzz.fuzzify(hum_value)
        rule_results = self.rule_eval.evaluate(temp_fuzzy, hum_fuzzy)
        return self.defuzz.defuzzify(rule_results)

```

SISTEM INFERENSI FUZZY

Kode ini mendefinisikan sebuah kelas bernama `FuzzySystem` yang bertanggung jawab untuk melakukan proses inferensi fuzzy. Penjelasannya:

1. Konstruktor (`__init__`):

- `temp_mfs` dan `hum_mfs`: Fungsi keanggotaan (Membership Functions) untuk variabel suhu (temperature) dan kelembapan (humidity).
- `rules`: Aturan fuzzy yang akan digunakan dalam sistem.
- `output_ranges`: Rentang output yang akan dihasilkan setelah proses defuzzifikasi.
- Objek Fuzzy:
 - `self.temp_fuzz`: Mengolah data suhu menggunakan fuzzifikasi.
 - `self.hum_fuzz`: Mengolah data kelembapan menggunakan fuzzifikasi.
 - `self.rule_eval`: Evaluasi aturan fuzzy berdasarkan input fuzzy.

2. Metode `calculate`:

- `temp_value` dan `hum_value`: Input suhu dan kelembapan yang diberikan ke sistem.
- Langkah-langkah:

1. Fuzzifikasi:

- `temp_fuzzy` mengubah suhu menjadi nilai fuzzy.
- `hum_fuzzy` mengubah kelembapan menjadi nilai fuzzy.

2. Evaluasi Aturan:

- `rule_results` mengevaluasi aturan fuzzy berdasarkan input fuzzy.

3. Defuzzifikasi:

- Menghasilkan nilai output konkret (misalnya, kecepatan kipas atau pengaturan suhu).

Definisi Fungsi Keanggotaan

Kode ini mendefinisikan fungsi keanggotaan (Membership Functions) untuk variabel input suhu (temp_mfs) dan kelembapan (hum_mfs).

```
[7]: # Definisi Fungsi Keanggotaan
temp_mfs = [
    MembershipFunction("Cold", [0, 0, 20]),
    MembershipFunction("Moderate", [15, 25, 35]),
    MembershipFunction("Hot", [30, 40, 40]),
]

hum_mfs = [
    MembershipFunction("Dry", [0, 0, 50]),
    MembershipFunction("Normal", [25, 50, 75]),
    MembershipFunction("Humid", [50, 100, 100]),
]
```

1. Fungsi Keanggotaan Suhu (temp_mfs):

- Terdiri dari tiga kategori:
 - Cold: [0, 0, 20] — Suhu rendah.
 - Moderate: [15, 25, 35] — Suhu sedang.
 - Hot: [30, 40, 40] — Suhu tinggi.
- Setiap kategori memiliki rentang nilai untuk mendefinisikan derajat keanggotaannya.

2. Fungsi Keanggotaan Kelembapan (hum_mfs):

- Terdiri dari tiga kategori:
 - Dry: [0, 0, 50] — Kelembapan rendah.
 - Normal: [25, 50, 75] — Kelembapan sedang.
 - Humid: [50, 100, 100] — Kelembapan tinggi.
- Setiap kategori juga memiliki rentang nilai.

Aturan Fuzzy, Output Ranges, dan Inisialisasi Sistem

```
# Aturan Fuzzy
rules = [
    ("Cold", "Dry", "Slow"),
    ("Moderate", "Normal", "Medium"),
    ("Hot", "Humid", "Fast"),
]

# Output Ranges (Crisp)
output_ranges = {
    "Slow": 2.5,
    "Medium": 5.0,
    "Fast": 7.5,
}

# Inisialisasi Sistem
fuzzy_system = FuzzySystem(temp_mfs, hum_mfs, rules, output_ranges)
```

3. Inisialisasi Sistem Fuzzy:

- Sistem fuzzy diinisialisasi menggunakan fungsi FuzzySystem() dengan parameter:
 - temp_mfs: Fungsi keanggotaan suhu.
 - hum_mfs: Fungsi keanggotaan kelembapan.
 - rules: Aturan fuzzy yang sudah didefinisikan.
 - output_ranges: Nilai crisp untuk output.

1.⊗ Aturan Fuzzy (rules):

- Program mendefinisikan aturan fuzzy untuk menentukan kecepatan kipas berdasarkan suhu (temperature) dan kelembapan (humidity).
- Contoh aturan:
 - Jika suhu "Cold" dan kelembapan "Dry", maka kecepatan kipas "Slow".
 - Jika suhu "Moderate" dan kelembapan "Normal", maka kecepatan kipas "Medium".
 - Jika suhu "Hot" dan kelembapan "Humid", maka kecepatan kipas "Fast".

2.⊗ Output Ranges (Crisp Values):

- Output sistem dikategorikan menjadi tiga tingkat kecepatan kipas dengan nilai crisp:
 - "Slow" = 2.5
 - "Medium" = 5.0
 - "Fast" = 7.5



Input, Perhitungan Output, dan Hasil

```
# Input
temp_input = 28 # Suhu (°C)
hum_input = 60 # Kelembapan (%)

# Hitung Output
fan_speed = fuzzy_system.calculate(temp_input, hum_input)

# Output
print("Kecepatan Kipas: {:.2f}")
```

Kecepatan Kipas: 5.00

1. Input:

- Suhu (temp_input) = 28°C.
- Kelembapan (hum_input) = 60%.

2. Hitung Output:

- Sistem menghitung kecepatan kipas menggunakan metode fuzzy_system.calculate() dengan input suhu dan kelembapan.

3. Output Kecepatan Kipas:

- Hasil perhitungan adalah kecepatan kipas dengan nilai 5.00.
- Nilai ini termasuk dalam kategori "Medium" sesuai dengan output_ranges.

Visualisasi Fungsi Keanggotaans

```
[14]: # Visualisasi Fungsi Keanggotaan
def plot_membership_functions():
    x_temp = np.linspace(0, 40, 1000)
    x_hum = np.linspace(0, 100, 1000)
    x_speed = np.linspace(0, 10, 1000)

    plt.figure(figsize=(10, 4))
    for mf in temp_mfs:
        mf.plot(x_temp)
    plt.title("Fungsi Keanggotaan Suhu")
    plt.xlabel("Suhu (°C)")
    plt.ylabel("Derajat Keanggotaan")
    plt.legend()
    plt.grid()
    plt.show()

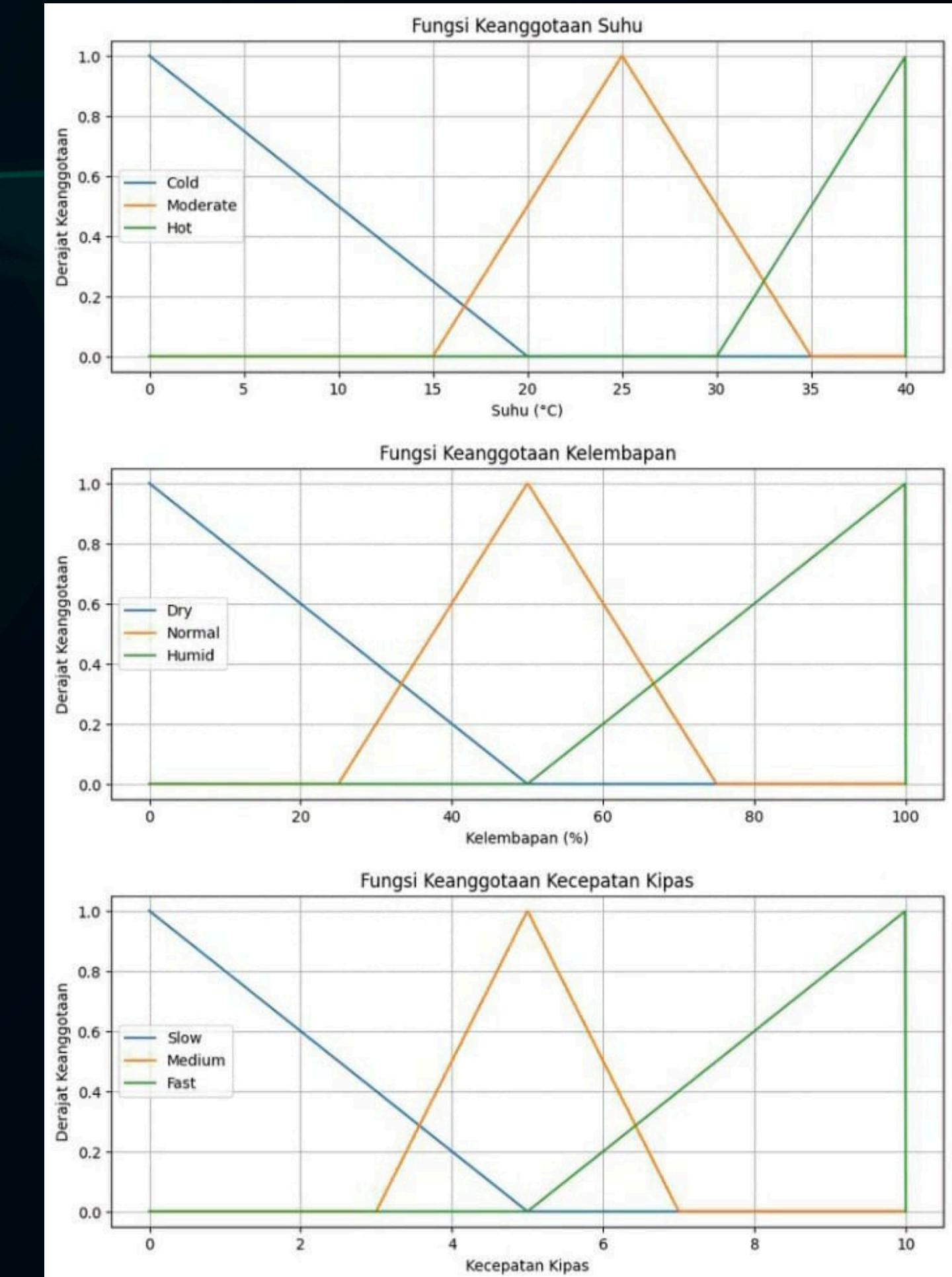
    plt.figure(figsize=(10, 4))
    for mf in hum_mfs:
        mf.plot(x_hum)
    plt.title("Fungsi Keanggotaan Kelembapan")
    plt.xlabel("Kelembapan (%)")
    plt.ylabel("Derajat Keanggotaan")
    plt.legend()
    plt.grid()
    plt.show()

    plt.figure(figsize=(10, 4))
    slow = MembershipFunction("Slow", [0, 0, 5])
    medium = MembershipFunction("Medium", [3, 5, 7])
    fast = MembershipFunction("Fast", [5, 10, 10])
    for mf in [slow, medium, fast]:
        mf.plot(x_speed)
    plt.title("Fungsi Keanggotaan Kecepatan Kipas")
    plt.xlabel("Kecepatan Kipas")
    plt.ylabel("Derajat Keanggotaan")
    plt.legend()
    plt.grid()
    plt.show()

[15]: # Menjalankan Visualisasi
plot_membership_functions()
```

- Fungsi Visualisasi:
 - Fungsi `plot_membership_functions()` digunakan untuk memvisualisasikan fungsi keanggotaan suhu, kelembapan, dan kecepatan kipas.
- Visualisasi Fungsi Keanggotaan:
 - Suhu (Temperature):
 - Fungsi keanggotaan untuk kategori seperti "Cold", "Moderate", dan "Hot".
 - Kelembapan (Humidity):
 - Fungsi keanggotaan untuk kategori seperti "Dry", "Normal", dan "Humid".
 - Kecepatan Kipas (Fan Speed):
 - Fungsi keanggotaan untuk kategori "Slow", "Medium", dan "Fast".
- Menjalankan Visualisasi:
 - Fungsi dipanggil dengan `plot_membership_functions()`, yang menampilkan grafik fungsi keanggotaan.

HASIL DARI MENJALANKAN VISUALISASI



THANK YOU!