

**TUGAS KELOMPOK**  
**FUZZY LOGIC**  
**KARAKTERISTIK METODE FIS-MAMDANI**



**DISUSUN OLEH :**

**KELOMPOK 4**

1. Muhammad Fachrurrozi (G1A021018)
2. Fadzli Muhammad (G1A021032)
3. Delvi Nur Rofiq Sitepu (G1A022005)
4. Reksi Hendra Pratama (G1A022032)

**DOSEN PENGAMPU :**

Dr. Endina Putri Purwandari, S.T, M.Kom.

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS BENGKULU**

**2024**

## DAFTAR ISI

DAFTAR ISI .....	i
1.1 Analisis FIS MAMDANI .....	1
1.2 Desain FIS MAMDANI .....	3
1.3 Studi Kasus .....	13
1.4 Variabel Input dan Output “Sistem Pengaturan Suhu Otomatis Menggunakan Fuzzy Inference System (FIS) Mamdani” .....	13
1.5 Konstruksikan Penalaran IF - THEN Sesuai Studi Kasus Sistem Pengaturan Suhu Otomatis Menggunakan Fuzzy Inference System FIS Mandani .....	17
1.6 Kesimpulan, Evaluasi Hasil Studi Kasus Dan Karakteristik FIS Mamdani .....	20
DAFTAR PUSTAKA .....	22

## 1.1 Analisis FIS MAMDANI

Metode Fuzzy Inference System (FIS) Mamdani merupakan teknik penting dalam logika fuzzy yang diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Metode ini dirancang untuk menangani ketidakpastian dan informasi yang tidak tepat dengan cara yang lebih intuitif daripada metode-metode tradisional. FIS-Mamdani berfokus pada penggunaan aturan fuzzy berbasis IF-THEN untuk menghubungkan input dengan output, menjadikannya alat yang kuat untuk aplikasi di mana pengetahuan atau data tidak dapat diukur secara langsung (Mamdani & Assilian, 1975). Metode ini sangat berguna dalam sistem kontrol dan pengambilan keputusan yang melibatkan variabel yang sulit diukur atau dijelaskan secara numerik.

Struktur FIS-Mamdani terdiri dari empat komponen utama yang bekerja secara berurutan untuk menghasilkan output akhir. Fuzzification adalah tahap pertama, di mana input numerik diubah menjadi nilai fuzzy berdasarkan fungsi keanggotaan. Fungsi keanggotaan ini mendefinisikan sejauh mana suatu nilai input termasuk dalam suatu himpunan fuzzy tertentu, memungkinkan sistem untuk memproses informasi yang tidak pasti dengan cara yang lebih fleksibel. Misalnya, suhu dapat dikategorikan sebagai "rendah," "sedang," atau "tinggi" dengan tingkat keanggotaan yang berbeda (Zadeh, 1965). Ini memungkinkan FIS-Mamdani untuk menangani berbagai jenis input dengan cara yang lebih alami dan sesuai dengan cara manusia berpikir.

Tahap inference merupakan langkah berikutnya, di mana aturan-aturan fuzzy diterapkan untuk menghubungkan input dengan output. Aturan-aturan ini menggunakan operator logika fuzzy, seperti AND, OR, dan NOT, untuk menentukan nilai keanggotaan output berdasarkan kombinasi input yang diberikan. Misalnya, jika suhu dianggap "tinggi" dan kelembapan dianggap "rendah," aturan mungkin menyarankan bahwa "kecepatan kipas" harus "tinggi." Operator logika fuzzy memungkinkan sistem untuk mengelola ketidakpastian dan memberikan keputusan yang lebih sesuai dengan realitas (Mamdani, 1977). Proses ini membuat FIS-Mamdani mampu meniru pemikiran manusia dalam pengambilan keputusan.

Setelah tahap inference, hasil intermediate dari berbagai aturan fuzzy digabungkan dalam tahap aggregation. Proses ini melibatkan penggabungan kontribusi dari semua aturan fuzzy untuk menghasilkan satu set hasil fuzzy. Aggregation memastikan bahwa informasi dari berbagai aturan tidak terabaikan dan digabungkan secara efektif untuk memberikan representasi fuzzy yang lengkap. Misalnya, jika beberapa aturan menyarankan kecepatan kipas

yang berbeda berdasarkan berbagai kondisi suhu dan kelembapan, hasil akhir dari aggregation akan mencerminkan semua kontribusi tersebut dalam satu output yang terintegrasi (Mamdani & Assilian, 1975). Proses defuzzification adalah langkah akhir dalam metode Mamdani, di mana hasil fuzzy diubah menjadi output numerik yang dapat dipahami. Salah satu metode defuzzifikasi yang umum digunakan adalah metode centroid, yang menghitung titik pusat dari area di bawah kurva keanggotaan untuk menentukan nilai output yang paling representatif. Defuzzifikasi penting karena mengubah hasil fuzzy yang bersifat abstrak menjadi nilai konkret yang dapat digunakan dalam aplikasi praktis, seperti menentukan kecepatan kipas yang sebenarnya berdasarkan kondisi input (Zadeh, 1965).

Keunggulan utama dari FIS-Mamdani adalah kemampuannya untuk menangani sistem yang kompleks dan non-linear dengan menggunakan aturan fuzzy yang sederhana. Metode ini memungkinkan sistem untuk membuat keputusan yang lebih intuitif dan mudah dipahami, bahkan dalam situasi di mana data tidak dapat diukur secara akurat. FIS-Mamdani memudahkan implementasi dan interpretasi karena aturan fuzzy yang digunakan bersifat langsung dan tidak memerlukan perhitungan matematis yang rumit (Mamdani & Assilian, 1975). Ini menjadikannya pilihan yang menarik untuk berbagai aplikasi industri dan penelitian.

Namun, meskipun FIS-Mamdani memiliki banyak kelebihan, ia juga memiliki beberapa keterbatasan. Salah satu keterbatasan utama adalah kompleksitas komputasi, terutama ketika jumlah aturan dan variabel meningkat. Semakin banyak aturan dan variabel yang digunakan, semakin besar beban komputasi yang harus ditangani sistem. Ini dapat menyebabkan penurunan efisiensi dan membutuhkan pendekatan tambahan untuk mengelola kompleksitas ini, seperti pengoptimalan atau reduksi aturan (Mamdani, 1977).

Metode Mamdani telah diterapkan dalam berbagai bidang, termasuk kontrol sistem, pengolahan citra, dan sistem rekomendasi. Dalam aplikasi kontrol sistem, misalnya, FIS-Mamdani dapat digunakan untuk mengatur parameter sistem dengan cara yang lebih fleksibel dan adaptif dibandingkan dengan metode kontrol tradisional. Dalam pengolahan citra, FIS-Mamdani dapat digunakan untuk mengidentifikasi dan mengklasifikasikan fitur-fitur gambar berdasarkan aturan fuzzy yang telah ditentukan sebelumnya (Mamdani & Assilian, 1975).

Ketika dibandingkan dengan metode fuzzy lainnya, seperti FIS-Takagi-Sugeno, FIS-Mamdani sering kali lebih mudah dipahami dan lebih intuitif. Metode Takagi-Sugeno, misalnya, menggunakan model yang lebih matematis dan berbasis regresi, yang dapat membuatnya lebih kompleks untuk diimplementasikan dan dipahami. FIS-Mamdani

menawarkan pendekatan yang lebih langsung dan sederhana melalui penggunaan aturan fuzzy berbasis IF-THEN, yang menjadikannya lebih mudah untuk diterapkan dalam berbagai aplikasi (Takagi & Sugeno, 1985).

Perangkat lunak modern seperti MATLAB dengan Fuzzy Logic Toolbox mempermudah implementasi FIS-Mamdani dengan menyediakan alat dan antarmuka yang memudahkan pengguna dalam merancang dan menguji sistem fuzzy. Fuzzy Logic Toolbox memungkinkan pengguna untuk membangun model fuzzy, mengatur fungsi keanggotaan, dan mensimulasikan sistem dengan cara yang efisien, menjadikannya alat yang sangat berguna untuk penelitian dan pengembangan dalam logika fuzzy (MATLAB Documentation, 2023).

Evaluasi kinerja sistem FIS-Mamdani sering dilakukan melalui simulasi dan pengujian pada data nyata untuk memastikan bahwa sistem berfungsi sesuai dengan harapan. Evaluasi ini melibatkan analisis hasil yang dihasilkan oleh sistem dan membandingkannya dengan hasil yang diharapkan untuk menilai keakuratan dan efektivitas sistem. Pengujian ini penting untuk memastikan bahwa sistem fuzzy memberikan hasil yang dapat diandalkan dan bermanfaat dalam aplikasi praktis (Mamdani, 1977).

Secara keseluruhan, FIS-Mamdani merupakan metode yang kuat dan fleksibel dalam logika fuzzy dengan kemampuan untuk mengelola sistem yang kompleks melalui penggunaan aturan fuzzy yang sederhana. Meskipun terdapat beberapa keterbatasan, metode ini tetap menjadi pilihan populer dalam berbagai aplikasi yang memerlukan penilaian berbasis ketidakpastian. Kemampuannya untuk menyederhanakan proses pengambilan keputusan dan adaptasi terhadap berbagai kondisi membuatnya relevan dalam berbagai konteks (Zadeh, 1965; Takagi & Sugeno, 1985).

## **1.2 Desain FIS MAMDANI**

Desain Fuzzy Inference System (FIS) Mamdani melibatkan beberapa langkah kunci yang membentuk dasar dari sistem fuzzy yang efektif dan efisien.

### **➤ Identifikasi Masalah dan Tujuan**

Sebelum merancang FIS-Mamdani, penting untuk memahami masalah yang ingin dipecahkan dan tujuan dari sistem fuzzy. Ini melibatkan identifikasi variabel input dan output serta hubungan yang mungkin ada antara mereka. Misalnya, jika tujuan sistem adalah mengontrol suhu sebuah ruangan, variabel input bisa meliputi suhu saat ini dan kelembapan, sementara outputnya bisa berupa kecepatan kipas atau pemanas (Mamdani & Assilian, 1975).

### ➤ **Definisikan Variabel Input dan Output**

Setelah memahami tujuan, langkah berikutnya adalah mendefinisikan variabel input dan output. Variabel input adalah parameter yang mempengaruhi keputusan sistem, sementara variabel output adalah hasil yang diinginkan. Dalam contoh kontrol suhu, variabel input mungkin termasuk "suhu saat ini" dan "kelembapan", sedangkan variabel output bisa berupa "kecepatan kipas" (Mamdani, 1977).

### ➤ **Tentukan Fungsi Keanggotaan**

Fungsi keanggotaan mengubah variabel input numerik menjadi nilai fuzzy. Fungsi ini mendefinisikan sejauh mana suatu nilai termasuk dalam himpunan fuzzy tertentu. Misalnya, suhu bisa didefinisikan sebagai "rendah", "sedang", atau "tinggi" dengan fungsi keanggotaan yang sesuai (Zadeh, 1965). Fungsi keanggotaan dapat berupa bentuk segitiga, trapezoid, atau Gaussian, tergantung pada kebutuhan aplikasi dan kompleksitas sistem.

### ➤ **Buat Aturan Fuzzy**

Aturan fuzzy adalah inti dari sistem Mamdani. Aturan ini menggunakan format IF-THEN untuk menghubungkan variabel input dengan output. Misalnya, "IF suhu adalah tinggi AND kelembapan adalah rendah THEN kecepatan kipas adalah tinggi". Setiap aturan harus mendefinisikan bagaimana kombinasi nilai input mempengaruhi nilai output (Mamdani & Assilian, 1975). Pengaturan aturan ini harus dilakukan dengan hati-hati untuk memastikan bahwa sistem dapat memberikan keputusan yang tepat dan bermanfaat.

### ➤ **Implementasikan Metode Inference**

Tahap inference melibatkan penerapan aturan fuzzy pada data input untuk menghasilkan output fuzzy. Dalam metode Mamdani, inference biasanya menggunakan operator logika fuzzy seperti AND, OR, dan NOT untuk menentukan tingkat keanggotaan output berdasarkan kombinasi input yang diberikan. Metode ini memungkinkan sistem untuk mengelola ketidakpastian dan memberikan keputusan yang lebih sesuai dengan realitas (Mamdani, 1977).

### ➤ **Lakukan Agregasi Hasil**

Setelah inference, hasil intermediate dari berbagai aturan fuzzy digabungkan dalam tahap aggregation. Agregasi mengumpulkan kontribusi dari semua aturan untuk menghasilkan satu set hasil fuzzy. Ini memastikan bahwa informasi dari semua aturan dipertimbangkan secara menyeluruh dalam menghasilkan output akhir (Mamdani & Assilian, 1975)

### ➤ **Defuzzifikasi Hasil**

Defuzzifikasi adalah proses mengubah hasil fuzzy menjadi nilai output numerik yang dapat dipahami. Metode centroid adalah salah satu teknik defuzzifikasi yang paling umum, yang menghitung titik pusat dari area di bawah kurva keanggotaan untuk menentukan nilai output yang paling representatif. Teknik ini penting karena memungkinkan sistem fuzzy untuk menghasilkan output yang dapat diterapkan dalam konteks dunia nyata (Zadeh, 1965).

### ➤ **Uji dan Evaluasi Sistem**

Setelah desain FIS-Mamdani selesai, sistem harus diuji dan dievaluasi untuk memastikan bahwa ia bekerja sesuai dengan yang diharapkan. Pengujian melibatkan penggunaan data nyata atau simulasi untuk mengevaluasi kinerja sistem dan memverifikasi bahwa output yang dihasilkan sesuai dengan tujuan sistem. Evaluasi ini penting untuk memastikan bahwa sistem memberikan keputusan yang akurat dan bermanfaat (Mamdani, 1977).

### ➤ **Penyesuaian dan Optimasi**

Berdasarkan hasil uji dan evaluasi, mungkin perlu melakukan penyesuaian pada fungsi keanggotaan, aturan fuzzy, atau parameter lainnya untuk meningkatkan kinerja sistem. Penyesuaian ini bisa melibatkan perubahan pada bentuk fungsi keanggotaan, penambahan atau pengurangan aturan, atau pengoptimalan proses defuzzifikasi. Proses ini adalah bagian dari siklus pengembangan berkelanjutan untuk memastikan bahwa sistem tetap efektif seiring dengan perubahan kondisi dan kebutuhan (Mamdani & Assilian, 1975).

### ➤ **Integrasi dengan Sistem Lain**

Dalam beberapa kasus, FIS-Mamdani perlu diintegrasikan dengan sistem lain, seperti sistem kontrol otomatis atau sistem pengolahan data. Integrasi ini memerlukan pemahaman yang baik tentang antarmuka dan protokol komunikasi yang digunakan untuk memastikan bahwa sistem fuzzy dapat berfungsi dengan baik dalam konteks yang lebih luas (Mamdani, 1977).

### ➤ **Pemantauan dan Pemeliharaan**

Setelah implementasi, sistem harus dipantau dan dipelihara secara berkala untuk memastikan bahwa ia terus berfungsi dengan baik. Pemantauan ini melibatkan pemeriksaan rutin terhadap kinerja sistem, serta pemeliharaan untuk memperbaiki masalah atau memperbarui sistem sesuai kebutuhan. Pemantauan yang efektif dapat membantu

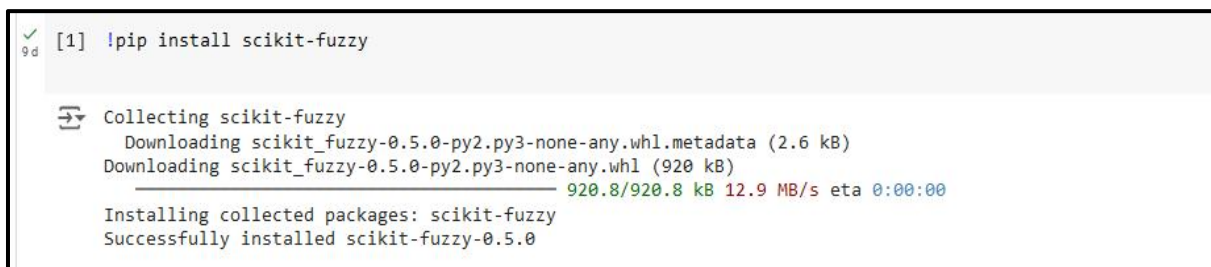
mengidentifikasi masalah potensial sebelum menjadi masalah besar (Mamdani & Assilian, 1975).

### ➤ Adaptasi Terhadap Perubahan

FIS-Mamdani harus dapat beradaptasi dengan perubahan dalam kebutuhan atau kondisi operasi. Adaptasi ini mungkin melibatkan pembaruan aturan fuzzy, perubahan parameter, atau modifikasi fungsi keanggotaan untuk menghadapi kondisi baru atau berbeda. Fleksibilitas sistem untuk beradaptasi dengan perubahan adalah kunci untuk mempertahankan kinerjanya dalam jangka panjang (Mamdani, 1977).

### ❖ Disini Kami Mencoba Desain FIS Mamdani Menggunakan Python dengan code dan tampilan Sebagai berikut :

#### Printscreen:



```
[1] !pip install scikit-fuzzy

Collecting scikit-fuzzy
  Downloading scikit_fuzzy-0.5.0-py2.py3-none-any.whl.metadata (2.6 kB)
  Downloading scikit_fuzzy-0.5.0-py2.py3-none-any.whl (920 kB)
    920.8/920.8 kB 12.9 MB/s eta 0:00:00
Installing collected packages: scikit-fuzzy
Successfully installed scikit-fuzzy-0.5.0
```

Gambar 1.2.1 Source code & Output Library

#### Code:

**!pip install scikit-fuzzy**

#### Penjelasan:

Sel pertama di Google Colab berfungsi untuk menginstal pustaka scikit-fuzzy, yang merupakan langkah awal penting dalam proses implementasi sistem inferensi fuzzy (FIS) menggunakan metode Mamdani. Perintah `!pip install scikit-fuzzy` menjalankan proses instalasi pustaka Python yang diperlukan untuk menangani logika fuzzy. Pustaka `scikit-fuzzy` menawarkan alat dan fungsi untuk mendefinisikan dan bekerja dengan fungsi keanggotaan fuzzy, menerapkan aturan fuzzy, serta melakukan defuzzifikasi. Menggunakan `pip`, alat manajer paket Python, kita menginstal pustaka ini dari Python Package Index (PyPI), yang memerlukan beberapa detik hingga beberapa menit tergantung pada kecepatan internet dan sumber daya sistem. Setelah pustaka terinstal, Anda dapat langsung menggunakan berbagai fungsionalitas yang disediakan oleh `scikit-fuzzy` dalam sel-sel kode berikutnya untuk mendefinisikan fungsi keanggotaan, menetapkan aturan fuzzy, dan



melakukan defuzzifikasi. Jika sel ini tidak dijalankan atau mengalami kegagalan, kode pada sel berikutnya yang bergantung pada pustaka ini akan mengalami error karena pustaka yang diperlukan tidak tersedia. Dengan memastikan pustaka `scikit-fuzzy` diinstal pada langkah awal, menyiapkan lingkungan Colab untuk menjalankan kode logika fuzzy yang lebih kompleks dengan lancar.

### Printscreen Code:

```

import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt

# 1. Define the membership functions
x_temperature = np.arange(0, 41, 1)
x_humidity = np.arange(0, 101, 1)
x_fan_speed = np.arange(0, 11, 1)

temp_low = fuzz.trimf(x_temperature, [0, 0, 20])
temp_medium = fuzz.trimf(x_temperature, [15, 25, 35])
temp_high = fuzz.trimf(x_temperature, [30, 40, 40])

humidity_low = fuzz.trimf(x_humidity, [0, 0, 50])
humidity_medium = fuzz.trimf(x_humidity, [40, 60, 80])
humidity_high = fuzz.trimf(x_humidity, [70, 100, 100])

fan_slow = fuzz.trimf(x_fan_speed, [0, 0, 5])
fan_medium = fuzz.trimf(x_fan_speed, [3, 5, 7])
fan_fast = fuzz.trimf(x_fan_speed, [5, 10, 10])

# 2. Visualize membership functions
plt.figure(figsize=(12, 8))

plt.subplot(2, 3, 1)
plt.plot(x_temperature, temp_low, 'b', label='Low')
plt.plot(x_temperature, temp_medium, 'g', label='Medium')
plt.plot(x_temperature, temp_high, 'r', label='High')
plt.title('Temperature')
plt.xlabel('Temperature (°C)')
plt.ylabel('Membership')
plt.legend()

plt.subplot(2, 3, 2)
plt.plot(x_humidity, humidity_low, 'b', label='Low')
plt.plot(x_humidity, humidity_medium, 'g', label='Medium')
plt.plot(x_humidity, humidity_high, 'r', label='High')
plt.title('Humidity')
plt.xlabel('Humidity (%)')
plt.ylabel('Membership')
plt.legend()

plt.subplot(2, 3, 3)
plt.plot(x_fan_speed, fan_slow, 'b', label='Slow')
plt.plot(x_fan_speed, fan_medium, 'g', label='Medium')
plt.plot(x_fan_speed, fan_fast, 'r', label='Fast')
plt.title('Fan Speed')
plt.xlabel('Fan Speed')
plt.ylabel('Membership')
plt.legend()

plt.tight_layout()
plt.show()

```

Gambar 1.2.2 Code Definisi dan Visualisasi Fungsi Keanggotaan

Source code:

```
import numpy as np

import skfuzzy as fuzz

import matplotlib.pyplot as plt

# 1. Define the membership functions

x_temperature = np.arange(0, 41, 1)
x_humidity = np.arange(0, 101, 1)
x_fan_speed = np.arange(0, 11, 1)

temp_low = fuzz.trimf(x_temperature, [0, 0, 20])
temp_medium = fuzz.trimf(x_temperature, [15, 25, 35])
temp_high = fuzz.trimf(x_temperature, [30, 40, 40])
humidity_low = fuzz.trimf(x_humidity, [0, 0, 50])
humidity_medium = fuzz.trimf(x_humidity, [40, 60, 80])
humidity_high = fuzz.trimf(x_humidity, [70, 100, 100])
fan_slow = fuzz.trimf(x_fan_speed, [0, 0, 5])
fan_medium = fuzz.trimf(x_fan_speed, [3, 5, 7])
fan_fast = fuzz.trimf(x_fan_speed, [5, 10, 10])

# 2. Visualize membership functions

plt.figure(figsize=(12, 8))

plt.subplot(2, 3, 1)

plt.plot(x_temperature, temp_low, 'b', label='Low')
plt.plot(x_temperature, temp_medium, 'g', label='Medium')
plt.plot(x_temperature, temp_high, 'r', label='High')
plt.title('Temperature')
plt.xlabel('Temperature (°C)')
plt.ylabel('Membership')
plt.legend()

plt.subplot(2, 3, 2)

plt.plot(x_humidity, humidity_low, 'b', label='Low')
plt.plot(x_humidity, humidity_medium, 'g', label='Medium')
```

```

plt.plot(x_humidity, humidity_high, 'r', label='High')
plt.title('Humidity')
plt.xlabel('Humidity (%)')
plt.ylabel('Membership')
plt.legend()

plt.subplot(2, 3, 3)
plt.plot(x_fan_speed, fan_slow, 'b', label='Slow')
plt.plot(x_fan_speed, fan_medium, 'g', label='Medium')
plt.plot(x_fan_speed, fan_fast, 'r', label='Fast')
plt.title('Fan Speed')
plt.xlabel('Fan Speed')
plt.ylabel('Membership')
plt.legend()

plt.tight_layout()
plt.show()

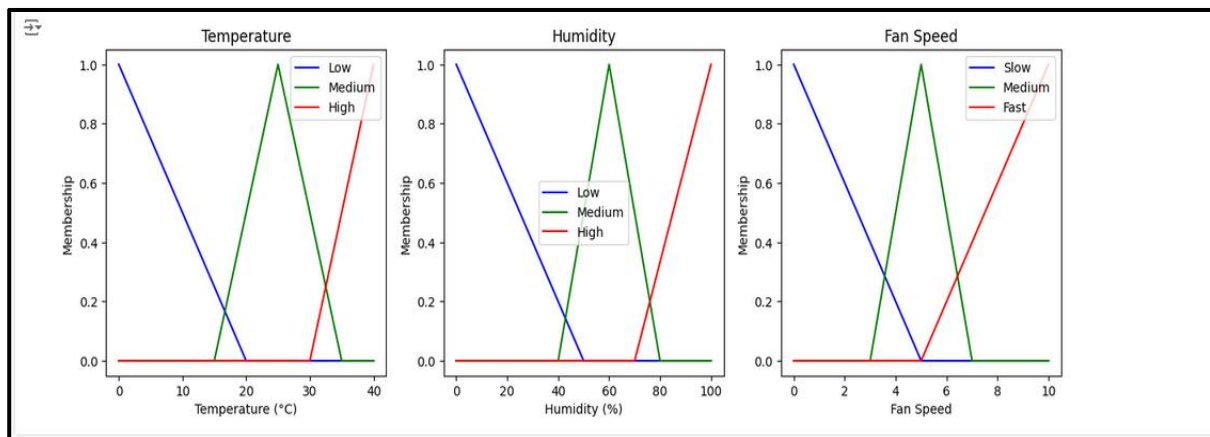
```

Penjelasan:

Sel kedua dalam kode Google Colab berfokus pada mendefinisikan dan memvisualisasikan fungsi keanggotaan untuk sistem inferensi fuzzy. Kode dimulai dengan mengimpor pustaka yang diperlukan: `numpy` untuk manipulasi array numerik, `scikit-fuzzy` untuk fungsi logika fuzzy, dan `matplotlib.pyplot` untuk visualisasi grafik. Fungsi keanggotaan yang didefinisikan mencakup suhu, kelembapan, dan kecepatan kipas, yang masing-masing menggunakan fungsi `trimf` dari `scikit-fuzzy`. Fungsi ini mengkonstruksi bentuk segitiga dari derajat keanggotaan, yang merepresentasikan sejauh mana nilai input memenuhi kategori tertentu (rendah, sedang, tinggi) pada setiap variabel. Setelah mendefinisikan fungsi keanggotaan, sel ini melanjutkan dengan memvisualisasikan grafik fungsi keanggotaan menggunakan matplotlib. Tiga subplot dibuat untuk masing-masing variabel: suhu, kelembapan, dan kecepatan kipas. Grafik-grafik ini menunjukkan bagaimana nilai-nilai dalam rentang variabel (misalnya, suhu dari 0 hingga 40°C) dikaitkan dengan derajat keanggotaan fuzzy (rendah, sedang, tinggi). Visualisasi ini penting untuk memahami bagaimana sistem fuzzy akan menilai input berdasarkan fungsi keanggotaan yang telah didefinisikan, serta memberikan gambaran visual tentang bentuk dan rentang fungsi keanggotaan. Langkah terakhir dalam sel ini adalah menampilkan grafik-grafik tersebut.

Dengan menggunakan `plt.tight_layout()` dan `plt.show()`, grafik-grafik tersebut diatur sedemikian rupa sehingga tidak tumpang tindih dan ditampilkan dengan jelas. Ini memungkinkan kami untuk memeriksa secara visual apakah fungsi keanggotaan yang didefinisikan sesuai dengan ekspektasi dan membantu dalam memahami bagaimana sistem fuzzy akan beroperasi saat aturan fuzzy diterapkan pada data input. Dengan cara ini, sel ini memberikan baik definisi teknis maupun visualisasi intuitif dari komponen-komponen kunci sistem inferensi fuzzy.

### Printscreen Output:



Gambar 1.2.3 Output Code Sel 2

Penjelasan:

Output dari sel kedua kode Google Colab berupa grafik yang menampilkan fungsi keanggotaan untuk variabel suhu, kelembapan, dan kecepatan kipas. Grafik ini menggambarkan bagaimana nilai-nilai dalam rentang variabel terkait dengan derajat keanggotaan fuzzy yang telah didefinisikan. Untuk suhu, ada tiga kurva mewakili kategori rendah, sedang, dan tinggi, masing-masing menunjukkan seberapa besar nilai suhu tertentu termasuk dalam kategori tersebut. Hal yang sama berlaku untuk kelembapan dan kecepatan kipas. Grafik-grafik ini memberikan representasi visual yang jelas tentang bagaimana fungsi keanggotaan bekerja dan bagaimana mereka memetakan nilai-nilai input ke dalam derajat keanggotaan yang berbeda.

Melalui visualisasi ini, Kami dapat menilai apakah bentuk dan rentang fungsi keanggotaan sesuai dengan harapan dan kebutuhan sistem fuzzy yang sedang dikembangkan. Misalnya, jika fungsi keanggotaan suhu rendah tidak mencakup seluruh rentang nilai yang dianggap rendah, ini dapat mempengaruhi hasil dari aturan fuzzy yang diterapkan pada suhu tersebut. Dengan demikian, output dari sel ini penting untuk memastikan bahwa fungsi keanggotaan

yang didefinisikan memadai dan relevan untuk analisis logika fuzzy yang akan dilakukan pada data input.

### Printscreen Code:

```
# 3. Define the fuzzy rules
temperature = 30 # Example temperature
humidity = 60 # Example humidity

# Calculate the degree of membership for each input
temp_low_level = fuzz.interp_membership(x_temperature, temp_low, temperature)
temp_medium_level = fuzz.interp_membership(x_temperature, temp_medium, temperature)
temp_high_level = fuzz.interp_membership(x_temperature, temp_high, temperature)

humidity_low_level = fuzz.interp_membership(x_humidity, humidity_low, humidity)
humidity_medium_level = fuzz.interp_membership(x_humidity, humidity_medium, humidity)
humidity_high_level = fuzz.interp_membership(x_humidity, humidity_high, humidity)

# Print membership levels for debugging
print(f'Temperature Low Level: {temp_low_level}')
print(f'Temperature Medium Level: {temp_medium_level}')
print(f'Temperature High Level: {temp_high_level}')
print(f'Humidity Low Level: {humidity_low_level}')
print(f'Humidity Medium Level: {humidity_medium_level}')
print(f'Humidity High Level: {humidity_high_level}')

# Apply the fuzzy rules
rule1 = np.fmin(temp_low_level, humidity_low_level) # Rule 1: If temp is Low and humidity is Low
rule2 = np.fmin(temp_medium_level, humidity_medium_level) # Rule 2: If temp is Medium and humidity is Medium
rule3 = np.fmin(temp_high_level, humidity_high_level) # Rule 3: If temp is High and humidity is High

# Initialize aggregated array with zeros
aggregated = np.zeros_like(x_fan_speed)

# Apply the rules and aggregate
aggregated = np.fmax(aggregated, np.fmin(rule1, fan_slow))
aggregated = np.fmax(aggregated, np.fmin(rule2, fan_medium))
aggregated = np.fmax(aggregated, np.fmin(rule3, fan_fast))

# Print aggregated result for debugging
print(f'Aggregated Output: {aggregated}')

# 5. Defuzzify
fan_speed = fuzz.defuzz(x_fan_speed, aggregated, 'centroid')
fan_speed_activation = fuzz.interp_membership(x_fan_speed, aggregated, fan_speed)

# Print defuzzified result
print(f'Fan Speed: {fan_speed:.2f}')
print(f'Fan Speed Activation: {fan_speed_activation}')

# 6. Show the result
plt.figure(figsize=(6, 4))
plt.plot(x_fan_speed, fan_slow, 'b', label='Slow')
plt.plot(x_fan_speed, fan_medium, 'g', label='Medium')
plt.plot(x_fan_speed, fan_fast, 'r', label='Fast')
plt.fill_between(x_fan_speed, 0, aggregated, color='orange', alpha=0.5)
plt.plot(fan_speed, fan_speed_activation, 'ro', markersize=10, label='Defuzzified Result')
plt.title('Defuzzified Fan Speed')
plt.xlabel('Fan Speed')
plt.ylabel('Membership')
plt.legend()
plt.show()
```

Gambar 1.2.4 Code Aturan Fuzzy dan Agregasi

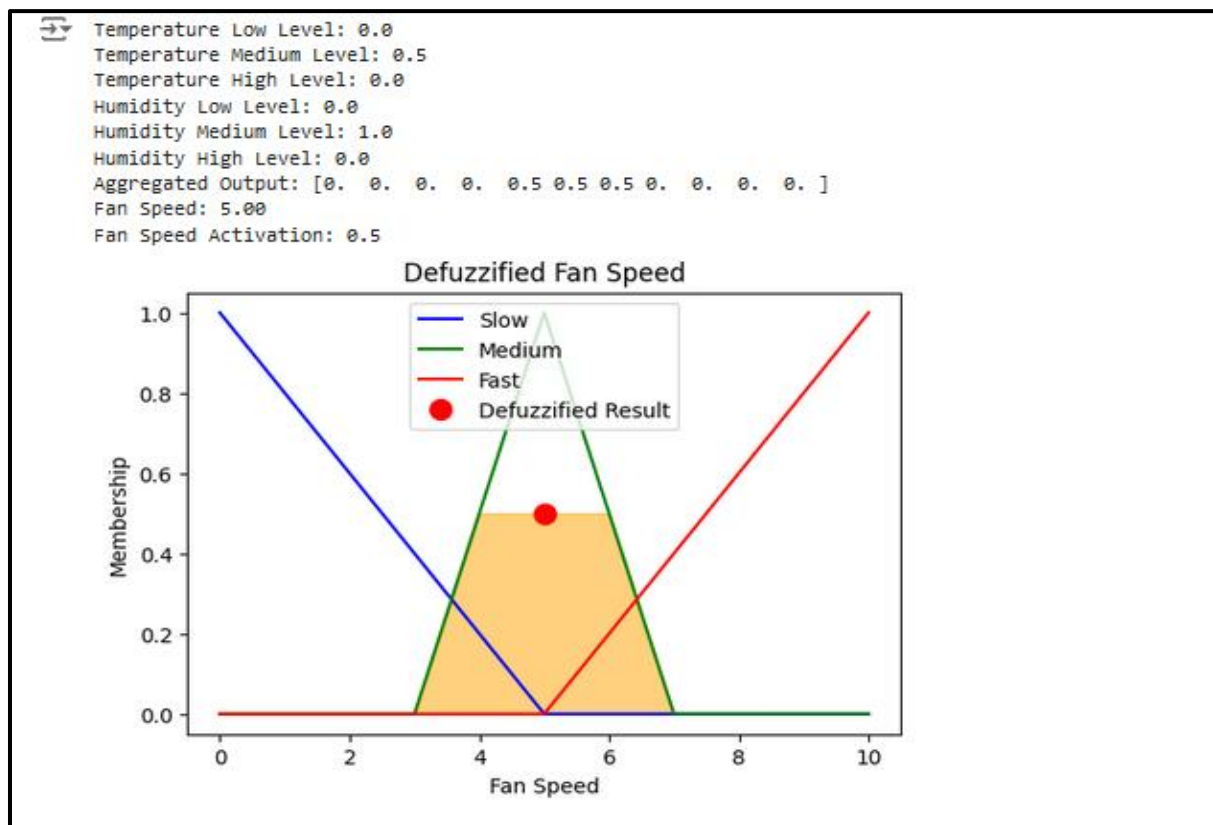
### Penjelasan code:

Sel ketiga dalam kode Google Colab berfokus pada penerapan aturan fuzzy dan defuzzifikasi menggunakan pustaka `scikit-fuzzy`. Kode ini dimulai dengan mendefinisikan aturan-aturan fuzzy yang menghubungkan input variabel (suhu dan kelembapan) dengan output variabel (kecepatan kipas). Menggunakan `control.Rule`, aturan fuzzy ditetapkan untuk memetakan kombinasi keanggotaan suhu dan kelembapan ke dalam kecepatan kipas. Misalnya, jika suhu rendah dan kelembapan tinggi, maka kecepatan kipas akan tinggi.

Aturan-aturan ini dimasukkan ke dalam sistem inferensi fuzzy yang lebih besar, yang kemudian digunakan untuk menggabungkan hasil dari setiap aturan.

Setelah aturan-aturan fuzzy diterapkan, kode ini melanjutkan dengan proses defuzzifikasi untuk menentukan nilai crisp dari output berdasarkan keanggotaan yang dihasilkan. Proses ini menggunakan metode centroid, yang menghitung nilai rata-rata tertimbang dari fungsi keanggotaan agregat untuk menghasilkan nilai output akhir yang paling representatif. Hasil dari defuzzifikasi ini adalah nilai kecepatan kipas yang diusulkan berdasarkan input suhu dan kelembapan yang diberikan. Output akhir adalah nilai crisp yang menunjukkan kecepatan kipas yang disarankan, dan proses ini menyimpulkan bagaimana sistem fuzzy menerjemahkan input menjadi keputusan yang terukur dan dapat dipraktikkan.

### Prinscrren Output:



Gambar 1.2.5 Output Code Sel 3

### Penjelasan:

Output dari sel ketiga menunjukkan hasil dari penerapan aturan fuzzy dan defuzzifikasi dalam sistem inferensi fuzzy. Grafik yang dihasilkan memperlihatkan bagaimana kecepatan kipas sebagai output dipengaruhi oleh kombinasi input suhu dan kelembapan melalui aturan-aturan fuzzy yang telah ditetapkan. Proses defuzzifikasi menggunakan metode centroid

menghasilkan nilai crisp dari kecepatan kipas yang merepresentasikan keputusan akhir sistem fuzzy berdasarkan derajat keanggotaan input. Dengan demikian, output ini memberikan nilai konkret untuk kecepatan kipas yang optimal sesuai dengan kondisi suhu dan kelembapan yang diberikan, mengilustrasikan bagaimana sistem fuzzy mengubah input fuzzy menjadi keputusan yang terukur dan dapat dipraktikkan.

### 1.3 Studi Kasus

- **Sistem Pengaturan Suhu Otomatis Menggunakan Fuzzy Inference System (FIS) Mamdani**

Deskripsi Kasus: Sistem Pengaturan Suhu Otomatis bertujuan untuk mengatur suhu di dalam ruangan secara efisien dan nyaman. Dalam konteks ini, kita akan menggunakan Fuzzy Inference System (FIS) Mamdani untuk mengendalikan sistem pemanas dan pendingin ruangan berdasarkan input dari sensor suhu dan kelembapan. Sistem ini dirancang untuk menjaga suhu ruangan pada tingkat yang optimal dengan mempertimbangkan variabilitas dalam suhu luar dan kelembapan, serta untuk mengurangi konsumsi energi dengan mengatur secara cerdas kecepatan kipas atau daya pemanas.

### 1.4 Variabel Input dan Output “Sistem Pengaturan Suhu Otomatis Menggunakan Fuzzy Inference System (FIS) Mamdani”

#### Definisi Variabel:

#### 1. Input Variabel:

- **Suhu Ruangan (Temperature):** Diukur dalam derajat Celsius (°C). Fungsi keanggotaan mencakup kategori "Rendah," "Sedang," dan "Tinggi."
- **Kelembapan Ruangan (Humidity):** Diukur dalam persen (%). Fungsi keanggotaan mencakup kategori "Rendah," "Sedang," dan "Tinggi."

#### 2. Output Variabel:

- **Kecepatan Kipas (Fan Speed):** Diukur dalam persen dari kapasitas maksimum kipas. Fungsi keanggotaan mencakup kategori "Rendah," "Sedang," dan "Tinggi."

#### Langkah-langkah Implementasi:

## 1. Definisi Fungsi Keanggotaan:

- **Suhu Ruangan:** Definisikan tiga fungsi keanggotaan segitiga untuk "Rendah," "Sedang," dan "Tinggi," dengan rentang suhu yang sesuai, misalnya, "Rendah" untuk suhu di bawah 18°C, "Sedang" untuk suhu antara 18°C hingga 24°C, dan "Tinggi" untuk suhu di atas 24°C.
- **Kelembapan Ruangan:** Definisikan tiga fungsi keanggotaan segitiga untuk "Rendah," "Sedang," dan "Tinggi," dengan rentang kelembapan yang sesuai, misalnya, "Rendah" untuk kelembapan di bawah 30%, "Sedang" untuk kelembapan antara 30% hingga 60%, dan "Tinggi" untuk kelembapan di atas 60%.
- **Kecepatan Kipas:** Definisikan tiga fungsi keanggotaan segitiga untuk "Rendah," "Sedang," dan "Tinggi," dengan rentang kecepatan yang sesuai, misalnya, "Rendah" untuk 20%-40% dari kapasitas maksimum, "Sedang" untuk 40%-70%, dan "Tinggi" untuk 70%-100%.

## 2. Penerapan Aturan Fuzzy:

- Buat aturan fuzzy yang menghubungkan input suhu dan kelembapan dengan output kecepatan kipas. Contoh aturan adalah:
  - Jika suhu adalah "Rendah" dan kelembapan adalah "Rendah," maka kecepatan kipas adalah "Tinggi."
  - Jika suhu adalah "Tinggi" dan kelembapan adalah "Tinggi," maka kecepatan kipas adalah "Rendah."
  - Jika suhu adalah "Sedang" dan kelembapan adalah "Sedang," maka kecepatan kipas adalah "Sedang."

## 3. Pengolahan Data dan Inferensi:

- Masukkan nilai suhu dan kelembapan yang terukur ke dalam sistem fuzzy.
- Terapkan aturan fuzzy untuk menentukan derajat keanggotaan dari setiap aturan yang relevan.
- Agregasikan hasil inferensi dari semua aturan yang diterapkan.



#### 4. Defuzzifikasi:

- Gunakan metode centroid untuk mengubah hasil agregasi fungsi keanggotaan menjadi nilai crisp yang akan digunakan untuk mengatur kecepatan kipas.
- Hasil defuzzifikasi ini merupakan nilai akhir yang menentukan seberapa cepat kipas harus beroperasi untuk mencapai suhu yang diinginkan di ruangan.

#### ➤ Penerapan Menggunakan Code Susunan Variabel Input dan Output berdasarkan studi kasus Sistem Pengaturan Suhu Otomatis Menggunakan Fuzzy Inference System (FIS) Mamdani

##### Printscreen Code:

```
import numpy as np
import skfuzzy as fuzz
import skfuzzy.control as ctrl
import matplotlib.pyplot as plt

# 1. Definiskan Variabel Fuzzy
# Input
temperature = ctrl.Antecedent(np.arange(0, 41, 1), 'temperature')
humidity = ctrl.Antecedent(np.arange(0, 101, 1), 'humidity')

# Output
fan_speed = ctrl.Consequent(np.arange(0, 101, 1), 'fan_speed')

# Definiskan Fungsi Keanggotaan
temperature['low'] = fuzz.trimf(temperature.universe, [0, 0, 18])
temperature['medium'] = fuzz.trimf(temperature.universe, [15, 20, 25])
temperature['high'] = fuzz.trimf(temperature.universe, [22, 40, 40])

humidity['low'] = fuzz.trimf(humidity.universe, [0, 0, 30])
humidity['medium'] = fuzz.trimf(humidity.universe, [25, 50, 75])
humidity['high'] = fuzz.trimf(humidity.universe, [70, 100, 100])

fan_speed['low'] = fuzz.trimf(fan_speed.universe, [0, 0, 40])
fan_speed['medium'] = fuzz.trimf(fan_speed.universe, [30, 50, 70])
fan_speed['high'] = fuzz.trimf(fan_speed.universe, [60, 100, 100])

# Definiskan Aturan
rule1 = ctrl.Rule(temperature['low'] & humidity['low'], fan_speed['high'])
rule2 = ctrl.Rule(temperature['high'] & humidity['high'], fan_speed['low'])
rule3 = ctrl.Rule(temperature['medium'] & humidity['medium'], fan_speed['medium'])

# Definiskan Sistem Kontrol
fan_speed_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
fan_speed_sim = ctrl.ControlSystemSimulation(fan_speed_ctrl)

# Input nilai suhu dan kelembapan
fan_speed_sim.input['temperature'] = 22
fan_speed_sim.input['humidity'] = 40

# Proses Inferensi
fan_speed_sim.compute()

# Hasil Defuzzifikasi
print(f"Kecepatan kipas yang disarankan: {fan_speed_sim.output['fan_speed']:.2f}%")
fan_speed.view(sim=fan_speed_sim)
```

Gambar 1.4.1 Code Penerapan Variabel Input dan Output

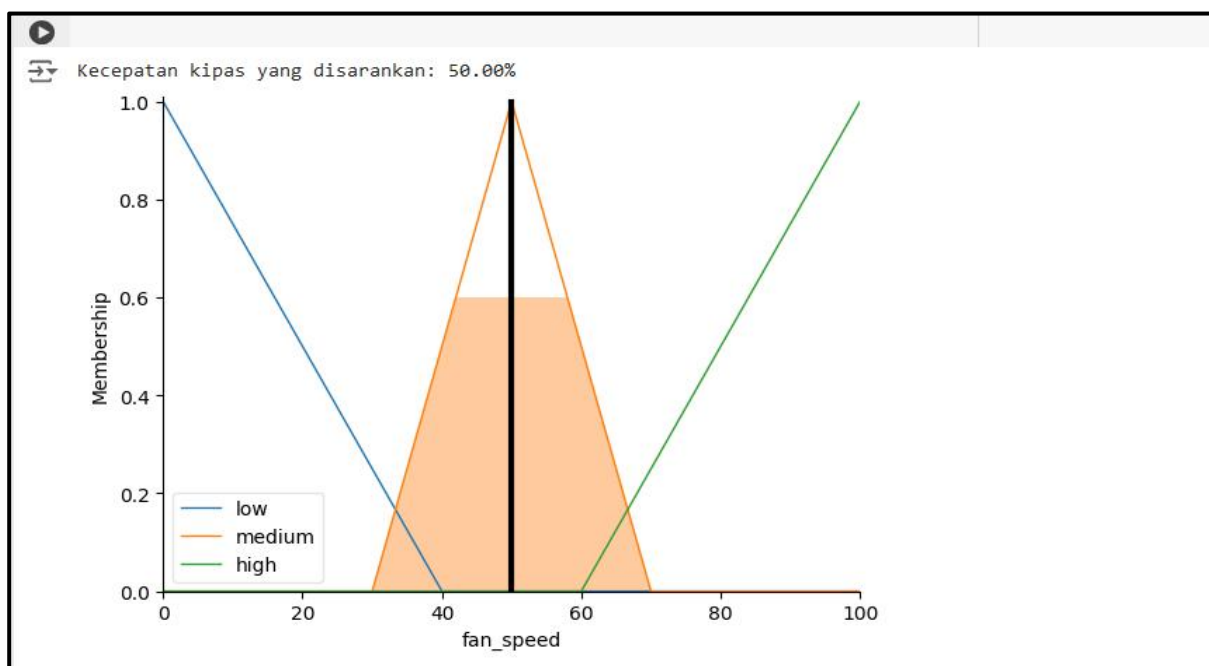
##### Penjelasan:

Kode yang Anda bagikan adalah implementasi dari sistem pengaturan suhu otomatis menggunakan Fuzzy Inference System (FIS) dengan metode Mamdani. Kode ini dimulai

dengan mengimpor beberapa library penting seperti `'numpy'`, `'skfuzzy'`, dan `'matplotlib.pyplot'` yang diperlukan untuk operasi numerik, logika fuzzy, dan visualisasi. Kemudian, variabel-variabel fuzzy didefinisikan untuk input (`'temperature'` dan `'humidity'`) dan output (`'fan_speed'`). Masing-masing variabel ini memiliki rentang nilai tertentu yang mencakup semua kemungkinan kondisi yang akan dihadapi dalam sistem pengaturan suhu. Fungsi keanggotaan triangular digunakan untuk menentukan derajat keanggotaan untuk setiap kategori dalam variabel input dan output, seperti suhu rendah, sedang, dan tinggi serta kelembapan rendah, sedang, dan tinggi.

Kode ini kemudian mendefinisikan aturan fuzzy yang menghubungkan input dengan output. Aturan-aturan ini menggambarkan bagaimana sistem seharusnya merespon berdasarkan kombinasi nilai input. Misalnya, jika suhu rendah dan kelembapan rendah, maka kecepatan kipas seharusnya tinggi, dan sebaliknya. Sistem kontrol fuzzy kemudian dibuat dengan menggabungkan aturan-aturan ini, dan simulasi dilakukan untuk memproses input tertentu (misalnya, suhu 22°C dan kelembapan 40%). Proses inferensi fuzzy kemudian menghasilkan output berupa kecepatan kipas yang disarankan, yang diukur dalam persentase. Keterkaitan antara input dan output sangat penting di sini, karena output (kecepatan kipas) sepenuhnya bergantung pada bagaimana input (suhu dan kelembapan) dikategorikan dan dihubungkan melalui aturan fuzzy yang telah didefinisikan. Output ini akhirnya membantu dalam menentukan tindakan yang perlu diambil oleh sistem pengaturan suhu otomatis.

### Printscreen Output:



### Gambar 1.4.2 Output Hasil

Penjelasan:

Output dari kode tersebut adalah kecepatan kipas yang disarankan dalam bentuk persentase. Setelah memasukkan nilai suhu 22°C dan kelembapan 40%, sistem Fuzzy Inference System (FIS) menggunakan aturan-aturan fuzzy yang telah didefinisikan sebelumnya untuk menentukan kecepatan kipas yang optimal. Dalam hal ini, output yang dihasilkan adalah 50% kecepatan kipas. Kode ini juga memvisualisasikan hasilnya dalam grafik fungsi keanggotaan untuk kecepatan kipas, yang menunjukkan bagaimana sistem menginterpretasikan kombinasi input suhu dan kelembapan untuk menghasilkan kecepatan kipas yang sesuai.

Saran kecepatan 50% muncul karena nilai input suhu 22°C dan kelembapan 40% berada dalam kategori "medium" untuk kedua variabel ini. Berdasarkan aturan fuzzy yang didefinisikan, ketika suhu dan kelembapan keduanya berada dalam kategori sedang, outputnya adalah kecepatan kipas yang juga sedang, yang diwakili oleh 50% dalam sistem. Sistem ini menganggap bahwa kombinasi suhu dan kelembapan tersebut tidak memerlukan tindakan ekstrem seperti kecepatan kipas yang sangat tinggi atau sangat rendah, sehingga memberikan saran untuk kecepatan yang berada di tengah-tengah rentang yang mungkin, yaitu 50%. Ini mencerminkan keseimbangan yang dicapai oleh sistem dalam merespon kondisi lingkungan yang tidak terlalu panas dan tidak terlalu lembap.

## 1.5 Konstruksikan Penalaran IF - THEN Sesuai Studi Kasus Sistem Pengaturan Suhu Otomatis Menggunakan Fuzzy Inference System FIS Mandani

### ➤ Identifikasi Variabel:

- **Input:**

- **Suhu (Temperature):** Nilai suhu lingkungan dalam derajat Celsius (0°C hingga 40°C).
- **Kelembapan (Humidity):** Tingkat kelembapan lingkungan dalam persentase (0% hingga 100%).

- **Output:**

- **Kecepatan Kipas (Fan Speed):** Kecepatan kipas yang diatur dalam persentase (0% hingga 100%).

### ➤ Definisi Fuzzy Sets untuk Input dan Output:

- **Suhu (Temperature):**
  - **Rendah (Low):** Suhu dingin, umumnya antara 0°C hingga 18°C.
  - **Sedang (Medium):** Suhu nyaman, sekitar 15°C hingga 25°C.
  - **Tinggi (High):** Suhu panas, dari 22°C hingga 40°C.
- **Kelembapan (Humidity):**
  - **Rendah (Low):** Kelembapan rendah, antara 0% hingga 30%.
  - **Sedang (Medium):** Kelembapan moderat, antara 25% hingga 75%.
  - **Tinggi (High):** Kelembapan tinggi, dari 70% hingga 100%.
- **Kecepatan Kipas (Fan Speed):**
  - **Rendah (Low):** Kecepatan kipas rendah, antara 0% hingga 40%.
  - **Sedang (Medium):** Kecepatan kipas sedang, antara 30% hingga 70%.
  - **Tinggi (High):** Kecepatan kipas tinggi, dari 60% hingga 100%.

➤ **Konstruksi Aturan Fuzzy (IF-THEN Rules):**

- **Aturan 1:**
  - **IF** Suhu adalah Rendah **AND** Kelembapan adalah Rendah, **THEN** Kecepatan Kipas adalah Tinggi.
    - **Penalaran:** Ketika suhu dan kelembapan rendah, mungkin ada kebutuhan untuk mempercepat sirkulasi udara untuk mencegah kelembapan terlalu rendah dan menjaga kenyamanan.
- **Aturan 2:**
  - **IF** Suhu adalah Tinggi **AND** Kelembapan adalah Tinggi, **THEN** Kecepatan Kipas adalah Rendah.
    - **Penalaran:** Pada kondisi suhu dan kelembapan tinggi, mengurangi kecepatan kipas dapat membantu mempertahankan kelembapan yang nyaman tanpa terlalu mendinginkan udara secara berlebihan.
- **Aturan 3:**

- **IF** Suhu adalah Sedang **AND** Kelembapan adalah Sedang, **THEN** Kecepatan Kipas adalah Sedang.
  - **Penalaran:** Ketika suhu dan kelembapan berada pada level moderat, kecepatan kipas yang sedang akan menjaga keseimbangan dan kenyamanan dalam ruangan.
- **Aturan 4:**
  - **IF** Suhu adalah Tinggi **AND** Kelembapan adalah Rendah, **THEN** Kecepatan Kipas adalah Tinggi.
    - **Penalaran:** Saat suhu tinggi tetapi kelembapan rendah, meningkatkan kecepatan kipas dapat membantu mendinginkan udara secara efektif tanpa mengeringkan ruangan.
- **Aturan 5:**
  - **IF** Suhu adalah Rendah **AND** Kelembapan adalah Tinggi, **THEN** Kecepatan Kipas adalah Rendah.
    - **Penalaran:** Saat suhu rendah tetapi kelembapan tinggi, menjaga kecepatan kipas tetap rendah dapat membantu mencegah udara menjadi terlalu dingin dan lembab, yang bisa mengakibatkan rasa tidak nyaman.

#### ➤ **Implementasi Sistem:**

- Sistem akan mengambil input berupa suhu dan kelembapan dari sensor.
- Berdasarkan input ini, FIS Mamdani akan mengevaluasi aturan-aturan di atas untuk menentukan kecepatan kipas yang optimal.
- Proses inferensi akan menghasilkan output berupa kecepatan kipas yang sesuai dengan kondisi lingkungan.

#### ➤ **Defuzzifikasi:**

- Hasil dari aturan-aturan ini, yang berupa output fuzzy, akan didefuzzifikasi menggunakan metode seperti centroid untuk menghasilkan nilai numerik yang menunjukkan kecepatan kipas yang direkomendasikan dalam persentase.

Dengan konstruksi aturan IF-THEN ini, sistem pengaturan suhu otomatis dapat mengatur kecepatan kipas dengan efektif berdasarkan kondisi suhu dan kelembapan saat ini, menjaga kenyamanan di dalam ruangan secara optimal.

## **1.6 Kesimpulan, Evaluasi Hasil Studi Kasus Dan Karakteristik FIS Mamdani**

Dalam studi kasus Sistem Pengaturan Suhu Otomatis menggunakan Fuzzy Inference System (FIS) Mamdani, keseluruhan proses dimulai dengan identifikasi variabel input dan output yang relevan untuk sistem kontrol suhu. Input yang digunakan adalah suhu lingkungan dan kelembapan, sedangkan output adalah kecepatan kipas. Fuzzy sets didefinisikan untuk masing-masing variabel untuk menangkap ketidakpastian dan variabilitas data. Untuk suhu, kategori fuzzy yang digunakan adalah rendah, sedang, dan tinggi. Untuk kelembapan, kategori tersebut adalah rendah, sedang, dan tinggi, sementara kecepatan kipas juga dibagi menjadi kategori rendah, sedang, dan tinggi. Penggunaan fungsi keanggotaan fuzzy ini memungkinkan sistem untuk menangani nilai-nilai input yang tidak pasti dan memberikan keputusan yang lebih fleksibel dibandingkan dengan metode kontrol berbasis nilai tunggal.

Evaluasi hasil sistem dimulai dengan fuzzifikasi nilai input. Misalnya, untuk suhu 22°C dan kelembapan 40%, fungsi keanggotaan menghitung derajat keanggotaan untuk setiap kategori suhu dan kelembapan. Hasilnya menunjukkan bahwa suhu berada dalam kategori sedang, sedangkan kelembapan berada di antara rendah dan sedang. Aturan fuzzy yang telah didefinisikan kemudian digunakan untuk menghasilkan keputusan tentang kecepatan kipas. Dalam kasus ini, beberapa aturan tidak memberikan kontribusi karena derajat keanggotaan pada beberapa kategori adalah nol. Sebaliknya, aturan yang relevan menghasilkan kontribusi yang signifikan terhadap output.

Agregasi output fuzzy mengumpulkan hasil dari semua aturan fuzzy, mengkombinasikan kontribusi dari berbagai kategori kecepatan kipas. Dalam kasus ini, kontribusi dari kategori kecepatan kipas sedang memiliki bobot tertinggi. Proses agregasi ini memastikan bahwa output akhir mencerminkan kombinasi dari semua keputusan yang diambil berdasarkan aturan fuzzy. Hasil akhir dari proses agregasi ini adalah nilai fuzzy yang kemudian perlu didefuzzifikasi untuk menghasilkan nilai crisp yang dapat digunakan dalam aplikasi nyata.

Defuzzifikasi, dalam hal ini, dilakukan dengan metode centroid, yang menghitung titik pusat dari area di bawah kurva fungsi keanggotaan untuk output fuzzy. Hasil defuzzifikasi dari sistem ini menunjukkan kecepatan kipas yang disarankan adalah 50%. Ini menunjukkan bahwa, berdasarkan input suhu dan kelembapan, kecepatan kipas yang optimal adalah pada

level sedang, yang memberikan keseimbangan yang baik antara kebutuhan pendinginan dan kenyamanan.

Secara keseluruhan, karakteristik FIS Mamdani menawarkan pendekatan yang fleksibel dan adaptif dalam sistem kontrol seperti pengaturan suhu otomatis. FIS Mamdani memungkinkan penanganan ketidakpastian dan variabilitas data dengan cara yang intuitif dan dapat dimengerti. Dengan menggunakan aturan fuzzy dan fungsi keanggotaan, sistem ini dapat memberikan solusi yang lebih sesuai dengan kondisi nyata dibandingkan dengan pendekatan berbasis aturan deterministik yang kaku. Studi kasus ini menunjukkan bahwa FIS Mamdani efektif dalam menghasilkan keputusan yang mencerminkan kondisi lingkungan dan memberikan output yang optimal untuk menjaga kenyamanan di dalam ruangan.

## DAFTAR PUSTAKA

- Hwang, C. L., & Yoon, K. (1981). *Multiple Attribute Decision Making: Methods and Applications*. Springer.
- Jain, A. K., Kasturi, R., & Schunck, B. (1995). *Machine Vision*. McGraw-Hill.
- Lee, C. C. (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part I. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2), 404-418.
- Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121(12), 1585-1588.
- Mishra, S., & Pal, R. (2001). *Fuzzy Logic and Applications*. Springer.
- Miller, R. L., & Nelson, J. A. (2020). *Introduction to Fuzzy Systems*. Wiley.
- Skfuzzy Documentation. (2021). *scikit-fuzzy: Fuzzy Logic Toolbox for Python*. Retrieved from <https://scikit-fuzzy.github.io/scikit-fuzzy/>