



DASAR KECERDASAN ARTIFISIAL

PREDIKSI BADAJ

— · — · —

dengan Fuzzy Logic

Disusun Oleh:

Veiron Vaya Yarief / 103012300100

Muhammad Ihsan Naufal / 103012300288



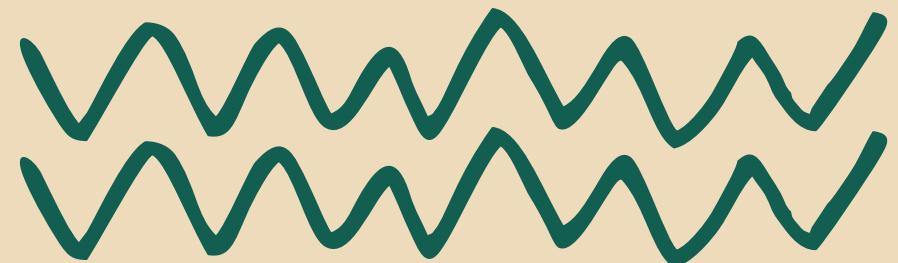
PENDAHULUAN

Rumusan Masalah

- Bagaimana merancang fungsi keanggotaan dan aturan fuzzy untuk variabel suhu, kelembapan, kecepatan angin, dan tekanan udara dalam rangka memprediksi datangnya badai?
- Sejauh mana prediksi badai yang dihasilkan model fuzzy

Tujuan

- Mengembangkan fungsi keanggotaan dan aturan fuzzy untuk variabel meteorologi utama (suhu, kelembapan, kecepatan angin, tekanan udara) dalam model prediksi badai.
- Menilai potensi penggunaan hasil prediksi badai oleh system fuzzy



DATA PROCESSING

```
In [235...]: data = pd.read_csv("processed_weather.csv", sep = ';', low_memory=False)

In [236...]: data.head()

Out[236...]:
```

	Date	Location	MinTemp	MaxTemp	Rainfall	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	Wind
0	01/12/2008	Albury	13.4	22.9	0.6	W	44.0	W	WNW	20.0	
1	02/12/2008	Albury	7.4	25.1	0.0	WNW	44.0	NNW	WSW	4.0	
2	03/12/2008	Albury	12.9	25.7	0.0	WSW	46.0	W	WSW	19.0	
3	04/12/2008	Albury	9.2	28.0	0.0	NE	24.0	SE	E	11.0	
4	05/12/2008	Albury	17.5	32.3	1.0	W	41.0	ENE	NW	7.0	

```
In [237...]: cols_to_convert = [
    'MinTemp', 'MaxTemp', 'Temp9am', 'Temp3pm',
    'Humidity9am', 'Humidity3pm',
    'WindSpeed9am', 'WindSpeed3pm', 'WindGustSpeed',
    'Pressure9am', 'Pressure3pm'
]

for col in cols_to_convert:
    data[col] = pd.to_numeric(data[col], errors = 'coerce')
```

DATA PROCESSING

In [238...]

```
data['temp_day_avg'] = data[['MinTemp', 'MaxTemp', 'Temp9am', 'Temp3pm']].mean(axis=1)
data['humidity_avg'] = data[['Humidity9am', 'Humidity3pm']].mean(axis=1)
data['wind_speed_avg'] = data[['WindSpeed9am', 'WindSpeed3pm', 'WindGustSpeed']].mean(axis=1)
data['pressure_avg'] = data[['Pressure9am', 'Pressure3pm']].mean(axis=1)
```

In [239...]

```
processed_data = data[['temp_day_avg', 'humidity_avg', 'wind_speed_avg', 'pressure_avg']]
processed_data.head()
```

DATA PROCESSING

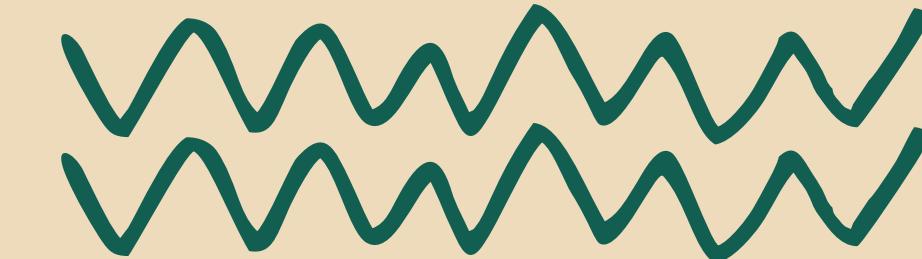
In [239...]

```
processed_data = data[['temp_day_avg', 'humidity_avg', 'wind_speed_avg', 'pressure_avg']]  
processed_data.head()
```

Out[239...]

	temp_day_avg	humidity_avg	wind_speed_avg	pressure_avg
0	18.750	46.5	29.333333	1007.40
1	18.500	34.5	23.333333	1009.20
2	20.700	34.0	30.333333	1008.15
3	20.450	30.5	14.666667	1015.20
4	24.325	57.5	22.666667	1008.40

MAMDANI



Metode Mamdani menggunakan fungsi keanggotaan fuzzy pada output-nya. Hasil dari proses inferensi ini masih berupa nilai fuzzy yang kemudian harus dilakukan proses defuzzifikasi untuk mendapatkan nilai crisp (tegas).

Karakteristik utama metode Mamdani:

- Lebih intuitif dan mudah dipahami oleh manusia.
- Cocok digunakan untuk sistem yang meniru pola pikir manusia.
- Hasil output cenderung lebih bervariasi.
- Proses perhitungan lebih kompleks karena melibatkan defuzzifikasi.

MAMDANI

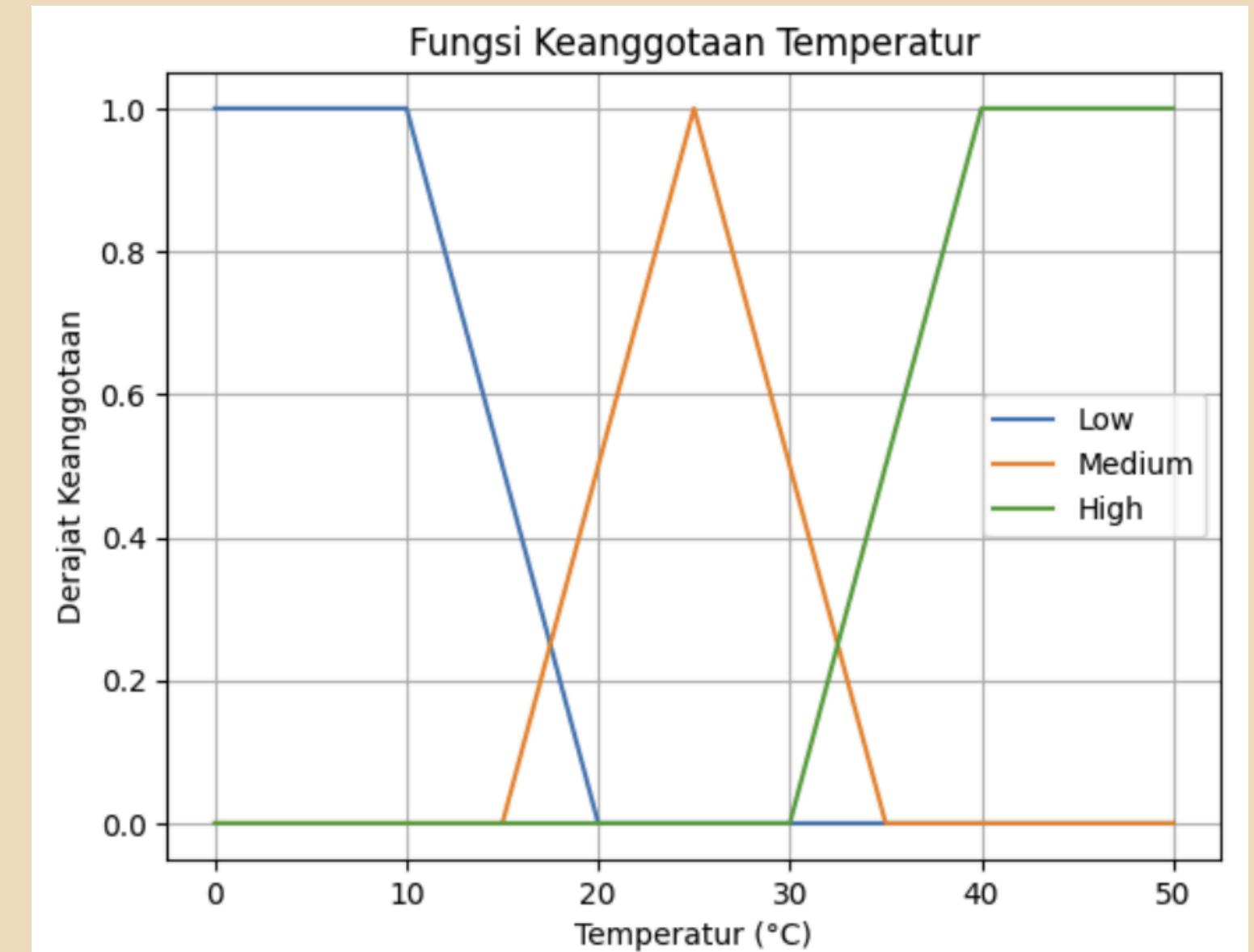
Var: Temperature

Fungsi Keanggotaan

```
temp = ctrl.Antecedent(np.arange(0, 51, 1), 'Temp')
temp['Low'] = fuzz.trapmf(temp.universe, [0, 0, 10, 20])
temp['Medium'] = fuzz.trimf(temp.universe, [15, 25, 35])
temp['High'] = fuzz.trapmf(temp.universe, [30, 40, 50, 50])
```

Batas Nilai Linguistik

Variabel	Universe	Label	Batas Numeric (a,b,c,d)
Temp	0 ... 50 °C	Low	(0, 0, 10, 20)
		Medium	(15, 25, 35)
		High	(30, 40, 50, 50)



MAMDANI

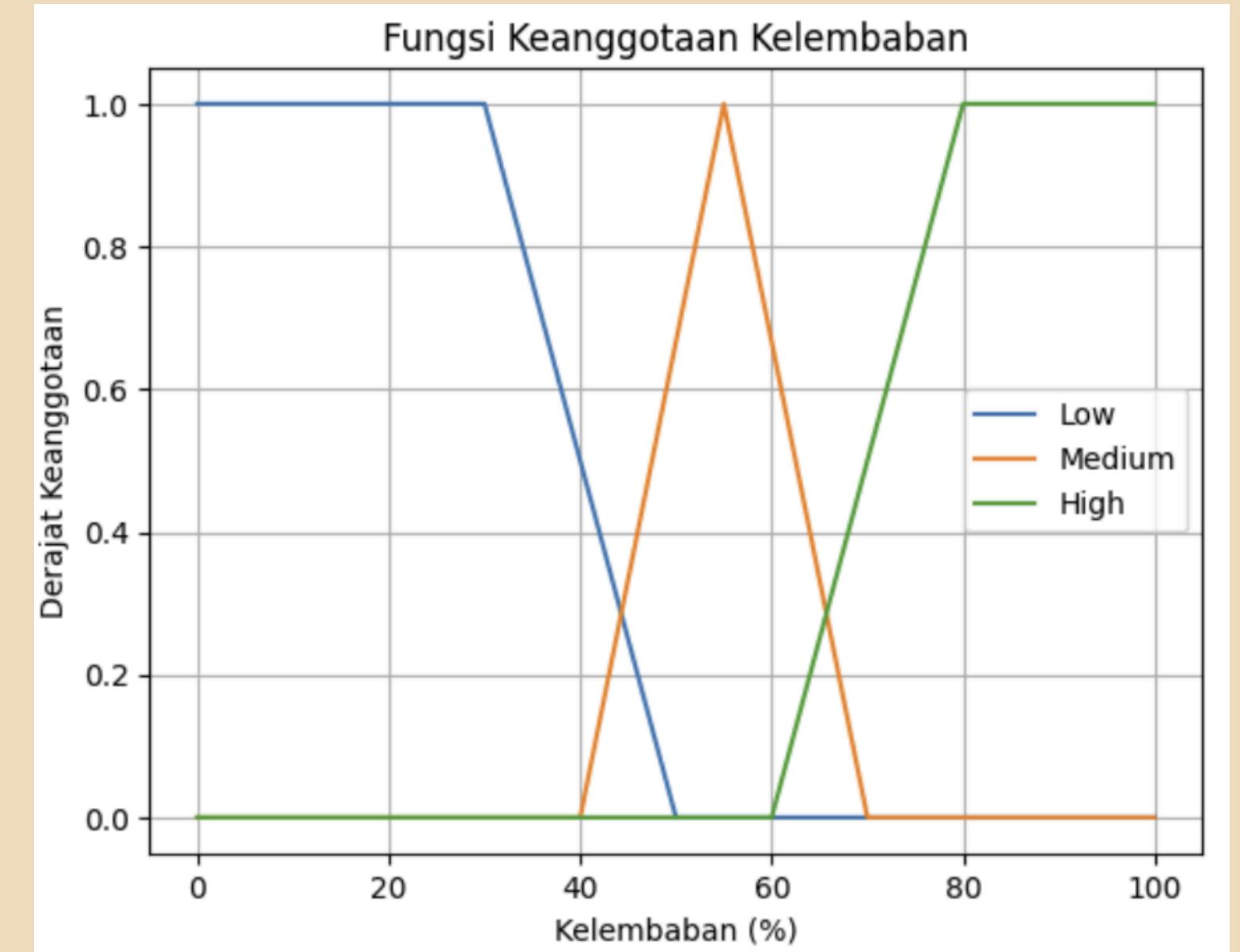
Var: Humidity

Fungsi Keanggotaan

```
humidity = ctrl.Antecedent(np.arange(0, 101, 1), 'Humidity')
humidity['Low']    = fuzz.trapmf(humidity.universe, [0, 0, 30, 50])
humidity['Medium'] = fuzz.trimf(humidity.universe, [40, 55, 70])
humidity['High']   = fuzz.trapmf(humidity.universe, [60, 80, 100, 100])
```

Batas Nilai Linguistik

Variabel	Universe	Label	Batas Numeric (a,b,c,d)
Humidity	0 ... 100 %	Low	(0, 0, 30, 50)
		Medium	(40, 55, 70)
		High	(60, 80, 100, 100)



MAMDANI

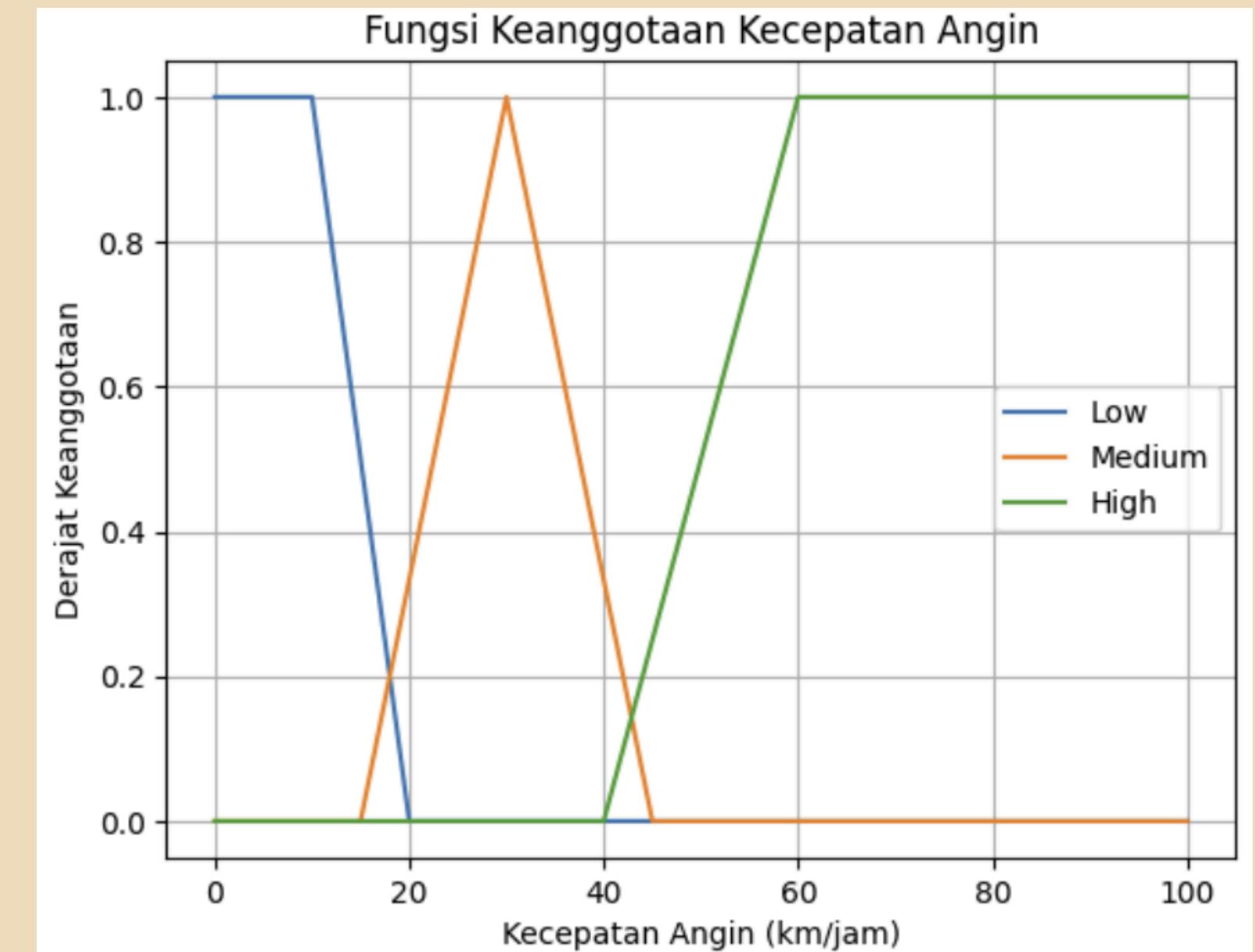
Var: WindSpeed

Fungsi Keanggotaan

```
wind = ctrl.Antecedent(np.arange(0, 101, 1), 'WindSpeed')
wind['Low'] = fuzz.trapmf(wind.universe, [0, 0, 10, 20])
wind['Medium'] = fuzz.trimf(wind.universe, [15, 30, 45])
wind['High'] = fuzz.trapmf(wind.universe, [40, 60, 100, 100])
```

Batas Nilai Linguistik

Variabel	Universe	Label	Batas Numeric (a,b,c,d)
WindSpeed	0 ... 100 km/jam	Low	(0, 0, 10, 20)
		Medium	(15, 30, 45)
		High	(40, 60, 100, 100)



MAMDANI

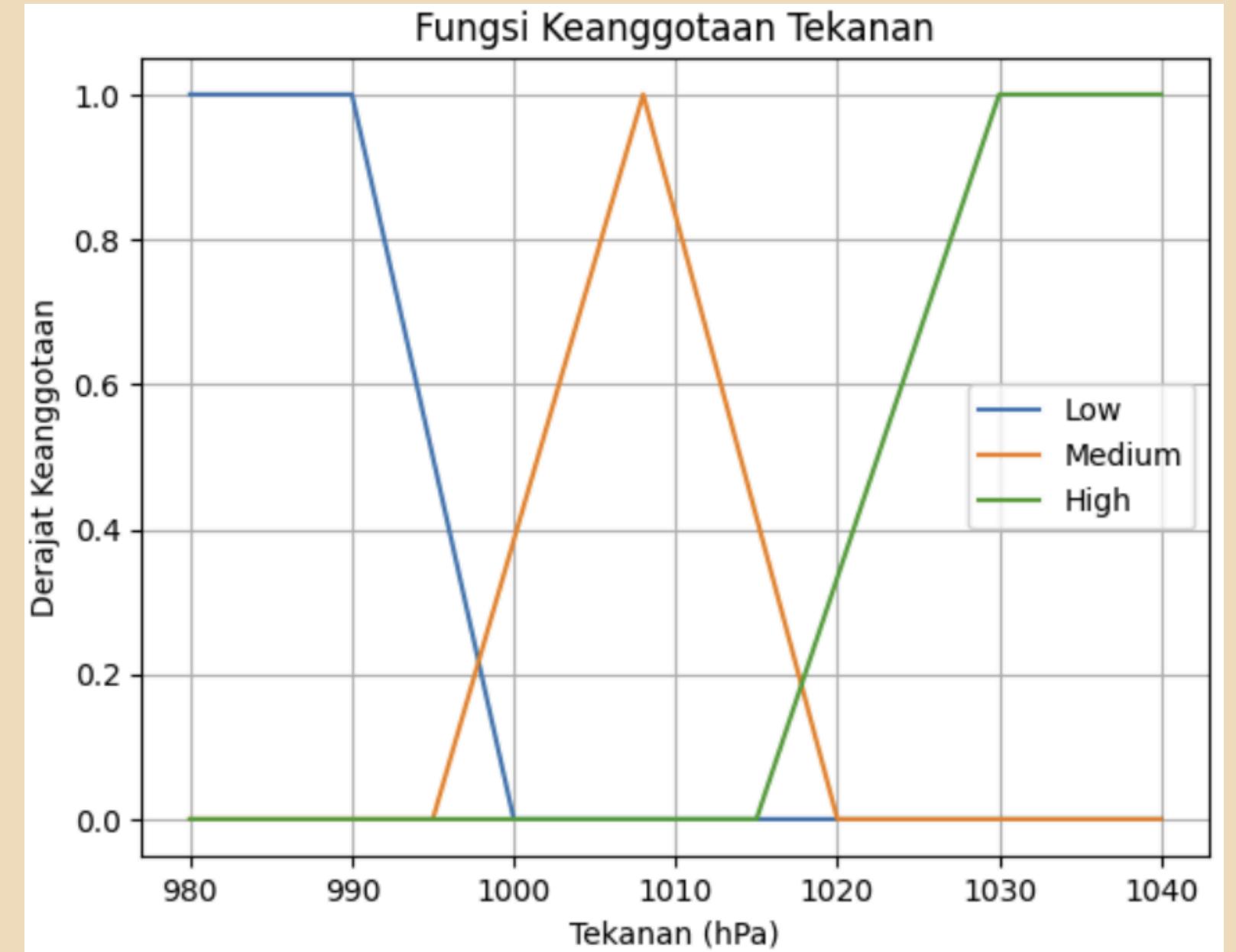
Var: Pressure

Fungsi Keanggotaan

```
pressure = ctrl.Antecedent(np.arange(980, 1041, 1), 'Pressure')
pressure['Low']    = fuzz.trapmf(pressure.universe, [980, 980, 990, 1000])
pressure['Medium'] = fuzz.trimf(pressure.universe, [995, 1008, 1020])
pressure['High']   = fuzz.trapmf(pressure.universe, [1015, 1030, 1040, 1040])
```

Batas Nilai Linguistik

Variabel	Universe	Label	Batas Numeric (a,b,c,d)
Pressure	980 ... 1040 hPa	Low	(980, 980, 990, 1000)
		Medium	(995, 1008, 1020)
		High	(1015, 1030, 1040, 1040)



MAMDANI

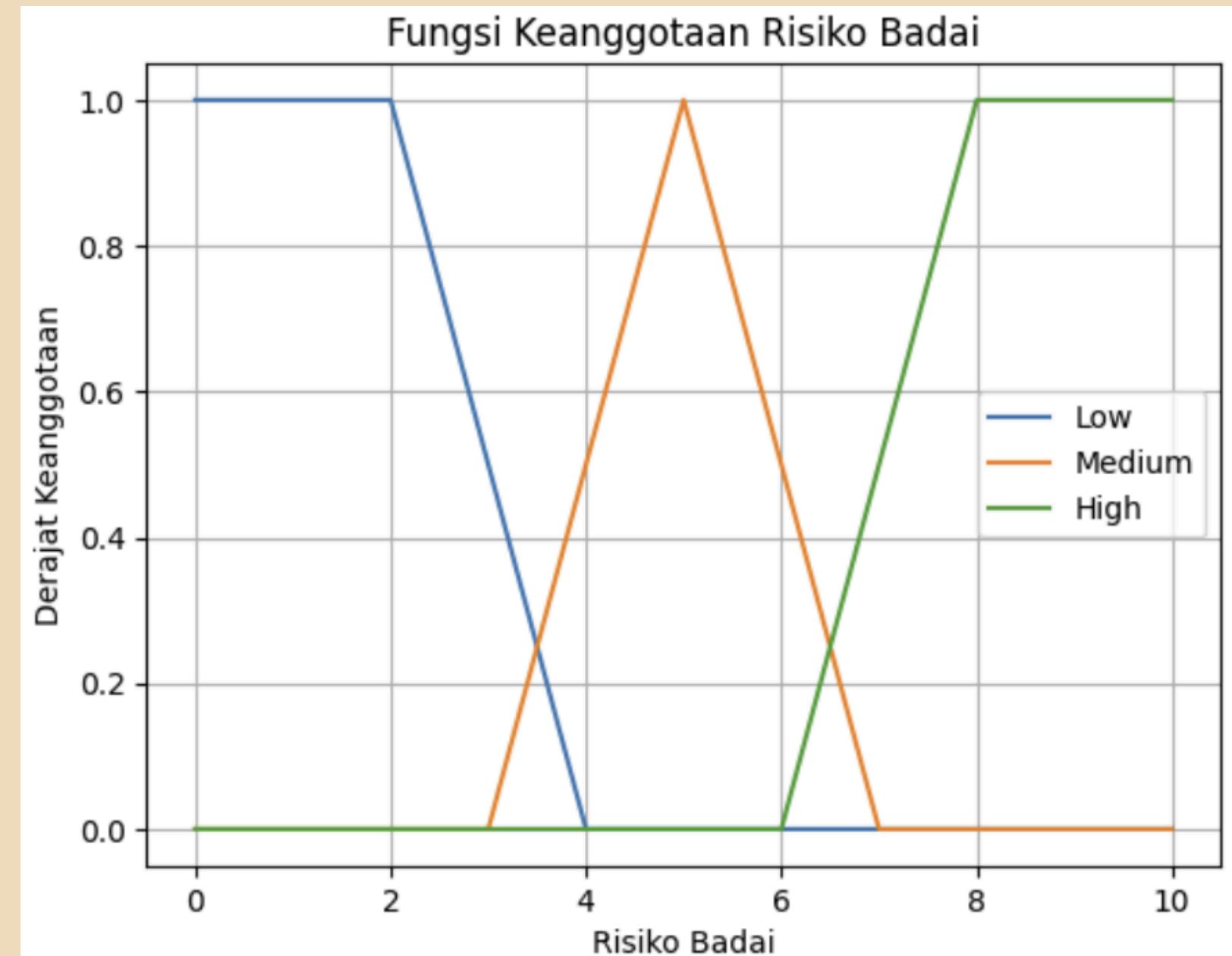
Var: StormRisk

Fungsi Keanggotaan

```
storm_risk = ctrl.Consequent(np.arange(0, 11, 1), 'StormRisk')
storm_risk['Low']    = fuzz.trapmf(storm_risk.universe, [0, 0, 2, 4])
storm_risk['Medium'] = fuzz.trimf(storm_risk.universe, [3, 5, 7])
storm_risk['High']   = fuzz.trapmf(storm_risk.universe, [6, 8, 10, 10])
```

Batas Nilai Linguistik

Variabel	Universe	Label	Batas Numeric (a,b,c,d)
StormRisk	0 ... 10	Low	(0, 0, 2, 4)
		Medium	(3, 5, 7)
		High	(6, 8, 10, 10)



MAMDANI

Fuzzy Rules

- High Risk jika: Pressure is Low OR WindSpeed is High
- High Risk jika: Humidity is High AND (WindSpeed is Medium OR Temp is High)
- Medium Risk jika: Humidity, Temp, dan WindSpeed semuanya Medium
- Low Risk jika: Pressure is Medium AND Humidity is Low
- Low Risk jika: WindSpeed is Low AND Pressure is High
- Default Medium (catch-all) - menjamin selalu ada firing rule: Temp is Low OR Medium OR High

```
# (1) High Risk if (Pressure is Low) OR (Wind is High)
rule1 = ctrl.Rule(
    pressure['Low'] | wind['High'],
    storm_risk['High']
)

# (2) High Risk if (Humidity is High) AND (Wind is Medium OR Temp is High)
rule2 = ctrl.Rule(
    (humidity['High'] & (wind['Medium'] | temp['High'])),
    storm_risk['High']
)

# (3) Medium Risk if (Humidity, Temp, and Wind are all Medium)
rule3 = ctrl.Rule(
    (humidity['Medium'] & temp['Medium'] & wind['Medium']),
    storm_risk['Medium']
)

# (4) Low Risk if (Pressure is Medium) AND (Humidity is Low)
rule4 = ctrl.Rule(
    (pressure['Medium'] & humidity['Low']),
    storm_risk['Low']
)

# (5) Low Risk if (Wind is Low) AND (Pressure is High)
rule5 = ctrl.Rule(
    (wind['Low'] & pressure['High']),
    storm_risk['Low']
)

# (6) Default: catch-all → Medium Risk
# Since for any numeric Temp, at Least one of temp['Low'], temp['Medium'], temp['High'] is
# this rule will always fire to some degree.
rule6 = ctrl.Rule(
    (temp['Low'] | temp['Medium'] | temp['High']),
    storm_risk['Medium']
)
```

MAMDANI

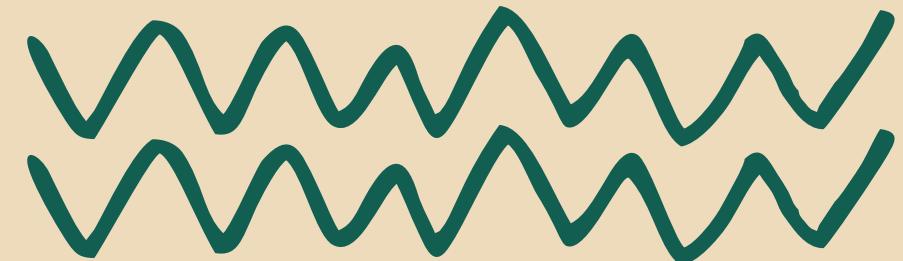
Final Storm Risk Prediction:

	Temp (°C)	Humidity (%)	Wind (km/h)	Pressure (hPa)	Storm Risk Score	Risk Category
0	18.750	46.5	29.333333	1007.40	3.923	Medium
1	18.500	34.5	23.333333	1009.20	2.686	Low
2	20.700	34.0	30.333333	1008.15	2.943	Low
3	20.450	30.5	14.666667	1015.20	3.482	Medium
4	24.325	57.5	22.666667	1008.40	5.000	Medium
5	23.450	39.0	33.000000	1007.30	3.393	Medium
6	20.500	34.0	31.333333	1008.90	2.926	Low
7	19.050	33.5	19.333333	1011.75	2.872	Low
8	22.525	25.5	38.333333	1006.25	2.995	Low
9	22.875	42.5	18.000000	1006.35	3.682	Medium
10	23.250	35.0	17.666667	1010.25	3.136	Medium

MAMDANI

11	17.625	90.0	19.666667	1007.35	6.772	Medium
12	16.925	84.5	39.000000	993.65	7.241	High
13	17.300	54.0	29.333333	1001.50	5.000	Medium
14	18.100	44.5	17.000000	1009.20	3.448	Medium
15	20.250	39.0	36.000000	1011.85	3.205	Medium
16	17.575	75.5	14.000000	1011.30	5.000	Medium
17	18.975	72.5	29.666667	1004.00	7.063	High
18	17.550	39.5	28.000000	1009.55	2.784	Low
19	18.600	35.5	16.333333	1018.15	3.763	Medium

SUGENO



Metode Sugeno menghasilkan output dalam bentuk fungsi linear atau konstan dari input. Oleh karena itu, hasil dari metode ini biasanya berupa nilai tegas tanpa perlu dilakukan defuzzifikasi yang kompleks.

Karakteristik utama metode Sugeno:

- Output lebih sederhana dan mudah dikomputasi.
- Cocok untuk sistem kontrol adaptif atau sistem yang membutuhkan respon cepat.

SUGENO

Fungsi Keanggotaan

Temperatur

```
subProcData['temp_norm'] = (subProcData['temp_day_avg'] - 0) / (50 - 0)
```

Humidity

```
subProcData['humidity_norm'] = subProcData['humidity_avg'] / 100
```

WindSpeed

```
subProcData['wind_norm'] = subProcData['wind_speed_avg'] / 100
```

Pressure

```
subProcData['pressure_norm'] = (subProcData['pressure_avg'] - 980) / (60)
```

Nilai Linguistik

Variabel	Nilai Linguistik
Temp	Tidak ada label linguistik
Humidity	Tidak ada label linguistic
WindSpeed	Tidak ada label linguistic
Pressure	Tidak ada label linguistik
StormRisk	Low, Medium, High

Batas Nilai Linguistik

Kategori	Rentang Storm Risk Score (0...10)
Low	$0.0 \leq \text{score} < 4.8$
Medium	$4.8 \leq \text{score} < 7.0$
High	$7.0 \leq \text{score} \leq 10.0$

SUGENO

Fuzzy Rules

w_temp = 0.6

w_hum = 0.4

w_wind = 0.2

w_pres = 0.2

SUGENO

Final Storm Risk Prediction (Sugeno):

	Temp (°C)	Humidity (%)	Wind (km/h)	Pressure (hPa)	Storm Risk Score	Risk Category
0	18.750	46.5	29.333333	1007.40	5.783	Medium
1	18.500	34.5	23.333333	1009.20	5.093	Medium
2	20.700	34.0	30.333333	1008.15	5.512	Medium
3	20.450	30.5	14.666667	1015.20	4.794	Low
4	24.325	57.5	22.666667	1008.40	6.726	Medium
5	23.450	39.0	33.000000	1007.30	6.124	Medium
6	20.500	34.0	31.333333	1008.90	5.483	Medium
7	19.050	33.5	19.333333	1011.75	4.954	Medium
8	22.525	25.5	38.333333	1006.25	5.615	Medium
9	22.875	42.5	18.000000	1006.35	5.927	Medium

SUGENO

10	23.250	35.0	17.666667	1010.25	5.535	Medium
11	17.625	90.0	19.666667	1007.35	7.197	High
12	16.925	84.5	39.000000	993.65	7.736	High
13	17.300	54.0	29.333333	1001.50	6.106	Medium
14	18.100	44.5	17.000000	1009.20	5.319	Medium
15	20.250	39.0	36.000000	1011.85	5.648	Medium
16	17.575	75.5	14.000000	1011.30	6.366	Medium
17	18.975	72.5	29.666667	1004.00	6.970	Medium
18	17.550	39.5	28.000000	1009.55	5.261	Medium
19	18.600	35.5	16.333333	1018.15	4.707	Low

ANALISIS

Berdasarkan output yang dihasilkan dari kedua metode:

- Metode Sugeno memberikan hasil yang cenderung tidak bervariasi, dengan rentang nilai output yang sempit. Hal ini dapat menyebabkan kurangnya fleksibilitas dalam interpretasi hasil, terutama jika variasi output dibutuhkan untuk pengambilan keputusan.
- Metode Mamdani menghasilkan output yang lebih bervariasi, memberikan representasi yang lebih kaya terhadap kondisi fuzzy dari input. Hal ini menjadikannya lebih sesuai untuk kasus-kasus yang membutuhkan pemahaman atau interpretasi kompleks.

EXTRA

MAMDANI X RANDOM FOREST

```
# Generate target labels using fuzzy logic
processed_data['StormRiskLabel'] = processed_data.apply(row_to_label, axis=1)

# Prepare for training
X = processed_data[['temp_day_avg', 'humidity_avg', 'wind_speed_avg', 'pressure_avg']]
y = processed_data['StormRiskLabel']

le = LabelEncoder()
y_encoded = le.fit_transform(y)

# Stratified split to include all classes
X_train, X_test, y_train, y_test = train_test_split(
    X, y_encoded,
    test_size=0.2,
    random_state=42,
    stratify=y_encoded
)
```

MAMDANI X RANDOM FOREST

```
# Train Random Forest
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

# Evaluation
all_labels = list(range(len(le.classes_)))
print("\nRandom Forest Classification Report:")
print(classification_report(
    y_test,
    y_pred,
    labels=all_labels,
    target_names=le.classes_,
    zero_division=0
))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred, labels=all_labels))
```

MAMDANI X RANDOM FOREST

Random Forest Classification Report:

	precision	recall	f1-score	support
High	0.00	0.00	0.00	2
Low	1.00	0.58	0.74	12
Medium	0.92	1.00	0.96	86
accuracy			0.93	100
macro avg	0.64	0.53	0.57	100
weighted avg	0.92	0.93	0.91	100

Confusion Matrix:

```
[[ 0  0  2]
 [ 0  7  5]
 [ 0  0 86]]
```

KESIMPULAN

Berdasarkan karakteristik dan hasil pengujian kedua metode fuzzy (Mamdani dan Sugeno), dapat disimpulkan bahwa metode Mamdani lebih cocok digunakan ketika diperlukan variasi output yang kaya dan pendekatan yang lebih intuitif menyerupai pola pikir manusia. Meski proses inferensinya lebih kompleks karena memerlukan defuzzifikasi, keunggulan variasi nilai yang dihasilkan memungkinkan interpretasi yang lebih mendalam terhadap kondisi sistem yang bersifat fuzzy. Sebaliknya, metode Sugeno, dengan output berupa fungsi linear atau konstan, memberikan nilai tegas tanpa memerlukan defuzzifikasi yang kompleks, sehingga lebih sederhana dan cepat dihitung. Hal ini menjadikan Sugeno lebih sesuai untuk aplikasi kontrol adaptif atau sistem real-time yang membutuhkan respons cepat meski dengan rentang output yang lebih sempit.

KESIMPULAN

Dari segi kinerja prediksi menggunakan algoritma Random Forest, terlihat bahwa model mampu mengenali kelas “Medium” dengan sangat baik (recall 1.00 dan F1-score 0.96), namun mengalami kesulitan pada kelas “High” yang jumlah datanya sangat sedikit (hanya 2 sampel). Model sama sekali tidak memprediksi kelas “High,” sehingga precision, recall, dan F1-score untuk kelas tersebut bernilai nol. Untuk kelas “Low,” model menunjukkan precision sempurna (1.00), tetapi recall yang relatif rendah (0.58) menandakan bahwa beberapa sampel “Low” terlewat dan diklasifikasikan sebagai “Medium.”

KESIMPULAN

Secara keseluruhan, akurasi total Random Forest mencapai 93%, yang didukung oleh tingginya proporsi kelas “Medium” dalam dataset. Nilai rata-rata macro (tanpa mempertimbangkan proporsi kelas) menunjukkan precision 0.64, recall 0.53, dan F1-score 0.57, mengindikasikan ketidakseimbangan performa antar-kelas. Sementara itu, rata-rata weighted (menghitung proporsi setiap kelas) menghasilkan precision 0.92, recall 0.93, dan F1-score 0.91, memperlihatkan bahwa model sangat optimal pada kelas yang dominan.

KESIMPULAN

Dari keseluruhan hasil, dapat diambil beberapa pelajaran. Pertama, untuk masalah dengan distribusi kelas tidak seimbang—terutama ketika kelas minoritas sangat sedikit—perlu strategi tambahan (misalnya oversampling, penyesuaian bobot, atau pengumpulan data lebih banyak) agar kelas kecil seperti “High” dapat dikenali dengan lebih baik. Kedua, jika kebutuhan aplikasi menitikberatkan pada variasi output fuzzy yang mendalam, metode Mamdani akan lebih tepat meski perhitungan defuzzifikasinya lebih rumit. Sebaliknya, apabila kecepatan komputasi dan keluaran yang sederhana lebih diutamakan, metode Sugeno adalah pilihan yang lebih efisien. Secara keseluruhan, pemilihan metode dalam implementasi sistem fuzzy atau algoritma machine learning harus mempertimbangkan kompleksitas perhitungan, variasi output yang diinginkan, dan distribusi data agar diperoleh kinerja optimal sesuai kebutuhan aplikasi.



DASAR KECERDASAN ARTIFISIAL

TERIMA KASIH

Kepada semuanya yang telah hadir