

LAPORAN TUGAS BESAR
Dasar Kecerdasan Artifisial
(DKA)
“Alcohol Effects On Study”



Disusun Oleh :
Muhammad Nayubi Adiva R (103012300469)
Sena Rizky Pratama (103012330375)

TELKOM UNIVERSITY
PRODI S1 INFORMATIKA
2025

Deskripsi Dataset: Alcohol Effects On Study

A. Latar Belakang

Performa akademik siswa dipengaruhi oleh berbagai faktor seperti waktu belajar, lingkungan sosial, dan kebiasaan pribadi, termasuk konsumsi alkohol. Konsumsi alkohol pada remaja dapat menurunkan prestasi belajar dan meningkatkan risiko putus sekolah. Oleh karena itu, penting untuk memprediksi prestasi siswa berdasarkan faktor-faktor tersebut agar intervensi dini dapat dilakukan. Proyek ini akan merancang sistem prediksi performa akademik siswa menggunakan logika fuzzy, dengan membandingkan dua metode inferensi yaitu Mamdani dan Sugeno, untuk menilai akurasi dan kemudahan interpretasi aturan.

B. Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang akan dijawab melalui penelitian ini adalah sebagai berikut:

1. Bagaimana melakukan prapemrosesan data pada dataset “Alcohol Effects on Study” agar siap digunakan dalam sistem prediksi, termasuk seleksi fitur, penanganan data hilang, dan normalisasi?
2. Bagaimana merancang dan mengimplementasikan sistem prediksi performa belajar siswa berbasis logika fuzzy menggunakan metode Mamdani dan Sugeno?
3. Bagaimana menentukan fungsi keanggotaan dan menyusun aturan fuzzy yang tepat untuk kedua metode?
4. Bagaimana mengevaluasi kinerja sistem fuzzy Mamdani dan Sugeno menggunakan metrik seperti akurasi, presisi, recall, dan F1-score?
5. Metode mana yang lebih efektif dan interpretatif antara Mamdani dan Sugeno dalam memprediksi performa belajar siswa?

C. Tujuan

Sesuai rumusan masalah di atas, tujuan dari penelitian tugas besar ini adalah:

1. Melakukan prapemrosesan dataset “Alcohol Effects on Study” dari Kaggle, termasuk seleksi fitur, penanganan data hilang, dan normalisasi.
2. Merancang dan mengimplementasikan sistem prediksi performa akademik siswa menggunakan logika fuzzy metode Mamdani dan Sugeno.
3. Menetapkan fungsi keanggotaan dan aturan fuzzy yang sesuai untuk masing-masing metode.
4. Mengevaluasi kinerja sistem prediksi menggunakan metrik seperti akurasi, presisi, recall, dan F1-score.
5. Menganalisis dan membandingkan efektivitas serta interpretabilitas metode Mamdani dan Sugeno dalam prediksi performa belajar siswa

D. Metode

- 1) Paparan, statistik, dan sumber dari dataset yang digunakan
 - Sumber dataset: Kaggle – *Alcohol Effects on Study*, berasal dari UCI Student Performance Dataset.
 - Target output: Nilai akhir G3 yang dikategorikan ke dalam {Rendah, Sedang, Tinggi}.
 - Fitur utama:
 - Konsumsi alkohol harian dan akhir pekan (Dalc, Walc)
 - Waktu belajar mingguan (studytime)
 - Dukungan keluarga (famsup)
 - Absensi, kesehatan, frekuensi bersosialisasi.
- 2) Paparan pre-processing dataset
 - Pengecekan nilai hilang dan kebersihan data.
 - Normalisasi data numerik jika diperlukan.
 - Penghapusan G1 dan G2 untuk menjaga validitas prediksi dari faktor eksternal.
- 3) Penjelasan mengenai rancangan dua metode yang digunakan
 - Jumlah nilai linguistik untuk setiap atribut:
 - Misal: Dalc (Rendah, Sedang, Tinggi), Studytime (Pendek, Sedang, Lama)
 - Fuzzy rule
 - Mamdani: menggunakan konsekuen linguistik (mis. “Performa = Rendah”).
 - Sugeno: menggunakan output numerik (mis. 5, 12, 18) atau fungsi linear sederhana.
 - Aturan disusun berdasarkan logika domain: contoh: “IF Dalc = Tinggi AND Studytime = Pendek THEN Performa = Rendah”

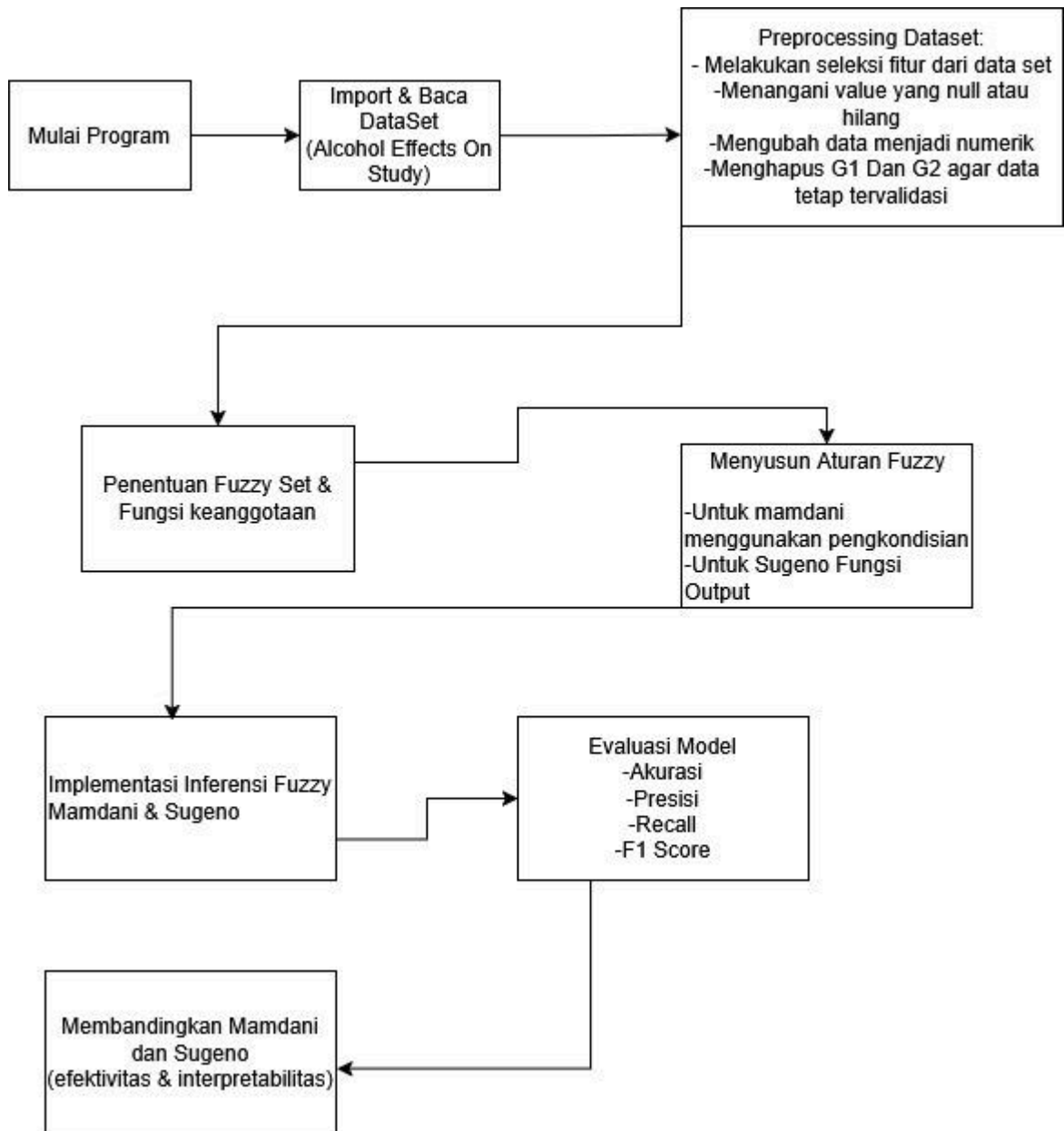
E. Metode

1. Paparan, statistik, dan sumber dari dataset yang digunakan
 - Sumber dataset: Kaggle – *Alcohol Effects on Study*, berasal dari UCI Student Performance Dataset.
 - Target output: Nilai akhir G3 yang dikategorikan ke dalam {Rendah, Sedang, Tinggi}.
 - Fitur utama:
 - Konsumsi alkohol harian dan akhir pekan (Dalc, Walc)
 - Waktu belajar mingguan (studytime)
 - Dukungan keluarga (famsup)
 - Absensi, kesehatan, frekuensi bersosialisasi.
2. Paparan pre-processing dataset
 - Pengecekan nilai hilang dan kebersihan data.
 - Normalisasi data numerik jika diperlukan.
 - Penghapusan G1 dan G2 untuk menjaga validitas prediksi dari faktor eksternal.
3. Penjelasan mengenai rancangan dua metode yang digunakan
 - Jumlah nilai linguistik untuk setiap atribut:
 - Misal: Dalc (Rendah, Sedang, Tinggi), Studytime (Pendek, Sedang, Lama)
 - Fuzzy rule
 - Mamdani: menggunakan konsekuen linguistik (mis. “Performa = Rendah”).
 - Sugeno: menggunakan output numerik (mis. 5, 12, 18) atau fungsi linear sederhana.
 - Aturan disusun berdasarkan logika domain: contoh: “IF Dalc = Tinggi AND Studytime = Pendek THEN Performa = Rendah”

F. Analisis

1. Evaluasi dilakukan pada data uji menggunakan metrik:
 - Akurasi
 - Presisi
 - Recall
 - F1-score
2. Hasil dari sistem Mamdani dan Sugeno dibandingkan dari segi:
 - Akurasi prediksi performa
 - Interpretabilitas aturan
 - Kompleksitas model

G. ALUR



H. Gambar untuk Fuzzy System

Isi Dataset awal:

Code:

```
#Isi Dataset awal
csv_path = Path('Alcohol_Effects_Study.csv')
assert csv_path.exists(), "Alcohol_Effects_Study.csv not found!"

data = pd.read_csv(csv_path)

print("Dataset awal", data.shape)
data.head()
```

Hasil :

```
Dataset awal (395, 33)
[62]:  school sex age address famsize Pstatus Medu Fedu Mjob Fjob ... famrel freetime goout Dalc Walc health absences G1 G2 G3
0      GP  F  18    U    GT3    A    4    4  at_home teacher ...    4    3    4    1    1    3    6    5    6    6
1      GP  F  17    U    GT3    T    1    1  at_home  other ...    5    3    3    1    1    3    4    5    5    6
2      GP  F  15    U    LE3    T    1    1  at_home  other ...    4    3    2    2    3    3   10    7    8   10
3      GP  F  15    U    GT3    T    4    2  health  services ...    3    2    2    1    1    5    2   15   14   15
4      GP  F  16    U    GT3    T    3    3    other    other ...    4    3    2    1    2    5    4    6   10   10

5 rows x 33 columns
```

Statistik Awal:

Code:

```
print("Statistik Awal:")
print(data.describe())
```

Hasil:

```
Statistik Awal:
count    age      Medu      Fedu  traveltime  studytime  failures  \
mean    16.696203  2.749367  2.521519   1.448101   2.035443   0.334177
std      1.276043  1.094735  1.088201   0.697505   0.839240   0.743651
min     15.000000  0.000000  0.000000   1.000000   1.000000   0.000000
25%     16.000000  2.000000  2.000000   1.000000   1.000000   0.000000
50%     17.000000  3.000000  2.000000   1.000000   2.000000   0.000000
75%     18.000000  4.000000  3.000000   2.000000   2.000000   0.000000
max     22.000000  4.000000  4.000000   4.000000   4.000000   3.000000

count    famrel  freetime  goout      Dalc      Walc      health  \
mean     3.944304   3.235443   3.108861   1.481013   2.291139   3.554430
std      0.896659   0.998862   1.113278   0.890741   1.287897   1.390303
min      1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
25%      4.000000   3.000000   2.000000   1.000000   1.000000   3.000000
50%      4.000000   3.000000   3.000000   1.000000   2.000000   4.000000
75%      5.000000   4.000000   4.000000   2.000000   3.000000   5.000000
max      5.000000   5.000000   5.000000   5.000000   5.000000   5.000000

count    absences      G1      G2      G3
mean     5.708861   10.908861   10.713924   10.415190
std      8.003096    3.319195    3.761505    4.581443
min      0.000000    3.000000    0.000000    0.000000
25%      0.000000    8.000000    9.000000    8.000000
50%      4.000000   11.000000   11.000000   11.000000
75%      8.000000   13.000000   13.000000   14.000000
max     75.000000   19.000000   19.000000   20.000000
```

Mempilkan jumlah missing value:

Code :

```
#Tampilkan jumlah missing value
missing_summary = pd.DataFrame({
    'Missing Values': missing_values
}).sort_values(by='Missing Values', ascending=False)

missing_summary.head(10)
```

Hasil :

Missing Values	
famsup	395
school	0
age	0
sex	0
famsize	0
Pstatus	0
Medu	0
address	0
Mjob	0
Fjob	0

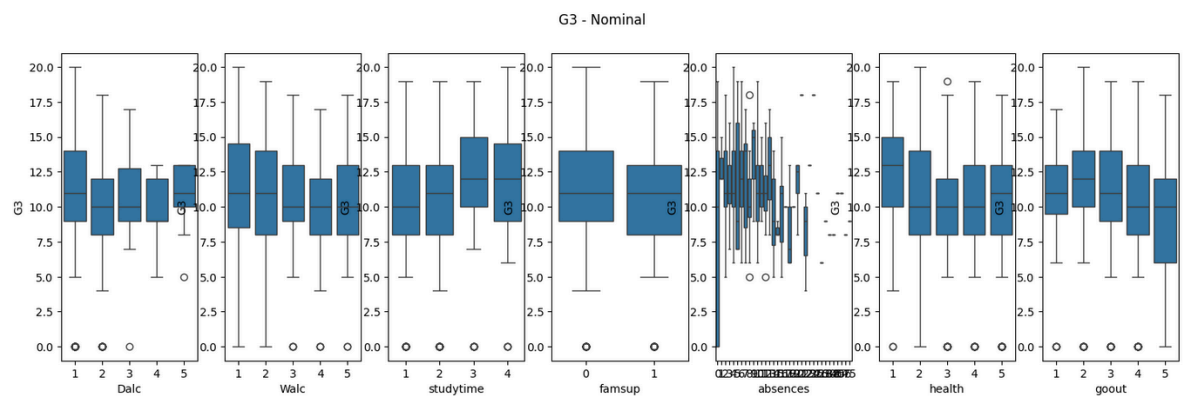
Fitur (boxplot)

Code :

```
features = ['Dalc', 'Walc', 'studytime', 'famsup', 'absences', 'health', 'goout']
#features = ['Dalc', 'Walc', 'studytime', 'health', 'goout']
target = 'G3'
data_selected = data_clean[features]
##visualisasi per g3 dari masing-masing feature

fig = plt.figure(figsize = (18,5))
g = gs.GridSpec(nrows = 1, ncols = 7, figure = fig)
plt.suptitle("G3 - Nominal")
j = 0
for feat in features:
    ax = plt.subplot(g[0,j])
    ax = sns.boxplot(data = data, x = data[feat], y = data["G3"])
    j = j+1
```

Hasil:



Fungsi keanggotaan fuzzy untuk G1, G2 , G3

Code:

```
#fungsi keanggotaan fuzzy untuk G1 dan G2
# Generate universe variables
x_g1 = np.arange(1, 19, 1)
x_g2 = np.arange(1, 19, 1)
x_g3 = np.arange(1, 20, 1)

# Generate fuzzy membership functions
x_g1_r = fuzz.trapmf(x_g1, [0, 0, 5, 7.5])
x_g1_s = fuzz.trapmf(x_g1, [5, 7.5, 10, 12.5])
x_g1_t = fuzz.trapmf(x_g1, [10, 12.5, 15, 17.5])

x_g2_r = fuzz.trapmf(x_g2, [0, 0, 5, 7.5])
x_g2_s = fuzz.trapmf(x_g2, [5, 7.5, 10, 12.5])
x_g2_t = fuzz.trapmf(x_g2, [10, 12.5, 15, 17.5])

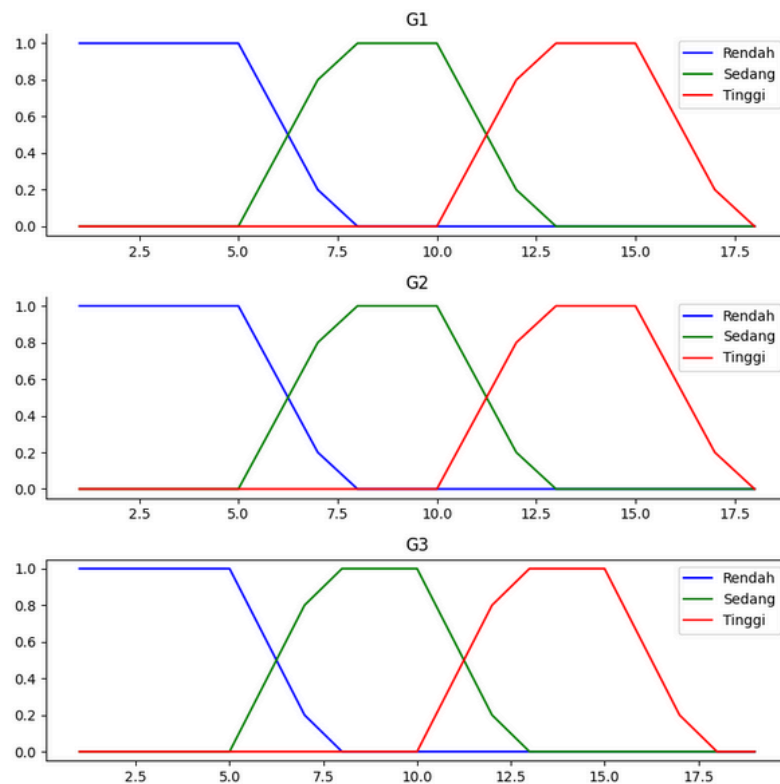
x_g3_r = fuzz.trapmf(x_g3, [0, 0, 5, 7.5])
x_g3_s = fuzz.trapmf(x_g3, [5, 7.5, 10, 12.5])
x_g3_t = fuzz.trapmf(x_g3, [10, 12.5, 15, 17.5])

SS =0.8
S =0.6
AS =0.4
TS =0.2

# Visualize these universes and membership functions
fig, (ax0, ax1, ax2) = plt.subplots(nrows=3, figsize=(8, 8))
ax0.plot(x_g1, x_g1_r, 'b', linewidth=1.5, label='Rendah')
ax0.plot(x_g1, x_g1_s, 'g', linewidth=1.5, label='Sedang')
ax0.plot(x_g1, x_g1_t, 'r', linewidth=1.5, label='Tinggi')
ax0.set_title('G1')
ax0.legend()
ax1.plot(x_g2, x_g2_r, 'b', linewidth=1.5, label='Rendah')
ax1.plot(x_g2, x_g2_s, 'g', linewidth=1.5, label='Sedang')
ax1.plot(x_g2, x_g2_t, 'r', linewidth=1.5, label='Tinggi')
ax1.set_title('G2')
ax1.legend()
ax2.plot(x_g3, x_g3_r, 'b', linewidth=1.5, label='Rendah')
ax2.plot(x_g3, x_g3_s, 'g', linewidth=1.5, label='Sedang')
ax2.plot(x_g3, x_g3_t, 'r', linewidth=1.5, label='Tinggi')
ax2.set_title('G3')
ax2.legend()
# Turn off top/right axes
for ax in (ax0, ax1):
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.get_xaxis().tick_bottom()
    ax.get_yaxis().tick_left()

plt.tight_layout()
```

Hasil:



Evaluasi

```
# Konversi 'famsup' menjadi numerik
data['famsup'] = data['famsup'].map({'yes': 1, 'no': 0})
```

```
# Hapus nilai G1 dan G2 untuk menjaga validitas prediksi
data = data.drop(['G1', 'G2'], axis=1)
```

```
# Kategorikan G3 menjadi Rendah, Sedang, Tinggi
def categorize_g3(score):
    if score <= 10:
        return 'Rendah'
    elif score <= 15:
        return 'Sedang'
    else:
        return 'Tinggi'
```

```
data['G3_cat'] = data['G3'].apply(categorize_g3)
```

```
# Bagi data menjadi fitur dan target
X = data[features]
y = data['G3_cat']
```

```
# Bagi data menjadi training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```

# Definisikan fungsi keanggotaan untuk setiap fitur
def fuzzify_studytime(value):
    if value <= 1:
        return {'Pendek': 1, 'Sedang': 0, 'Lama': 0}
    elif value == 2:
        return {'Pendek': 0.5, 'Sedang': 0.5, 'Lama': 0}
    elif value == 3:
        return {'Pendek': 0, 'Sedang': 0.5, 'Lama': 0.5}
    else:
        return {'Pendek': 0, 'Sedang': 0, 'Lama': 1}

def fuzzify_dalc(value):
    if value <= 1:
        return {'Rendah': 1, 'Sedang': 0, 'Tinggi': 0}
    elif value == 2:
        return {'Rendah': 0.5, 'Sedang': 0.5, 'Tinggi': 0}
    elif value == 3:
        return {'Rendah': 0, 'Sedang': 1, 'Tinggi': 0}
    elif value == 4:
        return {'Rendah': 0, 'Sedang': 0.5, 'Tinggi': 0.5}
    else:
        return {'Rendah': 0, 'Sedang': 0, 'Tinggi': 1}

```

```

# Definisikan aturan fuzzy Mamdani
def mamdani_inference(row):
    studytime = fuzzify_studytime(row['studytime'])
    dalc = fuzzify_dalc(row['Dalc'])

    # Aturan: IF Dalc = Tinggi AND Studytime = Pendek THEN Performa = Rendah
    rule1 = min(dalc['Tinggi'], studytime['Pendek'])

    # Aturan: IF Dalc = Sedang AND Studytime = Sedang THEN Performa = Sedang
    rule2 = min(dalc['Sedang'], studytime['Sedang'])

    # Aturan: IF Dalc = Rendah AND Studytime = Lama THEN Performa = Tinggi
    rule3 = min(dalc['Rendah'], studytime['Lama'])

    # Agregasi hasil
    output = {'Rendah': rule1, 'Sedang': rule2, 'Tinggi': rule3}
    return max(output, key=output.get)

# Terapkan inferensi Mamdani pada data uji
y_pred_mamdani = X_test.apply(mamdani_inference, axis=1)

```

```

# Definisikan fungsi inferensi Sugeno
def sugeno_inference(row):
    # Aturan: IF Dalc = Tinggi AND Studytime = Pendek THEN Performa = 5
    if row['Dalc'] >= 4 and row['studytime'] <= 1:
        return 5
    # Aturan: IF Dalc = Rendah AND Studytime = Lama THEN Performa = 18
    elif row['Dalc'] <= 1 and row['studytime'] >= 3:
        return 18
    # Aturan: IF Dalc = Sedang AND Studytime = Sedang THEN Performa = 12
    else:
        return 12

# Terapkan inferensi Sugeno pada data uji
y_pred_sugeno_numeric = X_test.apply(sugeno_inference, axis=1)

# Kategorikan hasil prediksi Sugeno
def categorize_sugeno(score):
    if score <= 10:
        return 'Rendah'
    elif score <= 15:
        return 'Sedang'
    else:
        return 'Tinggi'

y_pred_sugeno = y_pred_sugeno_numeric.apply(categorize_sugeno)

def mape(actual, pred):
    actual, pred = np.array(actual), np.array(pred)
    return np.mean(np.abs((actual - pred) / actual)) * 100

```

hasil

```

Evaluasi Model: Fuzzy Mamdani
Akurasi: 40.51%
Presisi: 32.67%
Recall : 31.12%
F1-Score: 27.41%
MAPE: 32.278481012658226

```

```

Evaluasi Model: Fuzzy Sugeno
Akurasi: 35.44%
Presisi: 49.23%
Recall : 32.17%
F1-Score: 22.66%
MAPE: 46.835443037974684

```

Visualisasi distribusi prediksi:

code

```
# Visualisasi distribusi prediksi
import seaborn as sns

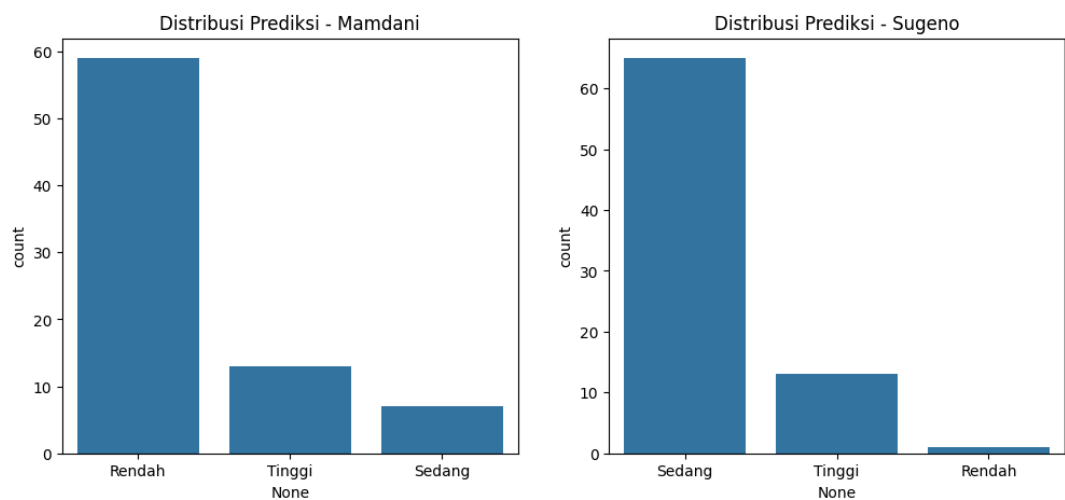
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.countplot(x=y_pred_mamdani)
plt.title('Distribusi Prediksi - Mamdani')

plt.subplot(1, 2, 2)
sns.countplot(x=y_pred_sugeno)
plt.title('Distribusi Prediksi - Sugeno')

plt.show()
```

plt.show()



I. Kesimpulan

- Sistem fuzzy baik Mamdani maupun Sugeno dapat digunakan untuk memprediksi performa belajar siswa.
- Pemilihan metode tergantung pada kebutuhan: jika transparansi penting → Mamdani; jika performa numerik utama → Sugeno.

Namun untuk data set kali ini

- Mamdani unggul di regresi pada eksperimen Ini karena kesalahan persentasenya ± 14 poin lebih rendah.
- Dengan kata lain, rata-rata prediksi Mamdani hanya meleset ± 32 % dari nilai G3 sebenarnya, sedangkan Sugeno meleset ± 47 %.

J. Referensi

<https://www.kaggle.com/code/alexandrepetit881234/alcohol-and-other-factors-effects-on>

<https://www.kaggle.com/datasets/whenamancodes/alcohol-effects-on-study>