

TUGAS KELOMPOK
FUZZY LOGIC (ANALISIS FIS-MAMDANI)



DISUSUN OLEH :

- | | |
|------------------------|-------------|
| 1. Davi Sulaiman | (G1A022001) |
| 2. Rafi Afrian | (G1A022033) |
| 3. Fahim Ahmad Saputra | (G1A022037) |
| 4. Ade Irawan | (G1A022083) |

Dosen Pengampu :

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

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
2024

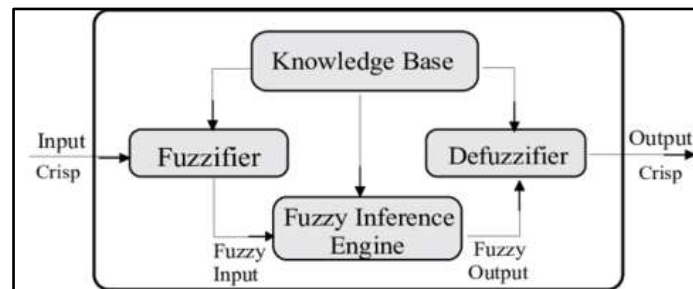
1.1. Analisa FIS-MAMDANI

Logika Fuzzy adalah salah satu pembentukan *soft computing* yang digunakan untuk memecahkan masalah dari input dan output, logika fuzzy memiliki beberapa kelebihan seperti mudah dimengerti, karena konsep matematis yang sederhana, fleksibel digunakan, toleransi terhadap data-data yang tidak tepat, mampu memodelkan fungsi-fungsi non-linier yang kompleks, dan sebagainya. Logika fuzzy memiliki banyak penerapan di industri seperti bidang kesehatan, ekonomi, dan teknologi yang dapat membantu manusia dalam memecahkan suatu masalah. Dalam penerapan logika ini terdapat beberapa proses, salah satunya yaitu sistem inferensi. Sistem inferensi fuzzy merupakan kerangka komputasi yang didasarkan pada teori himpunan fuzzy, aturan fuzzy berbentuk IF - THEN, dan penalaran fuzzy (Athiyah et al., 2021). Dalam penalaran inferensi fuzzy terdapat tiga metode yaitu, metode tsukamoto, metode mamdani, dan metode sugeno.

Metode Mamdani adalah metode yang paling sering di jumpai ketika membahas metodologi - metodologi fuzzy. Hal ini mungkin karena metode ini merupakan metode yang pertama kali dibangun dan berhasil diterapkan dalam rancang bangun sistem kontrol. Menggunakan teori himpunan fuzzy. Ebrahim Mamdani adalah yang pertama kali mengusulkan metode ini di tahun 1975 ketika membangun sistem kontrol mesin uap dan boiler. Mamdani menggunakan sekumpulan IF-THEN rule dan diperoleh dari operator/pakar yang berpengalaman. Mamdani sering juga dikenal dengan nama Metode Max – Min (Maryam et al., 2021).

Metode FIS Mamdani dapat disebut juga sebagai metode Min-Max. Dalam metode ini, setiap aturan berbentuk implikasi. Pada setiap aturan implikasi di metode Mamdani, anteseden akan mempunyai nilai keanggotaan berbentuk minimum (min), sedangkan konsekuennya akan memiliki nilai keanggotaan berbentuk maksimum (max). Menurut Kusumadewi dan Purnomo, dalam menyelesaikan permasalahan menggunakan metode FIS Mamdani, terdapat beberapa tahapan yang perlu dilakukan. Pertama, membentuk himpunan Fuzzy untuk setiap variabel beserta domainnya masing-masing. Kedua, membuat aturan implikasi yang mengkombinasikan semua variabel yang digunakan. Ketiga, melakukan komposisi aturan dengan menggunakan metode maksimum untuk mendapatkan Fuzzy tunggal. Keempat, melakukan proses defuzzifikasi yakni memecahkan besaran Fuzzy ke dalam bentuk nilai tegas (Mugirahayu et al., 2021)

1.2.Desain FIS-Mamdani



Gambar 1 Diagram FIS Mamdani

Pembentukan variabel Input, himpunan fuzzy, dan output fuzzy Variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan fuzzy Derajat keanggotaan Menentukan derajat keanggotaan berdasarkan input dan himpunan fuzzy Aplikasi Operator fuzzy Pada tahap ini menentukan α -predikat aturan dengan fungsi implikasi MIN dan selanjutnya menentukan nilai dari Z masing-masing aturan. Penegasan (defuzzy) Input dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Defuzzifikasi yang digunakan adalah Metode Centroid (Composite Moment), solusi crisp diperoleh dengan cara mengambil titik pusat (z^*) daerah fuzzy.

1.3.Studi Kasus FIS-Mamdani

Untuk studi kasus yang kami gunakan ialah studi kasus sistem rekomendasi game menggunakan Fuzzy Inference System (FIS) dengan metode Mamdani, berdasarkan harga dan rating game. Sistem ini akan membantu pengguna memilih game yang sesuai dengan preferensi mereka berdasarkan dua kriteria utama: harga dan rating(Klau et al., 2023).

a. Deskripsi Masalah

Pengguna seringkali mengalami kesulitan dalam memilih game yang tepat karena banyaknya pilihan dengan variasi harga dan rating. Sistem rekomendasi ini akan membantu pengguna dengan mempertimbangkan kedua faktor ini untuk memberikan rekomendasi game yang sesuai.

b. Pendekatan Fuzzy Inference System (FIS) – Mamdani

FIS-Mamdani adalah salah satu metode yang sering digunakan dalam sistem fuzzy untuk mengelola ketidakpastian dan informasi yang tidak jelas. Dalam konteks sistem rekomendasi ini, kami akan menggunakan FIS-Mamdani untuk memproses input

berupa harga dan rating game, dan menghasilkan output berupa rekomendasi tingkat kepuasan pengguna terhadap game.

c. Fuzzyfication

Proses ini mengubah input numerik menjadi nilai fuzzy. Kita akan mendefinisikan beberapa variabel fuzzy untuk harga dan rating game serta output yang dihasilkan adalah rekomendasi game :

➤ Input

Harga :

- Rendah : 0 – 1000
- Sedang : 500 – 2000
- Tinggi : 1250 – 2000

Rating :

- Rendah : 0 – 25
- Sedang : 25 – 100
- Tinggi : 62,5 – 100

➤ Output

Rekomendasi :

- Rendah : 0 – 25
- Sedang : 25 – 75
- Tinggi : 50 - 100

d. Rule Base (Basis Aturan)

Aturan fuzzy akan ditetapkan untuk menghubungkan harga dan rating dengan tingkat rekomendasi. Contoh aturan fuzzy:

1. IF Harga Rendah AND Rating Rendah, THEN Rekomendasi Rendah.
2. IF Harga Rendah AND Rating Sedang, THEN Rekomendasi Rendah.
3. IF Harga Rendah AND Rating Tinggi, THEN Rekomendasi Sedang..
4. IF Harga Sedang AND Rating Rendah, THEN Rekomendasi Rendah.
5. IF Harga Sedang AND Rating Sedang, THEN Rekomendasi Sedang.
6. IF Harga Sedang AND Rating Tinggi, THEN Rekomendasi Tinggi.

7. IF Harga Tinggi AND Rating Rendah, THEN Rekomendasi Sedang.
8. IF Harga Tinggi AND Rating Sedang, THEN Rekomendasi Tinggi.
9. IF Harga Tinggi AND Rating Tinggi, THEN Rekomendasi Tinggi.

e. Inference Engine

Inference engine akan menggabungkan aturan-aturan fuzzy yang telah ditetapkan untuk menghasilkan hasil fuzzy. Proses ini melibatkan penerapan operator logika fuzzy (seperti AND, OR) pada aturan-aturan fuzzy.

f. *Defuzzification*

Proses ini mengubah hasil fuzzy menjadi nilai crisp atau nilai numerik yang bisa dipahami. Untuk sistem rekomendasi ini, hasil defuzzification akan memberikan skor rekomendasi yang dapat digunakan untuk mengurutkan atau merekomendasikan game.

- Skor Rekomendasi: 0 (Buruk) hingga 100 (Sangat Baik).

g. Implementasi

Implementasi dari sistem ini dapat dilakukan dengan langkah-langkah berikut:

- Pengumpulan Data: Kumpulkan data harga dan rating dari berbagai game.
- Pembuatan Model Fuzzy: Gunakan perangkat lunak FIS seperti MATLAB atau Python dengan pustaka scikit-fuzzy untuk membuat model fuzzy berdasarkan aturan yang telah ditetapkan.
- Pengujian dan Validasi: Uji sistem dengan data game yang berbeda untuk memastikan akurasi rekomendasi.

h. Contoh Kasus

Misalnya, kita memiliki dua game dengan data berikut:

- Game A: Harga 550 , Rating 65
- Game B: Harga 1500, Rating 15

Proses fuzzyfication akan mengkategorikan game berdasarkan harga dan rating. Misalnya, Game A bisa dikategorikan sebagai "Harga Sedang" dan "Rating Tinggi", sehingga rekomendasinya adalah "Sangat Baik". Sebaliknya, Game B bisa dikategorikan sebagai "Harga Mahal" dan "Rating Rendah", sehingga rekomendasinya adalah "Buruk".

Dengan menggunakan FIS-Mamdani, kita dapat memberikan rekomendasi yang lebih sesuai dengan preferensi pengguna dengan mempertimbangkan ketidakpastian dan variasi dalam harga dan rating game.

1.4. Susun Variabel Input dan Output studi kasus *FIS-Mamdani*

Berdasarkan studi kasus yang sudah kami gunakan kami memiliki data mengenai kasus ini sesuai dengan metode *FIS-Mamdani*

Nama Game	Input		Output
	Harga	Rating	Rekomendasi
Final Fantasy VII Remake Intergrade	1029	90	72.3879
Watch_Dogs 2	928	93	69.2669
Sekiro™: Shadows Die Twice - GOTY Edition	891	90	64.9333
Marvel's Spider-Man Remastered	879	90	64.1404
The Last of Us™ Part I	879	90	64.1404
Call of Duty: Black Ops III	891	84	62.6638
Madden NFL 24	759	97	62.5792
STAR WARS Jedi: Survivor™	759	97	62.5792
Mortal Kombat 11 Ultimate	849	89	61.7208
Horizon Zero Dawn™ Complete Edition	729	97	60.7033
Resident Evil 4	830	87	59.8307
Street Fighter™ 6	830	87	59.8307
Persona 5 Royal	798	91	58.9068
Hogwarts Legacy	799	85	57.4104
Fate/Samurai Remnant	770	87	55.8278
Red Dead Redemption 2	640	97	55.1571
NBA 2K24	659	96	55.1287
Football Manager 2024	619	98	55.1007
Starfield	759	86	55.0615
EA SPORTS FCT™ 24	759	85	55.0615
Elden Ring	599	97	52.6127
Halo: The Master Chief Collection	599	97	52.6127
Cyberpunk 2077	699	91	52.4755

ARK: Survival Evolved	715	82	51.9781
Forza Horizon 5	699	90	51.6521
Dota 2	0	99	49.0337
Apex Legends™	0	98	48.128
PUBG: BATTLEGROUNDS	0	98	48.128
Overcooked! 2	199	98	48.0601
Tom Clancy's Ghost Recon Breakpoint	619	91	48.0249
The Elder Scrolls Online	266	98	47.9985
Dead Space	660	84	47.5904
Eternal Return	0	97	47.274
Assetto Corsa	165	97	47.22
Outer Wilds	182	97	47.2042
Echoes of the Eye	449	97	46.6661
The Forest	108	96	46.4632
BeamNG.drive	119	96	46.455
Battlefield 2042	659	70	46.3691
RimWorld	219	96	46.3413
Hunt: Showdown	319	96	46.1443
NieR™	580	91	45.8664
No Man's Sky	449	96	45.7162
STAR WARS™: The Old Republic™	0	95	45.6917
Wallpaper Engine	37	95	45.6917
Left 4 Dead 2	9	95	45.6917
Icarus	284	95	45.4321
Divinity: Original Sin 2 - Definitive Edition	335	95	45.2897
Crusader Kings III	375	95	45.1537
Warhammer 40,000: Rogue Trader	375	95	45.1537
The Evil Within 2	79	94	44.951
Assassin's Creed Valhalla	620	84	44.6055
War Robots	0	93	44.2372
The Sims 4	0	93	44.2372
War Thunder	0	93	44.2372

Hades	207	93	44.1962
Guilty Gear Strive	548	91	44.0973
Monster Hunter: World	334	93	43.8192
Dave the Diver	165	92	43.5457
Monster Hunter World: Iceborne	445	93	43.2689
Path of Exile	0	91	42.8724
Risk of Rain 2	119	90	42.2137
Valheim	108	90	42.2137
Sifu	229	90	42.2137
Terraria	90	90	42.2137
Alex Jones: NWO Wars	183	90	42.2137
Warframe	0	90	42.2137
Portal 2	9	90	42.2137
The Witcher 3: Wild Hunt	359	90	41.8056
Hitman 3	245	89	41.5661
Rust	289	89	41.5661
Stray	200	89	41.5661
Ready or Not	255	88	40.9264
Age of Empires II: Definitive Edition	265	88	40.9264
Overcooked! 2	199	88	40.9264
Project Zomboid	165	88	40.9264
Against the Storm	199	87	40.2913
Satisfactory	245	87	40.2913
Mount & Blade II: Bannerlord	549	80	39.708
Atomic Heart	549	80	39.708
Don't Starve Together	95	86	39.6579
Yu-Gi-Oh! Master Duel	0	86	39.6579
Sons Of The Forest	245	86	39.6579
Planet Zoo	440	87	39.6391
Russian Fishing 4	0	85	39.023
Squad	209	85	39.023
Destiny 2	0	85	39.023

Party Animals	205	85	39.023
Far Cry 5	92	85	39.023
American Truck Simulator	175	85	39.023
theHunter: Call of the Wild™	165	85	39.023
Farming Simulator 22	341	85	39.023
Torchlight: Infinite	0	85	39.023
Tiny Tina's Wonderlands	165	85	39.023
Cities: Skylines II	536	80	38.8574
Warhammer 40,000: Darktide	439	85	38.7044
Conan Exiles	249	84	38.3835
Call of Duty®: Warzone™	0	84	38.3835
Medieval Dynasty	250	84	38.3835
Dead by Daylight	149	83	37.7362
Sea of Thieves 2023 Edition	259	83	37.7362
The Finals	0	83	37.7362
Battlefield V	569	70	37.476
SMITE	0	82	37.0776
Cyberpunk 2077: Phantom Liberty	349	81	36.4041
Sunkenland	165	80	35.7117
Grounded	460	80	35.7117
BattleBit Remastered	151	80	35.7117
DREDGE	206	80	35.7117
Deep Rock Galactic	139	80	35.7117
Fallout 76	494	75	33.6806
Tom Clancy's Rainbow Six Siege	205	77	33.4744
Steam Deck Docking Station	0	0	8.33333

1.5. Kontruksi Penalaran IF – THEN sesuai studi kasus *FIS - Mamdani*

Dari variabel input dan output yang diamati bahwa konstruksi penalaran studi kasus ini ialah:

1. IF Harga Rendah AND Rating Rendah, THEN Rekomendasi Rendah.
2. IF Harga Rendah AND Rating Sedang, THEN Rekomendasi Rendah.

3. IF Harga Rendah AND Rating Tinggi, THEN Rekomendasi Sedang.
4. IF Harga Sedang AND Rating Rendah, THEN Rekomendasi Rendah.
5. IF Harga Sedang AND Rating Sedang, THEN Rekomendasi Sedang.
6. IF Harga Sedang AND Rating Tinggi, THEN Rekomendasi Tinggi.
7. IF Harga Tinggi AND Rating Rendah, THEN Rekomendasi Sedang.
8. IF Harga Tinggi AND Rating Sedang, THEN Rekomendasi Tinggi.
9. IF Harga Tinggi AND Rating Tinggi, THEN Rekomendasi Tinggi.

1.6. Kesimpulan, Evaluasi hasil studi kasus dan karakteristik FIS - Mamdani

Kesimpulan

Dalam laporan ini, kami telah membahas penerapan Fuzzy Inference System (FIS) dengan metode Mamdani untuk sistem rekomendasi game berdasarkan dua variabel utama: harga dan rating. Melalui penggunaan logika fuzzy, sistem ini mampu mengatasi ketidakpastian dan variasi dalam data, memberikan rekomendasi yang lebih sesuai dengan preferensi pengguna. Dengan menggabungkan aturan-aturan fuzzy yang telah ditetapkan, sistem ini menghasilkan skor rekomendasi yang dapat digunakan untuk mengurutkan game dari yang paling direkomendasikan hingga yang kurang direkomendasikan.

Evaluasi Hasil Studi Kasus

Hasil dari studi kasus menunjukkan bahwa sistem rekomendasi yang dibangun dapat memberikan rekomendasi yang relevan berdasarkan data harga dan rating game. Contoh kasus yang diujikan, seperti Game A dan Game B, menunjukkan bagaimana sistem dapat mengkategorikan game dengan tepat dan memberikan rekomendasi yang sesuai. Misalnya, Game A dengan harga sedang dan rating tinggi mendapatkan rekomendasi "Sangat Baik", sedangkan Game B dengan harga tinggi dan rating rendah mendapatkan rekomendasi "Buruk". Ini menunjukkan bahwa sistem dapat berfungsi dengan baik dalam memberikan rekomendasi yang bermanfaat bagi pengguna.

Karakteristik FIS-Mamdani

FIS-Mamdani memiliki beberapa karakteristik yang membuatnya sangat cocok untuk aplikasi seperti sistem rekomendasi ini:

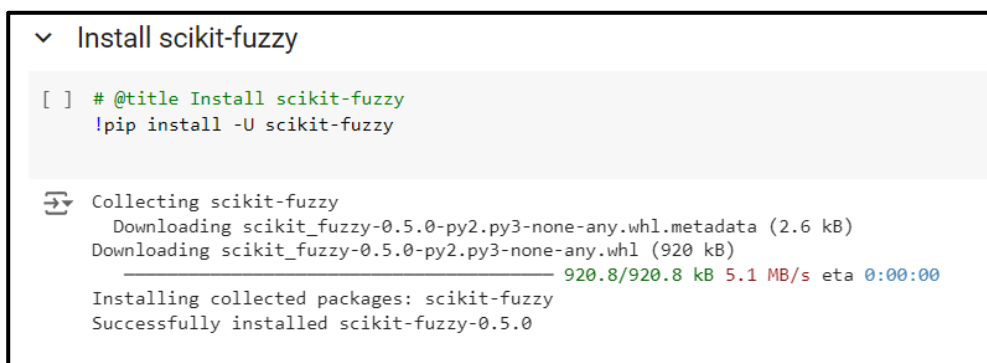
Sederhana dan Mudah Dipahami: Metode ini menggunakan aturan IF-THEN yang intuitif, sehingga mudah dipahami oleh pengguna dan pengembang.

1. Fleksibilitas: FIS-Mamdani dapat dengan mudah disesuaikan dengan berbagai jenis data dan variabel, memungkinkan pengguna untuk menambahkan atau mengubah aturan sesuai kebutuhan.
2. Toleransi terhadap Ketidakpastian: Metode ini mampu menangani data yang tidak tepat atau tidak lengkap, yang sering terjadi dalam situasi dunia nyata.
3. Kemampuan Memodelkan Fungsi Non-Linier: FIS-Mamdani dapat memodelkan hubungan yang kompleks antara input dan output, yang sangat berguna dalam sistem rekomendasi yang melibatkan banyak variabel.
4. Dengan karakteristik ini, FIS-Mamdani terbukti menjadi alat yang efektif dalam mengembangkan sistem rekomendasi yang dapat membantu pengguna dalam membuat keputusan yang lebih baik berdasarkan preferensi mereka.

1.7. Lampiran

Dalam studi kasus yang kami lakukan bahwa kami disini melakukan sebuah pengamatan berupa kami disini mencoba untuk menjelaskan sebuah sistem rekomendasi game menggunakan dataset dan fungsi dari *fuzzy* yakni, *FIS-Mamdani* yang dikerjakan menggunakan *google colab (python)* dan *matlab* sebagai acuan dalam studi kasus ini. Berikut penjelesan studi kasus menggunakan *python*

1. Install *scikit-fuzzy*



```

▼ Install scikit-fuzzy

[ ] # @title Install scikit-fuzzy
    !pip install -U 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 5.1 MB/s eta 0:00:00
Installing collected packages: scikit-fuzzy
Successfully installed scikit-fuzzy-0.5.0
```

Gambar 2 *Install scikit-fuzzy*

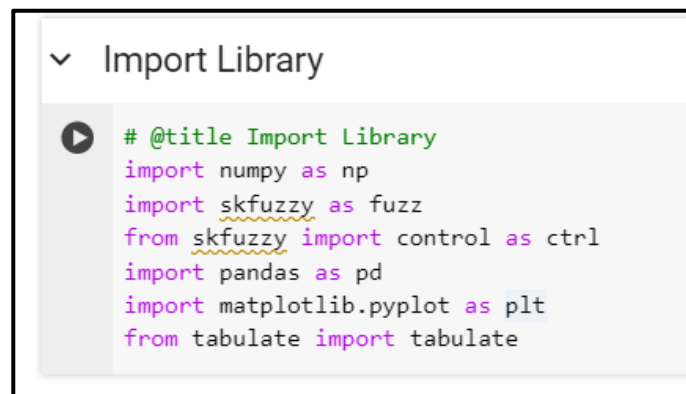
Pada saat kita ingin menggunakan sebuah logika *fuzzy* disini kita harus menginstal library *python* yakni, *scikit-learn* disini menggunakan `!pip install -U scikit-fuzzy` yang dimana:

!: merupkn perintah untuk mengitall di google colab

pip : merupakan ppackage manager untuk *python*

-U *scikit-fuzzy* : merupakan logika fuzzy yang ingin diinstal

2. Import Library

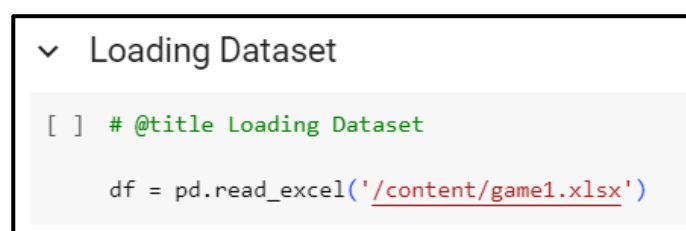


Gambar 3 *Import Library*

Kode program ini digunakan untuk mengimport berbagai *library* yang ada di *python* dijelaskan bahwa:

1. Import numpy as np: *import* numpy untuk menyediakan array besar multidimensi dan fungsi matematika yang efisien.
2. Import skfuzzy as fuzz: *import* pustaka komputasi logika *fuzzy* di Python. *Fuzz* adalah alias yang digunakan untuk memudahkan referensi yang sesuai.
3. From skfuzzy import control as ctrl: Mengimpor modul control dari *skfuzzy* yang digunakan untuk membangun dan mengontrol sistem *fuzzy logic*
4. Import panda as pd: Mengimpor Pandas, pustaka untuk manipulasi dan analisis data, khususnya dengan struktur data.
5. Import matplotlib.pyplot as plt: digunakan untuk, pustaka untuk membuat visualisasi data. Pyplot digunakan untuk membuat plot secara interaktif, dengan plt sebagai alias
6. from tabulate import tabulate: mengimpor fungsi tabulate dari pustaka Tabulate, yang digunakan untuk menampilkan data dalam format tabel yang mudah dibaca, misalnya dalam terminal atau output teks.

3. Loading Dataset



Gambar 4 *Loading Dataset*

4. Inisialisasi Variabel Input dan Output

▼ Inisialisasi Variabel Input dan Output

```
[ ] # @title Inisialisasi Variabel Input dan Output

# Input variables
harga = ctrl.Antecedent(np.arange(0, 2001, 1), 'Harga')
rating = ctrl.Antecedent(np.arange(0, 100, 1), 'Rating')

# Output variable
rekomendasi = ctrl.Consequent(np.arange(0, 101, 1), 'rekomendasi')
```

Gambar 5 Inisialisasi Variabel Input dan Output

Dari kode program diatas dapat kita ketahui bahwa disini kita akan melakukan inisialisasi variabel input dan output dalam *fuzzy logic*.

1. `harga = ctrl.Antecedent(np.arange(0, 2001, 1), 'Harga')`

Didefinisikan bahwa variabel input harga memiliki nilai dengan *range* 0 hingga 2000, dengan interval 1

2. `rating = ctrl.Antecedent(np.arange(0, 100, 1), 'Rating')`

Didefinisikan sebagai Rating dengan *range* nilai 0 hingga 100, dengan interval 1.

3. `rekomendasi = ctrl.Consequent(np.arange(0, 101, 1), 'rekomendasi')`

Didefinisikan bahwa variabel output rekomendasi memiliki range 0 sampai dengan 100, yang merupakan hasil akhir dari proses *fuzzy*

5. Definisi Fungsi Keanggotaan (*Membership Functions*)

▼ Definisi Fungsi Keanggotaan (Membership Functions)

```
[ ] # @title Definisi Fungsi Keanggotaan (Membership Functions)

# Membership Functions
harga['rendah'] = fuzz.trimf(harga.universe, [0, 0, 1000])
harga['sedang'] = fuzz.trimf(harga.universe, [500, 1250, 2000])
harga['tinggi'] = fuzz.trimf(harga.universe, [1250, 2000, 2000])

rating['rendah'] = fuzz.trimf(rating.universe, [0, 0, 25])
rating['sedang'] = fuzz.trimf(rating.universe, [25, 62.5, 100])
rating['tinggi'] = fuzz.trimf(rating.universe, [62.5, 100, 100])

rekomendasi['rendah'] = fuzz.trimf(rekomendasi.universe, [0, 0, 25])
rekomendasi['sedang'] = fuzz.trimf(rekomendasi.universe, [25, 50, 75])
rekomendasi['tinggi'] = fuzz.trimf(rekomendasi.universe, [50, 100, 100])
```

Gambar 6 Definisi Fungsi Keanggotaan

Kode program tersebut membahas tentang mendefinisikan keanggotaan untuk *fuzzy logic* yang digunakan untuk mengklasifikasi tiga variabel yakni, harga, rating, rekomendasi. Fungsi ini juga membantu untuk mengelompokkan setiap variabel seperti “rendah”, “sedang” dan “tinggi menggunakan kuala bentuk kurva tirangular (trimf).

6. Aturann Sistem Fuzzy (*Rule Base*)

```
Aturan Sistem Fuzzy (Rule Base)

# @title Aturan Sistem Fuzzy (Rule Base)

# Rule base
rule1 = ctrl.Rule(harga['rendah'] & rating['rendah'], rekomendasi['rendah'])
rule2 = ctrl.Rule(harga['rendah'] & rating['sedang'], rekomendasi['rendah'])
rule3 = ctrl.Rule(harga['rendah'] & rating['tinggi'], rekomendasi['sedang'])

rule4 = ctrl.Rule(harga['sedang'] & rating['rendah'], rekomendasi['rendah'])
rule5 = ctrl.Rule(harga['sedang'] & rating['sedang'], rekomendasi['sedang'])
rule6 = ctrl.Rule(harga['sedang'] & rating['tinggi'], rekomendasi['tinggi'])

rule7 = ctrl.Rule(harga['tinggi'] & rating['rendah'], rekomendasi['sedang'])
rule8 = ctrl.Rule(harga['tinggi'] & rating['sedang'], rekomendasi['tinggi'])
rule9 = ctrl.Rule(harga['tinggi'] & rating['tinggi'], rekomendasi['tinggi'])
```

Gambar 7 Aturan Sistem *Fuzzy* (*Rule Base*)

Kode program tersebut mendefinisikan tentang *rule base* yang menghubungkan dua variabel input, yaitu harga dan rating, dengan variabel output rekomendasi. Dapat dilihat bahwa aturan ini membentuk *fuzzy logic* berdasarkan kombinasi nilai *fuzzy* dari harga dan rating untuk menentukan hasil rekomendasi yang sesuai.

7. Pembuatan dan Simulasi Sistem Kontrol Fuzzy (*Control System*)

```
Pembuatan dan Simulasi Sistem Kontrol Fuzzy (Control System)

# @title Pembuatan dan Simulasi Sistem Kontrol Fuzzy (Control System)

rekendasi_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9])
rekendasi_simulasi = ctrl.ControlSystemSimulation(rekendasi_ctrl)
```

Gambar 8 *Control System*

Kode program ini akan diminta untuk memasukkan *rule base* yang dibuat sebelumnya. Kemudian disimulasikan untuk membuat simulasi dari sistem kontrol *fuzzy* yang dibangun untuk melakukan perhitungan berdasarkan input (harga dan rating) yang akan dimasukkan ke sistem fuzzy ini, kemudian menghasilkan output berupa rekomendasi

8. Inferensi Fuzzy Untuk Setiap Data pada Dataset (*Fuzzy Inference*)

```
Inferensi Fuzzy untuk Setiap Data pada Dataset (Fuzzy Inference)

[ ] # @title Inferensi Fuzzy untuk Setiap Data pada Dataset (Fuzzy Inference)

for idx, row in df.iterrows():
    rekomendasi_simulasi.input['Harga'] = row['Harga']
    rekomendasi_simulasi.input['Rating'] = row['Rating']
    rekomendasi_simulasi.compute()
    df.at[idx, 'rekomendasi'] = rekomendasi_simulasi.output['rekomendasi']
```

Gambar 9 *Fuzzy Inference*

Pada kode program sebelumnya disini akan diminta inferensi *fuzzy* untuk setiap data dalam sebuah dataset dengan menggunakan sistem kontrol fuzzy yang telah dibuat yang dimana akan dimulai dari iterasi melalui dataset, kemudian memberikan input ke sistem *fuzzy*, lalu dilakukan komputasi perhitungan *fuzzy*, yang terakhir akan dilakukan menyimpan hasil rekomendasiya.

9. Validasi dan Pemrosesan Hasil dari Dataset

```
✓ Validasi dan Pemrosesan Hasil dari Dataset

# @title Validasi dan Pemrosesan Hasil dari Dataset

# Print panjang variabel
print("Panjang rekomendasi.universe:", len(rekomendasi.universe))
print("Panjang df['rekomendasi'].values:", len(df['rekomendasi'].values))

# Pastikan tidak ada nilai NaN di kolom 'rekomendasi'
df['rekomendasi'].fillna(0, inplace=True)

# Sort DataFrame based on the 'rekomendasi' column in descending order
df_sorted = df.sort_values(by='rekomendasi', ascending=False)

⇒ Panjang rekomendasi.universe: 101
   Panjang df['rekomendasi'].values: 113
```

Gambar 10 Validasi

Pada kode program ini dapat dilihat bahwa akan melakukan print panjang variabel, mengatasi nilai NaN di kolom rekomendasi, serta mengurutkan data berdasarkan rekomendasi sesuai dengan DataFrame. Yang dimana untuk outputnya dapat dilihat di gambar 10

10. Menampilkan hasil dalam tabel

```
✓ Menampilkan hasil dalam tabel

# @title Menampilkan hasil dalam tabel

# Tampilkan tabel hasil dengan tabulate
table_headers = ['Nama Game', 'Harga', 'Rating', 'Rekomendasi']
table_data_sorted = []

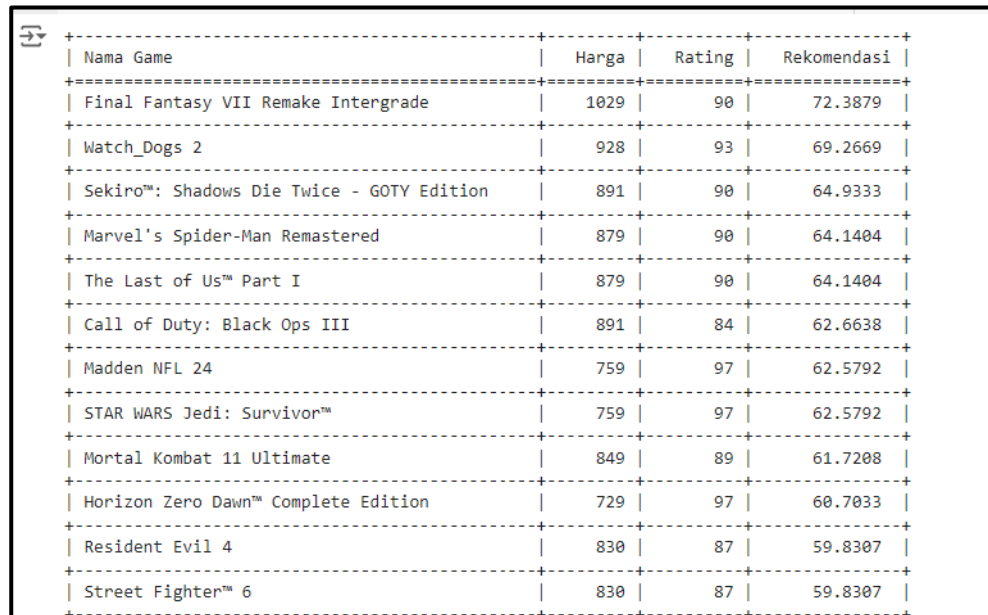
for idx, row in df_sorted.iterrows():
    table_data_sorted.append([row['Nama Game'], row['Harga'], row['Rating'], row['rekomendasi']])

table_str_sorted = tabulate(table_data_sorted, headers=table_headers, tablefmt='grid')
print(table_str_sorted)
```

Gambar 11 Kode Program Menampilkan hasil tabel

Disini untuk mengisi data tabel akan diambil dari dataframe yang sudah dilakukan penginputan sebelumnya. Pada data ini akan dilakukan untuk mengiterasi setiap

baris dalam dataframe seperti nama game harga, rating, dan rekomendasi yang dimasukkan ke dalam list tersebut.



Nama Game	Harga	Rating	Rekomendasi
Final Fantasy VII Remake Intergrade	1029	90	72.3879
Watch_Dogs 2	928	93	69.2669
Sekiro™: Shadows Die Twice - GOTY Edition	891	90	64.9333
Marvel's Spider-Man Remastered	879	90	64.1404
The Last of Us™ Part I	879	90	64.1404
Call of Duty: Black Ops III	891	84	62.6638
Madden NFL 24	759	97	62.5792
STAR WARS Jedi: Survivor™	759	97	62.5792
Mortal Kombat 11 Ultimate	849	89	61.7208
Horizon Zero Dawn™ Complete Edition	729	97	60.7033
Resident Evil 4	830	87	59.8307
Street Fighter™ 6	830	87	59.8307

Gambar 12 Output Menampilkan Tabel

Untuk output yang dapat dilihat lebih jelas dapat dilihat pada bagian 1.4 Susun Variabel Input dan Output studi kasus *FIS-Mamdani*.

11. Defuzifikasi dan Hasil Penentuan Hasil Akhir



```

# @title Defuzifikasi dan Penentuan Hasil Akhir

# Defuzzification
if len(rekomendasi.universe) == len(df['rekomendasi'].values):
    centroid = fuzz.defuzz(rekomendasi.universe, df['rekomendasi'].values, 'centroid')

# Output recommendation
if centroid <= 25:
    hasil_rekomendasi = 'Rekomendasi Rendah'
if centroid > 25:
    if centroid <= 75:
        hasil_rekomendasi = 'Rekomendasi Sedang'
    if centroid > 75:
        hasil_rekomendasi = 'Rekomendasi Tinggi'

# Print hasil rekomendasi
print(f'Hasil Rekomendasi: {hasil_rekomendasi}')

```

Gambar 13 Defuzifikasi dan Penentuan Hasil Akhir

Dari kode program diatas dapat dilihat bahwa pada kode program rekomendasi.universe sama dengan panjang nilai df['rekomendasi'].values untuk memastikan konsistensi data. Jika ya, defuzifikasi centroid digunakan untuk menghitung nilai pusat dari distribusi fuzzy rekomendasi.

12. Visualisasi Fungsi Keanggotaan

```
Visualisasi Fungsi Keanggotaan

[ ] # @title Visualisasi Fungsi Keanggotaan

# Plot the membership functions
fig, (ax0, ax1, ax2) = plt.subplots(nrows=3, figsize=(8, 9))

ax0.plot(harga.universe, fuzz.trimf(harga.universe, [0, 0, 1000]), 'r', linewidth=2, label='Rendah')
ax0.plot(harga.universe, fuzz.trimf(harga.universe, [500, 1250, 2000]), 'g', linewidth=2, label='Sedang')
ax0.plot(harga.universe, fuzz.trimf(harga.universe, [1250, 2000, 2000]), 'b', linewidth=2, label='Tinggi')
ax0.set_title('Harga')
ax0.legend()

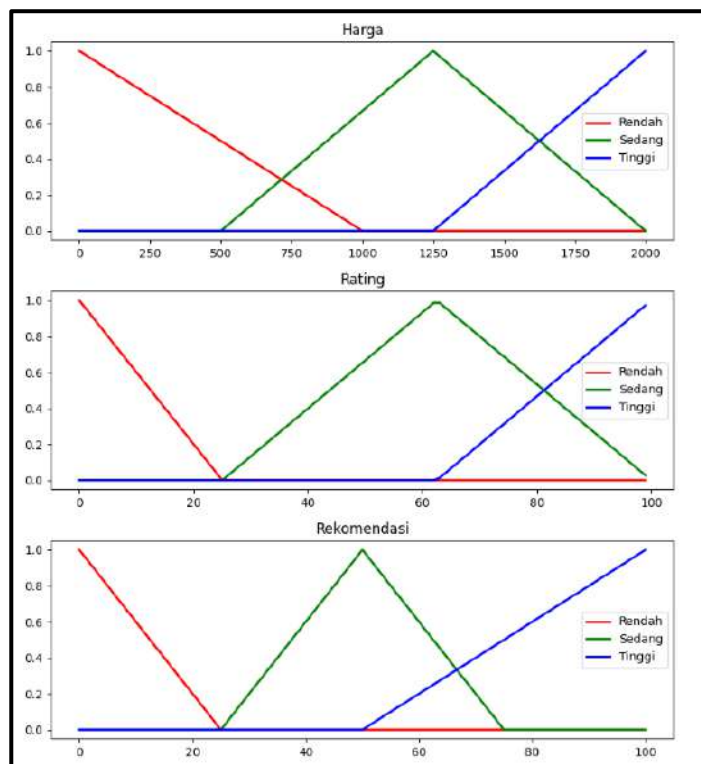
ax1.plot(rating.universe, fuzz.trimf(rating.universe, [0, 0, 25]), 'r', linewidth=2, label='Rendah')
ax1.plot(rating.universe, fuzz.trimf(rating.universe, [25, 62.5, 100]), 'g', linewidth=2, label='Sedang')
ax1.plot(rating.universe, fuzz.trimf(rating.universe, [62.5, 100, 100]), 'b', linewidth=2, label='Tinggi')
ax1.set_title('Rating')
ax1.legend()

ax2.plot(rekomendasi.universe, fuzz.trimf(rekomendasi.universe, [0, 0, 25]), 'r', linewidth=2, label='Rendah')
ax2.plot(rekomendasi.universe, fuzz.trimf(rekomendasi.universe, [25, 50, 75]), 'g', linewidth=2, label='Sedang')
ax2.plot(rekomendasi.universe, fuzz.trimf(rekomendasi.universe, [50, 100, 100]), 'b', linewidth=2, label='Tinggi')
ax2.set_title('Rekomendasi')
ax2.legend()

plt.tight_layout()
plt.show()
```

Gambar 14 Visualisasi Fungsi Keanggotaan

Kode program ini akan membentuk sebuah fungsi keanggotaan terhadap masing-masing data yakni harga, rating, dan rekomendasi dimana menggunakan sistem *fuzzy* matplotlib. Yang dimana akan diimplementasikan setiap subplot aan membentuk fungsi keanggotaannya berbeda-beda yang dapat dilihat dari gambar berikut



Gambar 15 Fungsi Keanggotaan dari Studi Kasus tersebut

Dapat dilihat bahwa fungsi keanggotaan ketika menggunakan Google Colab merupakan fungsi keanggotaan bahu.

Selain menggunakan google colab kami disini menggunakan matlab juga sebagai perhitungan kami pada kasus ini yang dimana ketika kita akan membuat fungsi matematikanya berupa:

a. Harga

1. Rendah

$$\mu \text{ Rendah } [x] = \begin{cases} 1 & \rightarrow x \leq 1000 \\ \frac{1000-x}{1000-0} & \rightarrow 0 \leq x \leq 1000 \\ 0 & \rightarrow x \geq 1000 \end{cases}$$

2. Sedang

$$\mu \text{ Sedang } [x] = \begin{cases} 0 & \rightarrow x \leq 500 \text{ atau } x \geq 2000 \\ \frac{x-500}{1250-500} & \rightarrow 500 \leq x \leq 1250 \\ \frac{2000-x}{2000-1250} & \rightarrow 1250 \leq x \leq 2000 \end{cases}$$

3. Tinggi

$$\mu \text{ Tinggi } [x] = \begin{cases} 0 & \rightarrow x \leq 1250 \\ \frac{x-1250}{2000-1250} & \rightarrow 0 \leq x \leq 100 \\ 1 & \rightarrow x \geq 2000 \end{cases}$$

b. Rating

1. Rendah

$$\mu \text{ Rendah } [x] = \begin{cases} 0 & \rightarrow x \leq 25 \\ \frac{25-x}{25-0} & \rightarrow 0 \leq x \leq 25 \\ 0 & \rightarrow x \geq 25 \end{cases}$$

2. Sedang

$$\mu \text{ Sedang } [x] = \begin{cases} 0 & \rightarrow x \leq 25 \text{ atau } x \geq 100 \\ \frac{x-25}{62.5-25} & \rightarrow 25 \leq x \leq 62.5 \\ \frac{100-x}{100-62.5} & \rightarrow 62.5 \leq x \leq 100 \end{cases}$$

3. Tinggi

$$\mu \text{ Tinggi } [x] = \begin{cases} 0 & \rightarrow x \leq 62.5 \\ \frac{x-62.5}{100-62.5} & \rightarrow 62.5 \leq x \leq 100 \\ 1 & \rightarrow x \geq 100 \end{cases}$$

c. Rekomendasi

1. Rendah

$$\mu_{\text{Rendah}}[x] = \begin{cases} 1 & \rightarrow x \leq 25 \\ \frac{25-x}{25-0} & \rightarrow 0 \leq x \leq 25 \\ 0 & \rightarrow x \geq 25 \end{cases}$$

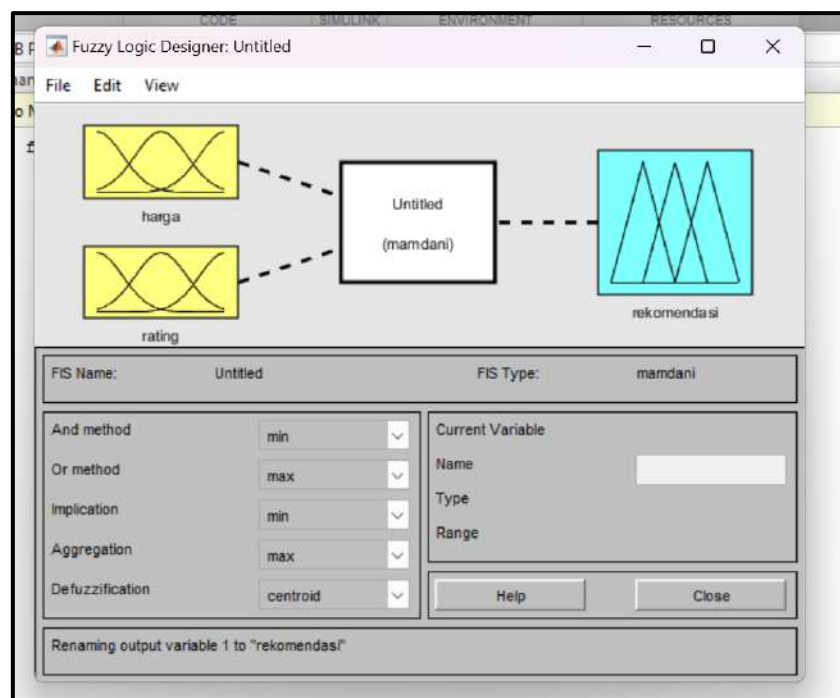
2. Sedang

$$\mu_{\text{Sedang}}[x] = \begin{cases} 0 & \rightarrow x \leq 25 \text{ atau } x \geq 75 \\ \frac{x-25}{50-25} & \rightarrow 25 \leq x \leq 50 \\ \frac{75-x}{75-50} & \rightarrow 50 \leq x \leq 75 \end{cases}$$

3. Tinggi

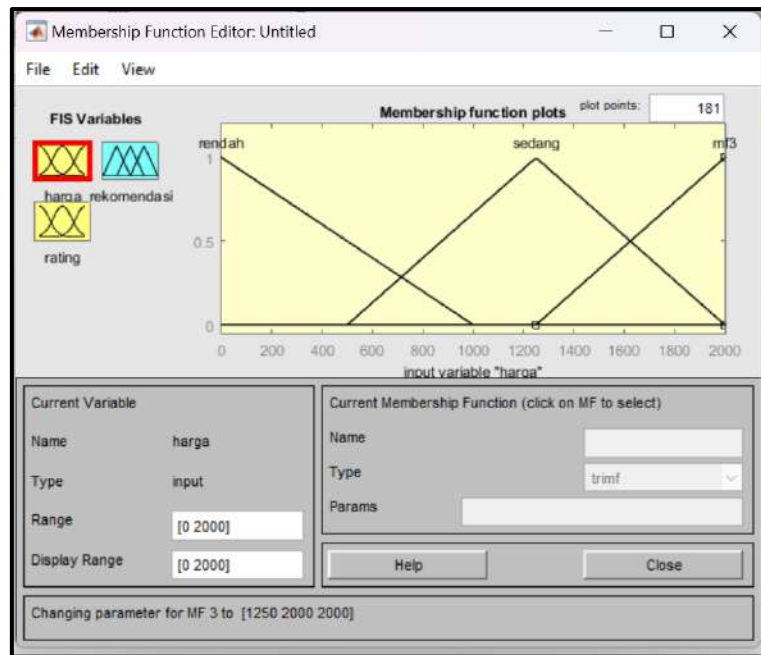
$$\mu_{\text{Tinggi}}[x] = \begin{cases} 0 & \rightarrow x \leq 50 \\ \frac{x-50}{100-50} & \rightarrow 0 \leq x \leq 100 \\ 1 & \rightarrow x \geq 100 \end{cases}$$

Ketika dijelaskann menggunakan matlab maka yang akan dilakukan diawal adalah



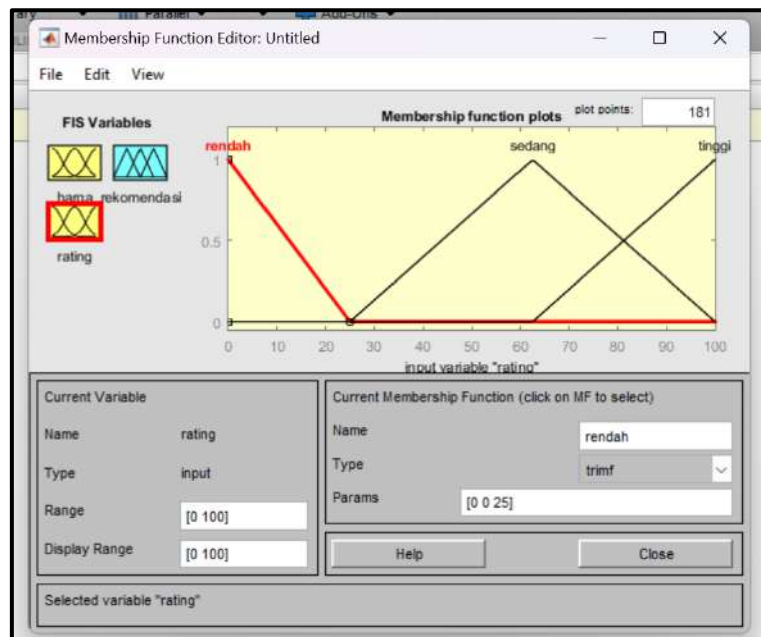
Gambar 16 Penambahan Fungsi Input dan
Menggantian Nama Input Output Sesuai dengan Kasus

Pada gambar ini kita akan melakukan penambahan sebuah grafik yang dimana gambaran diatas adalah sebuah elemen untuk menambahkan inputan sesuai dengan kasus yang kita lakukan

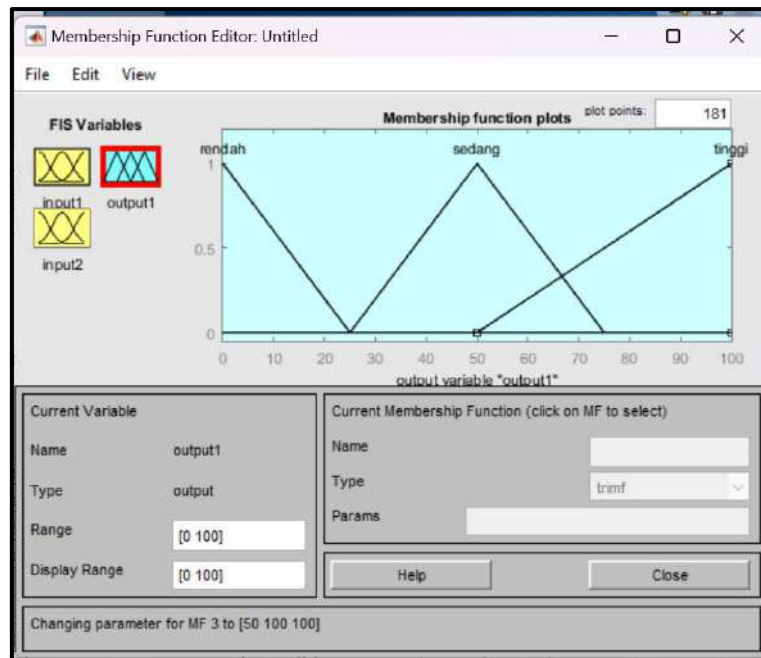


Gambar 17 Melakukan Penginputan Parameter dan Membentuk Fungsi Keanggotaan Harga

Pada gambar ini kita akan melakukan penginputan di params sebagai acuan kita atau range yang ada untuk di dalam grafik tersebut ketika kita membuat fungsi matematika sebelumnya. Lalu dari fungsi matematika tersebut akan kita input disesuaikan dengan display range kita dan kita meinput data di params ini akan dilakukan sama pada rating dan rekomendasi pada gambar 18 dan 19

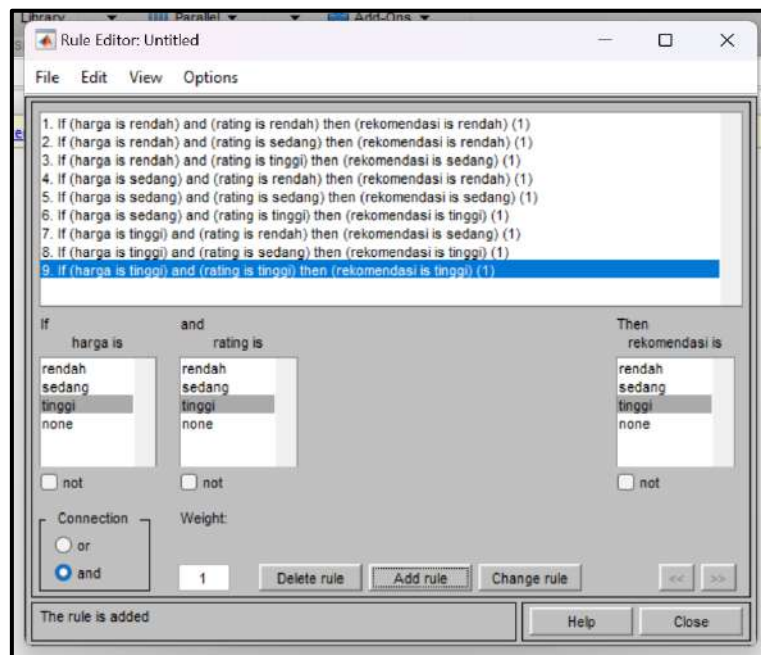


Gambar 18 Melakukan Penginputan Parameter dan Membentuk Fungsi Keanggotaan Rating



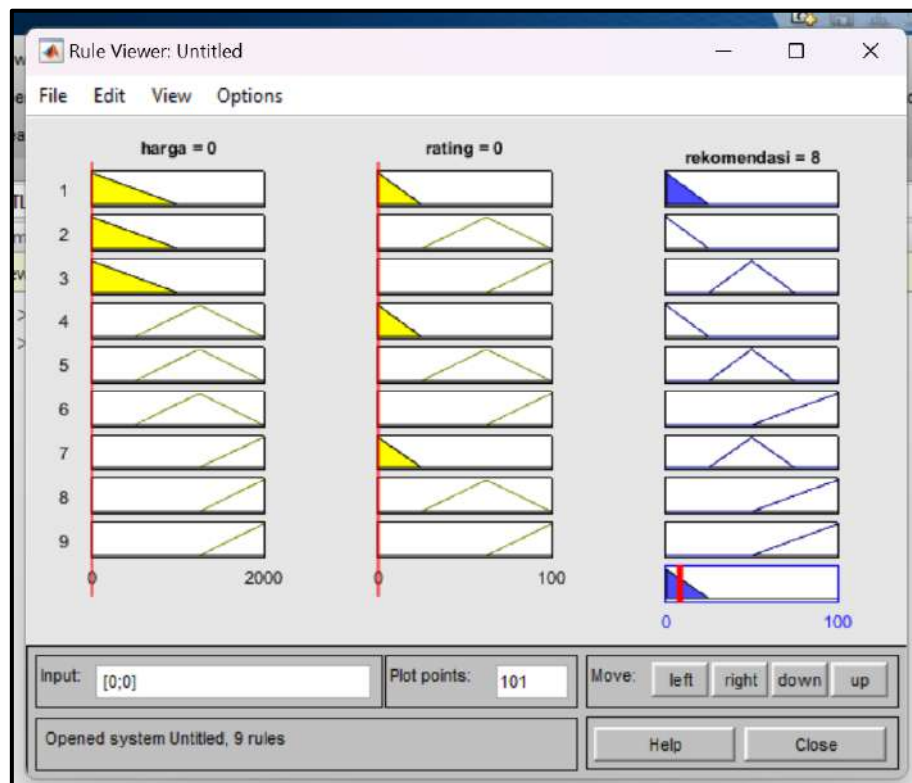
Gambar 18 Melakukan Penginputan Parameter dan Membentuk Fungsi Keanggotaan Rekomendasi

Dapat dilihat bahwa fungsi keanggotaan disini yang kami lakukan pada google colab dan kami lakukan juga pada matlab itu sama dalam bentuk fungsi keanggotaannya yakni, fungsi keanggotaan bahu. Untuk mengetahui selanjutnya kita akan memasukkan fungsi *base rule* yang ada pada matlab tersebut



Gambar 19 *Base Rull*

Disini kita akan melakukan penginputan base rull sebelum digunakan disini kita menggunakannya sama seperti yang ada pada di dalam 1.5 konstruksikan penalaran *if-then* langkah selanjutnya kita akan melakukan memasukkan data yang akan kita uji coba



Gambar 20 Data Uji Coba

Disini dapat kita lakukan uji coba disini saya menggunakan data game yakni, Steam Deck Docking Station dengan inputannya harga dan ratingnya masing-masing 0 dan untuk rekomendasi yang didapatkan adalah 8 ketika kita melihat perhitungan di google colab *python* menghasilkan angka sebesar 8.333 juga ini berarti bahwa data yang kita coba ini cocok dengan akurasi yang dikatakan cukup tinggi dalam uji coba kali ini dapat diambil kesimpulan bahwa mau menggunakan matlab ataupun menggunakan googl colab *python* itu hasilnya akan mirip dengan akurasi yang dikatakan cukup tinggi.

DAFTAR PUSTAKA

- Athiyah, U., Handayani, A. P., Aldean, M. Y., Putra, N. P., & Ramadhani, R. (2021). Sistem Inferensi Fuzzy: Pengertian, Penerapan, dan Manfaatnya. *Journal of Dinda : Data Science, Information Technology, and Data Analytics*, 1(2).
<https://doi.org/10.20895/dinda.v1i2.201>
- Klau, D. Y., Tursina, T., & Novriando, H. (2023). Implementasi Metode Fuzzy Inference System (FIS) Mamdani dalam Pemilihan Bidang Keahlian Mahasiswa. *Jurnal Impresi Indonesia*, 2(4). <https://doi.org/10.58344/jii.v2i4.2389>
- Maryam, S., Bu'ulolo, E., & Hatmi, E. (2021). Penerapan Metode Fuzzy Mamdani dan Fuzzy Tsukamoto Dalam Menentukan Harga Mobil Bekas. *Journal of Informatics, Electrical and Electronics Engineering*, 1(1).
- Mugirahayu, A. S., Linawati, L., & Setiawan, A. (2021). Penentuan Status Kewaspadaan COVID-19 Pada Suatu Wilayah Menggunakan Metode Fuzzy Inference System (FIS) Mamdani. *Jurnal Sains Dan Edukasi Sains*, 4(1).
<https://doi.org/10.24246/juses.v4i1p28-39>