REFERENCES

Abrori, M., & Prihamayu A. H. 2015. Aplikasi Metode Fuzzy Mamdani dalam Pengambilan Keputusan Penentuan Jumlah Produksi. *Kaunia*. 11(2), 92306.

Akhmad, P. 2014. Ekonomi Mikro (Teori dan Aplikasi di Dunia Usaha). Andi Offset. Yogyakarta

Bukhori, S. 2017. Development of Supply Chain Management Agribusiness Using Collaborative, Planning, Forecasting and Replenishment Concept. *Journal Advanced Science Letters*, 23(3), 2340-2343.

Bowerman, B. L., O'Connell, R. T., & Koehler, A. B. 2004. Forecasting, time series and regression: An applied approach. Thomson Brooks/Cole. Belmont.

Guttag, J. V. 2017. Introduction to Computation and Programming Using Python, second edition: With Application to Understanding Data 2nd Edition. The MIT Press. Cambridge.

Himanshu, S., & Yunis A. L. 2020. Deep Neuro-Fuzzy Systems with Python With Case Studies and Applications from the Industry. Apress. India. https://doi.org/10.1007/978-1-4842-5361-8.

Harliana, p., & Rahim, R. 2017. Comparative Analysis of Membership Function on Mamdani Fuzzy Inference System for Decision Making. *Journal of Physics: Conference Series.* Https://doi :10.1088/1742-6596/930/1/012029.

Haty, I. P. 2011. Pemanfaatan Batu Marmer Berdasarkan Analisa Kuat Tekan Dan Serapan Air Daerah Teras Kecamatan Campurdarat Kabupaten Tulungagung Propinsi Jawa Timur. *Jurnal Ilmiah MTG.* 4(2).

Irsan, M., Y., T., Kasau, M., I., & Simbolon, I., P. 2019. Penggunaan Fuzzy Logic & Metode Mamdani untuk Menghitung Pembelian, Penjualan dan Persediaan. *Journal of Applied Accounting and Finance*. 3(1), 37. https://doi.org/10.33021/ jaaf.v3i1.677

Jumingan. 2009. Studi Kelayakan Bisnis (Teori dan Pembuatan Proposal Kelayakan. Bumi Aksara. Jakarta.

Kurniawati, S., & Titisari, A. D. 2019. Rekomendasi Pemanfaatan Marmer Berdasarkan Karakteristiknya. *Jurnal Pengabdian Kepada Masyarakat.* 5(2), 251-266.

Kartika, D., Sovia, R., & Sandawa H. M. 2018. Penerapan Metode Fuzzy Mamdani Untuk Memprediksi Angka Penjualan Token Berdasarkan Persediaan Dan Jumlah Permintaan Pada Pt. Pln (Persero) Padang Berbasis Web. *UPI YPTK Jurnal KomTekInfo.* 5(1), 81-95.

Kim, S. & Kim, H. 2016. A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, 32, 669-679.

E. H. 1974. Application of fuzzy algorithms for control of simple dynamic plant. *PROC. E.* 121(12), 1585-1588.

[., Esmaeil N., Fallah M., & Farhad H. L.. 2019. Quantitative analysis of key rformance indicators of Green Supply Chain in FMCG industries using non-linear zzy method. *Mathematics.* 7(11), 1020. https://doi.org/10.3390/ math7111020.

Rahakbauw, D. L., Rianekuay, F. J., & Lesnussa Y. A. 2019. Penerapan Metode Fuzzy Mamdani Untuk Memprediksi Jumlah Produksi Karet (Studi Kasus: Data Persediaan Dan Permintaan Produksi Karet Pada Ptp Nusantara Xiv (Persero) Kebun Awaya, Teluk Elpaputih, Maluku-Indonesia). *Jurnal Ilmiah Matematika dan Terapan*, 16(1), 119-127.

Supardi, Y. 2020. Semua Bisa Menjadi Programmer Python Case Study. Kompas Gramedia. Jakarta.

Sukandy D. M., Basuki A. T., & Puspasari, S. 2014. Penerapan Metode Fuzzy Mamdani Untuk Memprediksi Jumlah Produksi Minyak Sawit Berdasarkan Data Persediaan Dan Jumlah Permintaan (Studi Kasus Pt Perkebunan Mitra Ogan Baturaja). 1-9.

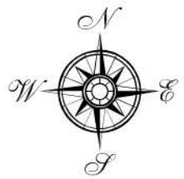
Wibowo, A. N. 2016. Analisis Strategi Pemasaraan Marmer. *SOSIO-E-KONS.* 8(3), 190-201.

Zadeh, L. A. 1965. *Fuzzy Sets. Information and Control.* 8, 338-353.

APPENDIX

APPENDIX A

WIUP MAP OF PT WUTAMA TRI MAKMUR



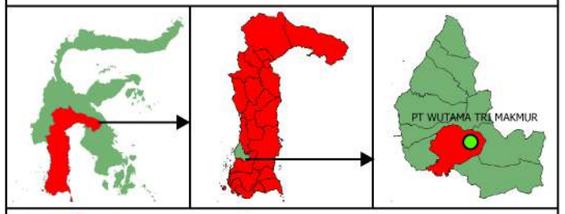
0 0,425 0,85 1,7

Kilometers

Legend:

- WIUP PT Wutama Tri Makmur
- PT Wutama Tri Makmur Factory
- Thesis Location

Cartographic Information
Unit Datum: WGS-84



MINING ENGINEERING DEPARTMENT
FACULTY OF ENGINEERING
HASANUDDIN UNIVERSITY
2024

THESIS

APPLICATION OF THE FUZZY MAMDANI
TO PREDICT THE AMOUNT OF MARBLE PRODUCTION
AT PT WUTAMA TRI MAKMUR, SOUTH SULAWESI PROVINCE

DRAWN BY	MUHRAM SULA IDRIS D111171503
SUPERVISOR	Dr. Aryanti Virtanti Anas, S.T., M.T. NIP.197010052008012026
CO-SUPERVISOR	Dr. Eng. RINI NOVRIANTI SUTARDJO TUI, S.T., M.BA., M.T. NIP. 198311142014042001

MAP SHOWING LOCATION

APPENDIX	PAGE
A	51

APPENDIX B

SUPPLY, DEMAND AND PRODUCTION DATA

Table of Production, Supply and Demand

YEAR	MONTH	PRODUCTION (M^3)	SUPPLY (M^3)	DEMAND (M^3)
2015	JANUARY	425,468	158,462	267,006
	FEBRUARY	251,806	0	410,268
	MARCH	127,963	0	127,963
	APRIL	326,785	0	326,783
	MAY	548,957	2	546,957
	JUNE	167,134	2	167,134
	JULY	307,517	2	307,517
	AUGUST	54,298	2	54,298
	SEPTEMBER	234,852	2	234,852
	OCTOBER	181,322	2	181,322
	NOVEMBER	96,531	2	96,531
	DECEMBER	480,204	47	435,204
2016	JANUARY	167,529	94,131	120,398
	FEBRUARY	168,355	94,131	168,355
	MARCH	134,565	94,131	134,565
	APRIL	241,028	94,131	241,028
	MAY	406,438	94,131	406,438
	JUNE	199,814	94,131	199,814
	JULY	224,758	94,131	224,758
	AUGUST	512,512	94,131	512,512
	SEPTEMBER	431,233	94,131	431,233
	OCTOBER	262,073	94,131	262,073
	NOVEMBER	102,754	94,131	102,754
	DECEMBER	111,504	94,131	111,504
2017	JANUARY	120,398	47	167,529
	FEBRUARY	168,355	47	168,355
	MARCH	134,565	47	134,565
	APRIL	241,028	47	241,028
	MAY	406,438	47	406,438
	JUNE	199,814	47	199,814
	JULY	224,758	47	224,758
	AUGUST	512,512	47	512,512
	SEPTEMBER	431,233	47	431,233
	OCTOBER	262,073	47	262,073
	NOVEMBER	102,754	47	102,754
	DECEMBER	111,504	47	111,504
	JANUARY	0	0	47
	FEBRUARY	0	0	0
	MARCH	0	0	0
	APRIL	0	0	0
	MAY	0	0	0

Table of Production, Supply and Demand (Continued)

YEAR	MONTH	PRODUCTION (M^3)	SUPPLY (M^3)	DEMAND (M^3)
	JUNE	0	0	0
	JULY	0	0	0
	AUGUST	0	0	0
	SEPTEMBER	0	0	0
	OCTOBER	327,314	327,314	0
	NOVEMBER	326,749	372,573	281,49
	DECEMBER	0	372,573	0
2019	JANUARY	371,427	390	354
	FEBRUARY	135,521	423,521	102
	MARCH	147,817	429,338	142
	APRIL	297,282	454,62	272
	MAY	200,668	463,288	192
	JUNE	103,404	451,692	115
	JULY	120,308	452	120
	AUGUST	202,681	411,681	243
	SEPTEMBER	140,38	444,061	108
	OCTOBER	105,873	439,934	110
	NOVEMBER	91,346	447,28	84
	DECEMBER	100	517,28	30
2020	JANUARY	107,66	537,546	87,394
	FEBRUARY	179,655	717,201	0
	MARCH	95,615	812,816	0
	APRIL	116,931	929,747	0
	MAY	120,758	1050,505	0
	JUNE	57,479	1107,984	0
	JULY	41,334	1149,318	0
	AUGUST	49,707	1199,025	0
	SEPTEMBER	73,814	1272,839	0
	OCTOBER	44,346	1271,099	46,086
	NOVEMBER	53,994	1280,624	44,469
	DECEMBER	38,994	1319,618	0
2021	JANUARY	118,555	1438,173	0
	FEBRUARY	116,196	1554,369	0

APPENDIX C

FUZZY MAMDANI CODING

```
Created on Tue Sep 28 11:03:36 2021

@author: MUHRAM SULA IDRIS
"""

#%%

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

#%%

data = pandas.read_csv("D:\ARSIP KAMPUS MUHRAM\Kampus Semester
8\TA\Anaconda\Data\Data.csv")

print (data)

#%%

x_supply = input('Supply Value (ton) = ')
y_demand = input('Demand Value (ton) = ')

     = int(x_supply)
     = int(y_demand)
```

```
x_supply = np.arange(0, 1554, 1)
x_demand = np.arange(0, 547, 1)
x_production = np.arange(0, 549, 1)


#Generate Fuzzy Membership Functions

supply_very_low = fuzz.trapmf(x_supply, [0, 0, 222, 444])
supply_low = fuzz.trimf(x_supply, [222, 444, 666])
supply_moderate_low = fuzz.trimf(x_supply, [444, 666, 888])
supply_moderate = fuzz.trimf(x_supply, [666, 888, 1110])
supply_moderate_high = fuzz.trimf(x_supply, [888, 1110, 1332])
supply_high = fuzz.trimf(x_supply, [1110, 1332, 1554])
supply_very_high = fuzz.trimf(x_supply, [1332, 1554, 1554])

demand_very_low = fuzz.trapmf(x_demand, [0, 0, 78, 156])
demand_low = fuzz.trimf(x_demand, [78, 156, 234])
demand_moderate_low = fuzz.trimf(x_demand, [156, 234, 313])
demand_moderate = fuzz.trimf(x_demand, [234, 313, 391])
demand_moderate_high = fuzz.trimf(x_demand, [313, 391, 469])
demand_high = fuzz.trimf(x_demand, [391, 469, 547])
demand_very_high = fuzz.trimf(x_demand, [469, 547, 547])

production_very_low = fuzz.trapmf(x_production, [0, 0, 78, 157])
    ion_low = fuzz.trimf(x_production, [78, 157, 235])
    ion_moderate_low = fuzz.trimf(x_production, [157, 235,
```

```
production_moderate = fuzz.trimf(x_production, [235, 314, 392])
production_moderate_high = fuzz.trimf(x_production, [314, 392,
471])
production_high = fuzz.trimf(x_production, [392, 471, 549])
production_very_high = fuzz.trimf(x_production, [471, 549, 549])


#Visualize These Universes and membership functions


fig, ax0 = plt.subplots(nrows=1, figsize=(8, 4))
ax0.plot(x_supply, supply_very_low, 'b', linewidth=1.5,
label='Very Low')
ax0.plot(x_supply, supply_low, 'g', linewidth=1.5, label='Low')
ax0.plot(x_supply, supply_moderate_low, 'c', linewidth=1.5,
label='Moderate Low')
ax0.plot(x_supply, supply_moderate, 'r', linewidth=1.5,
label='Moderate')
ax0.plot(x_supply, supply_moderate_high, 'm', linewidth=1.5,
label='Moderate High')
ax0.plot(x_supply, supply_high, 'y', linewidth=1.5,
label='High')
ax0.plot(x_supply, supply_very_high, 'orange', linewidth=1.5,
label='Very High')
ax0.set_title('SUPPLY GRAPHIC')
                ,end()


                _xlim([1, 1600])
```

```
ax0.spines['top'].set_visible(False)
ax0.spines['right'].set_visible(False)
ax0.get_xaxis().tick_bottom()
ax0.get_yaxis().tick_left()

plt.xlabel('Supply', fontsize=14)
plt.ylabel('Membership Number', fontsize=14)
plt.tight_layout()
plt.grid(True)
plt.legend(loc=4)
plt.show()

fig, ax1 = plt.subplots(nrows=1, figsize=(8, 4))
ax1.plot(x_demand, demand_very_low, 'b', linewidth=1.5,
label='Very Low')
ax1.plot(x_demand, demand_low, 'g', linewidth=1.5, label='Low')
ax1.plot(x_demand, demand_moderate_low, 'c', linewidth=1.5,
label='Moderate Low')
ax1.plot(x_demand, demand_moderate, 'r', linewidth=1.5,
label='Moderate')
ax1.plot(x_demand, demand_moderate_high, 'm', linewidth=1.5,
label='Moderate High')
ax1.plot(x_demand, demand_high, 'y', linewidth=1.5,
        High')
        t(x_demand, demand_very_high, 'orange', linewidth=1.5,
        Very High')
```

```
ax1.set_title('DEMAND GRAPHIC')
ax1.legend()

ax1.set_xlim([1, 600])

ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)
ax1.get_xaxis().tick_bottom()
ax1.get_yaxis().tick_left()

plt.xlabel('Demand', fontsize=14)
plt.ylabel('Membership Number', fontsize=14)
plt.tight_layout()
plt.grid(True)
plt.legend(loc=4)
plt.show()

fig, ax2 = plt.subplots(nrows=1, figsize=(8, 4))
ax2.plot(x_production, production_very_low, 'b', linewidth=1.5,
label='Very Low')
ax2.plot(x_production, production_low, 'g', linewidth=1.5,
label='Low')
ax2.plot(x_production, production_moderate_low, 'c',
linewidth=1.5, label='Moderate Low')
    t(x_production, production_moderate, 'r', linewidth=1.5,
    Moderate')
```

```
ax2.plot(x_production, production_moderate_high, 'm',
linewidth=1.5, label='Moderate High')
ax2.plot(x_production, production_high, 'y', linewidth=1.5,
label='High')
ax2.plot(x_production, production_very_high, 'orange',
linewidth=1.5, label='Very High')
ax2.set_title('PRODUCTION GRAPHIC')
ax2.legend()

ax2.set_xlim([1, 600])

ax2.spines['top'].set_visible(False)
ax2.spines['right'].set_visible(False)
ax2.get_xaxis().tick_bottom()
ax2.get_yaxis().tick_left()

plt.xlabel('Production', fontsize=14)
plt.ylabel('Membership Number', fontsize=14)
plt.tight_layout()
plt.grid(True)
plt.legend(loc=4)
plt.show()
```

`ikasi input SUPPLY

```
supply_level_very_low = fuzz.interp_membership(x_supply,
supply_very_low, supply)
supply_level_low = fuzz.interp_membership(x_supply, supply_low,
supply)
supply_level_moderate_low = fuzz.interp_membership(x_supply,
supply_moderate_low, supply)
supply_level_moderate = fuzz.interp_membership(x_supply,
supply_moderate, supply)
supply_level_moderate_high = fuzz.interp_membership(x_supply,
supply_moderate_high, supply)
supply_level_high = fuzz.interp_membership(x_supply,
supply_high, supply)
supply_level_very_high = fuzz.interp_membership(x_supply,
supply_very_high, supply)

#Fuzzifikasi input DEMAND

demand_level_very_low = fuzz.interp_membership(x_demand,
demand_very_low, demand)
demand_level_low = fuzz.interp_membership(x_demand, demand_low,
demand)
demand_level_moderate_low = fuzz.interp_membership(x_demand,
demand_moderate_low, demand)
demand_level_moderate = fuzz.interp_membership(x_demand,
    moderate, demand)
    level_moderate_high = fuzz.interp_membership(x_demand,
    moderate_high, demand)
```

```
demand_level_high = fuzz.interp_membership(x_demand,
demand_high, demand)
demand_level_very_high = fuzz.interp_membership(x_demand,
demand_very_high, demand)


print("")
print('Supply Membership Degree')
print(supply_level_very_low)
print(supply_level_low)
print(supply_level_moderate_low)
print(supply_level_moderate)
print(supply_level_moderate_high)
print(supply_level_high)
print(supply_level_very_high)
print("")
print('Demand Membership Degree')
print(demand_level_very_low)
print(demand_level_low)
print(demand_level_moderate_low)
print(demand_level_moderate)
print(demand_level_moderate_high)
print(demand_level_high)
print(demand_level_very_high)
print("")
```


```
rule1 = np.fmin(supply_level_very_low,
level_very_low)
```

```
active_rule2 = np.fmin(supply_level_very_low, demand_level_low)
active_rule3 = np.fmin(supply_level_very_low,
demand_level_moderate_low)
active_rule4 = np.fmin(supply_level_very_low,
demand_level_moderate)
active_rule5 = np.fmin(supply_level_very_low,
demand_level_moderate_high)
active_rule6 = np.fmin(supply_level_very_low, demand_level_high)
active_rule7 = np.fmin(supply_level_very_low,
demand_level_very_high)
active_rule8 = np.fmin(supply_level_low, demand_level_very_low)
active_rule9 = np.fmin(supply_level_low, demand_level_low)
active_rule10 = np.fmin(supply_level_low,
demand_level_moderate_low)
active_rule11 = np.fmin(supply_level_low, demand_level_moderate)
active_rule12 = np.fmin(supply_level_low,
demand_level_moderate_high)
active_rule13 = np.fmin(supply_level_low, demand_level_high)
active_rule14 = np.fmin(supply_level_low,
demand_level_very_high)
active_rule15 = np.fmin(supply_level_moderate_low,
demand_level_very_low)
active_rule16 = np.fmin(supply_level_moderate_low,
demand_level_low)
    rule17 = np.fmin(supply_level_moderate_low,
    level_moderate_low)
```

```
active_rule18 = np.fmin(supply_level_moderate_low,
demand_level_moderate)
active_rule19 = np.fmin(supply_level_moderate_low,
demand_level_moderate_high)
active_rule20 = np.fmin(supply_level_moderate_low,
demand_level_high)
active_rule21 = np.fmin(supply_level_moderate_low,
demand_level_very_high)
active_rule22 = np.fmin(supply_level_moderate,
demand_level_very_low)
active_rule23 = np.fmin(supply_level_moderate, demand_level_low)
active_rule24 = np.fmin(supply_level_moderate,
demand_level_moderate_low)
active_rule25 = np.fmin(supply_level_moderate,
demand_level_moderate)
active_rule26 = np.fmin(supply_level_moderate,
demand_level_moderate_high)
active_rule27 = np.fmin(supply_level_moderate,
demand_level_high)
active_rule28 = np.fmin(supply_level_moderate,
demand_level_very_high)
active_rule29 = np.fmin(supply_level_moderate_high,
demand_level_very_low)
active_rule30 = np.fmin(supply_level_moderate_high,
    level_low)
    rule31 = np.fmin(supply_level_moderate_high,
    level_moderate_low)
```

```
active_rule32 = np.fmin(supply_level_moderate_high,
demand_level_moderate)
active_rule33 = np.fmin(supply_level_moderate_high,
demand_level_moderate_high)
active_rule34 = np.fmin(supply_level_moderate_high,
demand_level_high)
active_rule35 = np.fmin(supply_level_moderate_high,
demand_level_very_high)
active_rule36 = np.fmin(supply_level_high,
demand_level_very_low)
active_rule37 = np.fmin(supply_level_high, demand_level_low)
active_rule38 = np.fmin(supply_level_high,
demand_level_moderate_low)
active_rule39 = np.fmin(supply_level_high,
demand_level_moderate)
active_rule40 = np.fmin(supply_level_high,
demand_level_moderate_high)
active_rule41 = np.fmin(supply_level_high, demand_level_high)
active_rule42 = np.fmin(supply_level_high,
demand_level_very_high)
active_rule43 = np.fmin(supply_level_very_high,
demand_level_very_low)
active_rule44 = np.fmin(supply_level_very_high,
demand_level_low)
    rule45 = np.fmin(supply_level_very_high,
    level_moderate_low)
```

```
active_rule46 = np.fmin(supply_level_very_high,
demand_level_moderate)
active_rule47 = np.fmin(supply_level_very_high,
demand_level_moderate_high)
active_rule48 = np.fmin(supply_level_very_high,
demand_level_high)
active_rule49 = np.fmin(supply_level_very_high,
demand_level_very_high)



production_activation_1 = np.fmin(active_rule1,
production_very_low)
production_activation_2 = np.fmin(active_rule2, production_low)
production_activation_3 = np.fmin(active_rule3,
production_moderate_low)
production_activation_4 = np.fmin(active_rule4,
production_moderate)
production_activation_5 = np.fmin(active_rule5,
production_moderate_high)
production_activation_6 = np.fmin(active_rule6, production_high)
production_activation_7 = np.fmin(active_rule7,
production_very_high)
production_activation_8 = np.fmin(active_rule8,
production_very_low)
    ion_activation_9 = np.fmin(active_rule9, production_low)
    ion_activation_10 = np.fmin(active_rule10,
    ion_moderate_low)
```

```
production_activation_11 = np.fmin(active_rule11,
production_moderate)
production_activation_12 = np.fmin(active_rule12,
production_moderate_high)
production_activation_13 = np.fmin(active_rule13,
production_high)
production_activation_14 = np.fmin(active_rule14,
production_very_high)
production_activation_15 = np.fmin(active_rule15,
production_very_low)
production_activation_16 = np.fmin(active_rule16,
production_low)
production_activation_17 = np.fmin(active_rule17,
production_moderate_low)
production_activation_18 = np.fmin(active_rule18,
production_moderate)
production_activation_19 = np.fmin(active_rule19,
production_moderate_high)
production_activation_20 = np.fmin(active_rule20,
production_high)
production_activation_21 = np.fmin(active_rule21,
production_very_high)
production_activation_22 = np.fmin(active_rule22,
production_very_low)
    ion_activation_23 = np.fmin(active_rule23,
    ion_low)
```

```
production_activation_24 = np.fmin(active_rule24,
production_moderate_low)
production_activation_25 = np.fmin(active_rule25,
production_moderate)
production_activation_26 = np.fmin(active_rule26,
production_moderate_high)
production_activation_27 = np.fmin(active_rule27,
production_high)
production_activation_28 = np.fmin(active_rule28,
production_very_high)
production_activation_29 = np.fmin(active_rule29,
production_very_low)
production_activation_30 = np.fmin(active_rule30,
production_low)
production_activation_31 = np.fmin(active_rule31,
production_moderate_low)
production_activation_32 = np.fmin(active_rule32,
production_moderate)
production_activation_33 = np.fmin(active_rule33,
production_moderate_high)
production_activation_34 = np.fmin(active_rule34,
production_high)
production_activation_35 = np.fmin(active_rule35,
production_very_high)
    ion_activation_36 = np.fmin(active_rule36,
    ion_very_low)
```

```
production_activation_37 = np.fmin(active_rule37,
production_low)
production_activation_38 = np.fmin(active_rule38,
production_moderate_low)
production_activation_39 = np.fmin(active_rule39,
production_moderate)
production_activation_40 = np.fmin(active_rule40,
production_moderate_high)
production_activation_41 = np.fmin(active_rule41,
production_high)
production_activation_42 = np.fmin(active_rule42,
production_very_high)
production_activation_43 = np.fmin(active_rule43,
production_very_low)
production_activation_44 = np.fmin(active_rule44,
production_low)
production_activation_45 = np.fmin(active_rule45,
production_moderate_low)
production_activation_46 = np.fmin(active_rule46,
production_moderate)
production_activation_47 = np.fmin(active_rule47,
production_moderate_high)
production_activation_48 = np.fmin(active_rule48,
production_high)
    ion_activation_49 = np.fmin(active_rule49,
    ion_very_high)
    ion0 = np.zeros_like(x_production)
```

```
#Accumulation Graph
fig, ax3 = plt.subplots(figsize=(8, 4))
ax3.plot (x_production, production_very_low,'b', linewidth =0.5)
ax3.plot (x_production, production_low,'g', linewidth =0.5)
ax3.plot (x_production, production_moderate_low,'c', linewidth
=0.5)
ax3.plot (x_production, production_moderate,'r', linewidth =0.5)
ax3.plot (x_production, production_moderate_high,'m', linewidth
=0.5)
ax3.plot (x_production, production_high,'y', linewidth =0.5)
ax3.plot (x_production, production_very_high,'orange', linewidth
=0.5)

ax3.fill_between(x_production, production0,
production_activation_1, facecolor='b', alpha=0.7)
ax3.plot(x_production, production_activation_1, 'b',
linewidth=0.5, linestyle='--', label= 'Production_Activation_1')

ax3.fill_between(x_production, production0,
production_activation_2, facecolor='g', alpha=0.7)
ax3.plot(x_production, production_activation_2, 'g',
linewidth=0.5, linestyle='--', label= 'Production_Activation_2')

l_between(x_production, production0,
ion_activation_3, facecolor='c', alpha=0.7)
```

```
ax3.plot(x_production, production_activation_3, 'c',
linewidth=0.5, linestyle='--', label= 'Production_Activation_3')

ax3.fill_between(x_production, production0,
production_activation_4, facecolor='r', alpha=0.7)
ax3.plot(x_production, production_activation_4, 'r',
linewidth=0.5, linestyle='--', label= 'Production_Activation_4')

ax3.fill_between(x_production, production0,
production_activation_5, facecolor='m', alpha=0.7)
ax3.plot(x_production, production_activation_5, 'm',
linewidth=0.5, linestyle='--', label= 'Production_Activation_5')

ax3.fill_between(x_production, production0,
production_activation_6, facecolor='y', alpha=0.7)
ax3.plot(x_production, production_activation_6, 'y',
linewidth=0.5, linestyle='--', label= 'Production_Activation_6')

ax3.fill_between(x_production, production0,
production_activation_7, facecolor='orange', alpha=0.7)
ax3.plot(x_production, production_activation_7, 'orange',
linewidth=0.5, linestyle='--', label= 'Production_Activation_7')

ax3.fill_between(x_production, production0,
ion_activation_8, facecolor='darkblue', alpha=0.7)
t(x_production, production_activation_8, 'darkblue',
th=0.5, linestyle='--', label= 'Production_Activation_8')
```

```
ax3.fill_between(x_production, production0,
production_activation_9, facecolor='darkgreen', alpha=0.7)
ax3.plot(x_production, production_activation_9, 'darkgreen',
linewidth=0.5, linestyle='--', label= 'Production_Activation_9')

ax3.fill_between(x_production, production0,
production_activation_10, facecolor='darkcyan', alpha=0.7)
ax3.plot(x_production, production_activation_10, 'darkcyan',
linewidth=0.5, linestyle='--', label=
'Production_Activation_10')

ax3.fill_between(x_production, production0,
production_activation_11, facecolor='darkred', alpha=0.7)
ax3.plot(x_production, production_activation_11, 'darkred',
linewidth=0.5, linestyle='--', label=
'Production_Activation_11')

ax3.fill_between(x_production, production0,
production_activation_12, facecolor='darkmagenta', alpha=0.7)
ax3.plot(x_production, production_activation_12, 'darkmagenta',
linewidth=0.5, linestyle='--', label=
'Production_Activation_12')

l_between(x_production, production0,
ion_activation_13, facecolor='gold', alpha=0.7)
```

```
ax3.plot(x_production, production_activation_13, 'gold',
linewidth=0.5, linestyle='--', label=
'Production_Activation_13')

ax3.fill_between(x_production, production0,
production_activation_14, facecolor='darkorange', alpha=0.7)
ax3.plot(x_production, production_activation_14, 'darkorange',
linewidth=0.5, linestyle='--', label=
'Production_Activation_14')

ax3.fill_between(x_production, production0,
production_activation_15, facecolor='mediumblue', alpha=0.7)
ax3.plot(x_production, production_activation_15, 'mediumblue',
linewidth=0.5, linestyle='--', label=
'Production_Activation_15')

ax3.fill_between(x_production, production0,
production_activation_16, facecolor='lime', alpha=0.7)
ax3.plot(x_production, production_activation_16, 'lime',
linewidth=0.5, linestyle='--', label=
'Production_Activation_16')

ax3.fill_between(x_production, production0,
production_activation_17, facecolor='aqua', alpha=0.7)
    t(x_production, production_activation_17, 'aqua',
    th=0.5, linestyle='--', label=
    tion_Activation_17')
```

```
ax3.fill_between(x_production, production0,
production_activation_18, facecolor='tomato', alpha=0.7)
ax3.plot(x_production, production_activation_18, 'tomato',
linewidth=0.5, linestyle='--', label=
'Production_Activation_18')

ax3.fill_between(x_production, production0,
production_activation_19, facecolor='fuchsia', alpha=0.7)
ax3.plot(x_production, production_activation_19, 'fuchsia',
linewidth=0.5, linestyle='--', label=
'Production_Activation_19')

ax3.fill_between(x_production, production0,
production_activation_20, facecolor='olive', alpha=0.7)
ax3.plot(x_production, production_activation_20, 'olive',
linewidth=0.5, linestyle='--', label=
'Production_Activation_20')

ax3.fill_between(x_production, production0,
production_activation_21, facecolor='orangered', alpha=0.7)
ax3.plot(x_production, production_activation_21, 'orangered',
linewidth=0.5, linestyle='--', label=
'Production_Activation_21')

l_between(x_production, production0,
ion_activation_22, facecolor='slateblue', alpha=0.7)
```

```
ax3.plot(x_production, production_activation_22, 'slateblue',
linewidth=0.5, linestyle='--', label=
'Production_Activation_22')

ax3.fill_between(x_production, production0,
production_activation_23, facecolor='seagreen', alpha=0.7)
ax3.plot(x_production, production_activation_23, 'seagreen',
linewidth=0.5, linestyle='--', label=
'Production_Activation_23')

ax3.fill_between(x_production, production0,
production_activation_24, facecolor='lightseagreen', alpha=0.7)
ax3.plot(x_production, production_activation_24,
'lightseagreen', linewidth=0.5, linestyle='--', label=
'Production_Activation_24')

ax3.fill_between(x_production, production0,
production_activation_25, facecolor='salmon', alpha=0.7)
ax3.plot(x_production, production_activation_25, 'salmon',
linewidth=0.5, linestyle='--', label=
'Production_Activation_25')

ax3.fill_between(x_production, production0,
production_activation_26, facecolor='orchid', alpha=0.7)
    t(x_production, production_activation_26, 'orchid',
    th=0.5, linestyle='--', label=
    tion_Activation_26')
```

```
ax3.fill_between(x_production, production0,
production_activation_27, facecolor='olivedrab', alpha=0.7)
ax3.plot(x_production, production_activation_27, 'olivedrab',
linewidth=0.5, linestyle='--', label=
'Production_Activation_27')

ax3.fill_between(x_production, production0,
production_activation_28, facecolor='coral', alpha=0.7)
ax3.plot(x_production, production_activation_28, 'coral',
linewidth=0.5, linestyle='--', label=
'Production_Activation_28')

ax3.fill_between(x_production, production0,
production_activation_29, facecolor='mediumslateblue',
alpha=0.7)
ax3.plot(x_production, production_activation_29,
'mediumslateblue', linewidth=0.5, linestyle='--', label=
'Production_Activation_29')

ax3.fill_between(x_production, production0,
production_activation_30, facecolor='mediumseagreen', alpha=0.7)
ax3.plot(x_production, production_activation_30,
'mediumseagreen', linewidth=0.5, linestyle='--', label=
    tion_Activation_30')
```

```
ax3.fill_between(x_production, production0,
production_activation_31, facecolor='turquoise', alpha=0.7)
ax3.plot(x_production, production_activation_31, 'turquoise',
linewidth=0.5, linestyle='--', label=
'Production_Activation_31')

ax3.fill_between(x_production, production0,
production_activation_32, facecolor='lightsalmon', alpha=0.7)
ax3.plot(x_production, production_activation_32, 'lightsalmon',
linewidth=0.5, linestyle='--', label=
'Production_Activation_32')

ax3.fill_between(x_production, production0,
production_activation_33, facecolor='violet', alpha=0.7)
ax3.plot(x_production, production_activation_33, 'violet',
linewidth=0.5, linestyle='--', label=
'Production_Activation_33')

ax3.fill_between(x_production, production0,
production_activation_34, facecolor='yellowgreen', alpha=0.7)
ax3.plot(x_production, production_activation_34, 'yellowgreen',
linewidth=0.5, linestyle='--', label=
'Production_Activation_34')

l_between(x_production, production0,
ion_activation_35, facecolor='firebrick', alpha=0.7)
```

```
ax3.plot(x_production, production_activation_35, 'firebrick',
linewidth=0.5, linestyle='--', label=
'Production_Activation_35')

ax3.fill_between(x_production, production0,
production_activation_36, facecolor='purple', alpha=0.7)
ax3.plot(x_production, production_activation_36, 'purple',
linewidth=0.5, linestyle='--', label=
'Production_Activation_36')

ax3.fill_between(x_production, production0,
production_activation_37, facecolor='springgreen', alpha=0.7)
ax3.plot(x_production, production_activation_37, 'springgreen',
linewidth=0.5, linestyle='--', label=
'Production_Activation_37')

ax3.fill_between(x_production, production0,
production_activation_38, facecolor='mediumturquoise',
alpha=0.7)
ax3.plot(x_production, production_activation_38,
'mediumturquoise', linewidth=0.5, linestyle='--', label=
'Production_Activation_38')

ax3.fill_between(x_production, production0,
        ion_activation_39, facecolor='darksalmon', alpha=0.7)
```

```
ax3.plot(x_production, production_activation_39, 'darksalmon',
linewidth=0.5, linestyle='--', label=
'Production_Activation_39')

ax3.fill_between(x_production, production0,
production_activation_40, facecolor='mediumvioletred',
alpha=0.7)
ax3.plot(x_production, production_activation_40,
'mediumvioletred', linewidth=0.5, linestyle='--', label=
'Production_Activation_40')

ax3.fill_between(x_production, production0,
production_activation_41, facecolor='greenyellow', alpha=0.7)
ax3.plot(x_production, production_activation_41, 'greenyellow',
linewidth=0.5, linestyle='--', label=
'Production_Activation_41')

ax3.fill_between(x_production, production0,
production_activation_42, facecolor='brown', alpha=0.7)
ax3.plot(x_production, production_activation_42, 'brown',
linewidth=0.5, linestyle='--', label=
'Production_Activation_42')

ax3.fill_between(x_production, production0,
        ion_activation_43, facecolor='mediumpurple', alpha=0.7)
```

```
ax3.plot(x_production, production_activation_43, 'mediumpurple',
linewidth=0.5, linestyle='--', label=
'Production_Activation_43')

ax3.fill_between(x_production, production0,
production_activation_44, facecolor='mediumspringgreen',
alpha=0.7)
ax3.plot(x_production, production_activation_44,
'mediumspringgreen', linewidth=0.5, linestyle='--', label=
'Production_Activation_44')

ax3.fill_between(x_production, production0,
production_activation_45, facecolor='paleturquoise', alpha=0.7)
ax3.plot(x_production, production_activation_45,
'paleturquoise', linewidth=0.5, linestyle='--', label=
'Production_Activation_45')

ax3.fill_between(x_production, production0,
production_activation_46, facecolor='mistyrose', alpha=0.7)
ax3.plot(x_production, production_activation_46, 'mistyrose',
linewidth=0.5, linestyle='--', label=
'Production_Activation_46')

ax3.fill_between(x_production, production0,
        ion_activation_47, facecolor='palevioletred', alpha=0.7)
```

```
ax3.plot(x_production, production_activation_47,
'palevioletred', linewidth=0.5, linestyle='--', label=
'Production_Activation_47' )

ax3.fill_between(x_production, production0,
production_activation_48, facecolor='palegreen', alpha=0.7)
ax3.plot(x_production, production_activation_48, 'palegreen',
linewidth=0.5, linestyle='--', label=
'Production_Activation_48')

ax3.fill_between(x_production, production0,
production_activation_49, facecolor='chocolate', alpha=0.7)
ax3.plot(x_production, production_activation_49, 'chocolate',
linewidth=0.5, linestyle='--', label=
'Production_Activation_49')

ax3.set_title('Output Membership Activity')

for ax in (ax0,):
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.get_xaxis().tick_bottom()
    ax.get_yaxis().tick_left()
    ax.legend()

    end(loc=4)

    ht_layout()
```

```
plt.show()

# Aggregate all three output membership functions together
agregated = np.fmax(production_activation_1,
                    np.fmax(production_activation_2,
                            np.fmax(production_activation_3,

np.fmax(production_activation_4,

np.fmax(production_activation_5,

np.fmax(production_activation_6,

np.fmax(production_activation_7,

np.fmax(production_activation_8,

np.fmax(production_activation_9,

np.fmax(production_activation_10,

np.fmax(production_activation_11,

np.fmax(production_activation_12,

:(production_activation_13,
```

```
np.fmax(production_activation_14,

np.fmax(production_activation_15,

np.fmax(production_activation_16,

np.fmax(production_activation_17,

np.fmax(production_activation_18,

np.fmax(production_activation_19,

np.fmax(production_activation_20,

np.fmax(production_activation_21,

np.fmax(production_activation_22,

np.fmax(production_activation_23,

np.fmax(production_activation_24,

np.fmax(production_activation_25,

:(production_activation_26,
```

```
np.fmax(production_activation_27,

np.fmax(production_activation_28,

np.fmax(production_activation_29,

np.fmax(production_activation_30,

np.fmax(production_activation_31,

np.fmax(production_activation_32,

np.fmax(production_activation_33,

np.fmax(production_activation_34,

np.fmax(production_activation_35,

np.fmax(production_activation_36,

np.fmax(production_activation_37,

np.fmax(production_activation_38,

:(production_activation_39,
```

```
np.fmax(production_activation_40,

np.fmax(production_activation_41,

np.fmax(production_activation_42,

np.fmax(production_activation_43,

np.fmax(production_activation_44,

np.fmax(production_activation_45,

np.fmax(production_activation_46,

np.fmax(production_activation_47,

np.fmax(production_activation_48,
production_activation_49))))))))))))))))))))))))))))))))))))))))))))
))))))))

# Calculate defuzzified result
production = fuzz.defuzz(x_production, agregated, 'centroid')
ion_activation = fuzz.interp_membership(x_production,
ed, production)
```

```
# Visualize this
fig, ax4 = plt.subplots(figsize=(8, 3))
ax4.plot(x_production, production_very_low, 'b', linewidth=0.5,
linestyle='--', )
ax4.plot(x_production, production_low, 'g', linewidth=0.5,
linestyle='--', )
ax4.plot(x_production, production_moderate_low, 'c',
linewidth=0.5, linestyle='--')
ax4.plot(x_production, production_moderate, 'r', linewidth=0.5,
linestyle='--')
ax4.plot(x_production, production_moderate_high, 'm',
linewidth=0.5, linestyle='--')
ax4.plot(x_production, production_high, 'y', linewidth=0.5,
linestyle='--')
ax4.plot(x_production, production_very_high, 'orange',
linewidth=0.5, linestyle='--')
ax4.fill_between(x_production, production0, agregated,
facecolor='darkorange', alpha=0.7)
ax4.plot([production, production], [0, production_activation],
'olive', linewidth=1.5, alpha=0.9)
ax4.set_title('Aggregated membership and result (line)')

# Turn off top/right axes
for ax in (ax0,):
    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()

    print("Defuzzification")

    print("Production: "+str(production))
```

Lampiran B 10

Kartu Konsultasi Tugas Akhir

JUDUL: Application of the fuzzy mamdani method to predict the amount of marble production at PT Utama Tri Makmur, South Sulawesi Province.

(Konsultasi minimal 8 kali)

TANGGAL	MATERI KONSULTASI	PARAF DOSEN
17/09/2021	• Koreksi Bab I • Koreksi Bab II	
23/05/2022	• Koreksi Bab III • MAPE	
14/06/2022	• Update Proses Coding Fuzzy mamdani di Python	
23/06/2022	• Lampiran Coding	
20/10/2022	• Penyalahgunaan tanda baca tidak / kurang tepat (terpisah dari kata yang mendahuluinya).	
10/05/2023	• Penjelasan Alur Penulisan Bab IV	

TANGGAL	MATERI KONSULTASI	PARAF DOSEN
27/12/2023	• Motivasi & Janji temu Konsultasi selanjutnya.	
5/01/2024	• Koreksi Bab IV • Penyesuaian penjelasan perhitungan secara manual dan perhitungan menggunakan bahasa program python	
17/04/2024	• Laporan skripsi selesai • Koreksi Bab III • Koreksi Bab IV	
30/04/2024	• Laporan skripsi selesai • Artikel selesai • Poster selesai	
26/05/2024	• Koreksi Bab I • Koreksi Flow Chart • Koreksi Bab V • Koreksi Poster • Koreksi Artikel.	

TANGGAL	MATERI KONSULTASI	PARAF DOSEN
29/05/2024	• Koreksi Rumusan Masalah • Koreksi Tujuan Penelitian • Koreksi Kesimpulan.	
30/05/2024	• ACC	
24/06/2024	• Koreksi Halaman Pengesahan	
25/06/2024	• Koreksi sampul • Koreksi Halaman Pengesahan • Koreksi Pernyataan Keaslian • ACC	

an: Lembar konsultasi asli dilampirkan pada satu dokumen skripsi.