

IMPLEMENTASI RANKING MODEL BERBASIS FUZZY INFERENCE SYSTEM

Azriel Naufal Aulia¹, Muhammad Faiz Abdurrahman Djauhar², Ryan Abdurohman³

¹1301190374, IF-43-10, S-1 Informatika, Telkom University

²1301190361, IF-43-10, S-1 Informatika, Telkom University

³1301191171, IF-43-10, S-1 Informatika, Telkom University

1. PENDAHULUAN

Dalam kecerdasan buatan, untuk merepresentasikan suatu masalah menjadi basis pengetahuan (*knowledge base*) maka dibutuhkan teknik *reasoning* (penalaran). terdapat 3 teknik *reasoning* (penalaran) yang digunakan, antara lain, yaitu : *propositional logic* (logika proposisi), *first order logic* atau *predicate calculus* (kalkulus predikat) dan *fuzzy logic* (logika samar). Untuk masing-masing teknik tersebut, masing-masing teknik memiliki parameter masalah yang cocok dari satu sama lain. Untuk *propositional logic* (logika proposisi) merupakan logic yang paling sederhana untuk digunakan dan dipahami dalam merepresentasikan suatu masalah. Dengan *propositional logic*, terdapat masalah dalam pemakaiannya. Masalahnya ketika kita dihadapkan dengan masalah yang memiliki kompleksitas yang lebih tinggi dan tidak cukup efektif apabila hanya menggunakan *propositional logic*. Oleh karena itu, dibuat teknik yang lebih efisien yaitu *first order logic* atau *predicate calculus* (kalkulus predikat). Untuk teknik penalaran terakhir yaitu *fuzzy logic*, dipakai ketika dihadapkan dengan masalah yang memiliki sifat ketidakpastian (*uncertainty*). Sistem yang menggunakan teknik *fuzzy logic* dinamakan *fuzzy system*. Data masukan pada *fuzzy system* ini dinamakan *crisp set*. Dengan *crisp set*, *crisp set* nantinya diubah menjadi derajat keanggotaan (berfungsi untuk menghasilkan keputusan yang akurat pada proses selanjutnya) dengan fungsi keanggotaan. Dua nilai tersebut, *crisp set* dan derajat keanggotaan dinamakan sebagai *fuzzy set*, yang merupakan data masukan untuk proses pertama pada *fuzzy system*, yaitu *fuzzification*. Hasil yang dikeluarkan akan dinamakan *fuzzy input*, yang berisi suatu interval numerik dan mempunyai nilai-nilai linguistik, yang dinamakan variabel linguistik. Contoh dari variabel linguistik antara lain, seperti ketika masukan data (*crisp set*) berupa besar suhu ruangan, maka variabel linguistik bisa didefinisikan pada interval $[-10^{\circ}\text{C}, 40^{\circ}\text{C}]$ dan memiliki nilai-nilai linguistik seperti ‘Dingin’, ‘Hangat’, dan ‘Panas’. Setelah itu, terdapat proses yang disebut *inference*. Proses *inference* melakukan penalaran menggunakan *fuzzy input* dan *fuzzy rules* yang telah ditentukan sehingga menghasilkan *fuzzy output*. Pada *fuzzy system*, masalah yang aplikasikan bisa jadi memiliki lebih dari 1 data masukan (*crisp set*), yang nanti akan menghasilkan lebih banyak variabel linguistik juga. Sehingga untuk mengambil keputusan yang adil dan efisien, akan dibuat suatu *fuzzy rules*, untuk mempertimbangkan semua variabel linguistik tersebut dan menghasilkan keputusan yang paling akurat. Pada proses terakhir dinamakan *defuzzification* yang berguna untuk mengubah *fuzzy output* menjadi *crisp value* dengan menggunakan fungsi keanggotaan yang sudah ditentukan.

2. METODOLOGI

Secara umum, kami adaptasi dari jurnal (Amindoust, Ahmed, Saghafinia & Bahreininejad, 2012) dalam menentukan paramater-parameter yang ada, dengan harapan parameter tersebut mampu memberikan hasil optimal.

2.1. Nama Linguistik

No.	Nama Linguistik
1.	Very weakly preferred (VWP)
2.	Weakly preferred (WP)
3.	Low moderately preferred (LMP)
4.	High moderately preferred (HMP)
5.	Strongly preferred (SP)
6.	Very strongly preferred (VSP)
7.	Extremely preferred (EP)

2.2. Bentuk dan Batas Fungsi Keanggotaan Input

Fungsi keanggotaan yang digunakan adalah fungsi trapesium dengan batas-batas a, b, c, dan d. Namun, ketika $b = c$ maka bentuk fungsinya menjadi segitiga.

$$f(x, a, b, c, d) = \begin{cases} \frac{x - a}{b - a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{x - d}{c - d}, & c \leq x \leq d \end{cases}$$

The image displays two screenshots of a Visual Studio Code editor window. The top screenshot shows a Python function named `trapezoid` that takes parameters `id`, `x`, and `jenis`. It initializes a dictionary `basas` and a fuzzy set `fuzz`. Depending on the `jenis` value ("pelayanan" or "makanan"), it processes a list of values `v` and calculates a membership value `temp` based on the position of `x` within the range of `v`. The bottom screenshot shows the initialization of `crisplist`, `fuzzyInput`, and two dictionaries `l1` and `l2` containing numerical ranges for different linguistic terms. A loop is shown starting at line 33.

```

43 def trapezoid(id, x, jenis):
44     batas = {}
45     fuzz = {}
46     "id" : id,
47     "crisp" : x,
48     "fuz" : {}
49 }
50 if jenis == "pelayanan":
51     for k,v in l1.items():
52         if v[0]<=x<=v[-1]:
53             batas[k] = v
54 elif jenis == "makanan":
55     for k,v in l2.items():
56         if v[0]<=x<=v[-1]:
57             batas[k] = v
58 for k, v in batas.items():
59     a, b, c, d = v[0], v[1], v[2], v[3]
60     if a<=x<=b:
61         temp = (x-a)/(b-a)
62     elif b<=x<=c:
63         temp = 1
64     elif c<=x<=d:
65         temp = (x-d)/(c-d)
66     if temp > 0:
67         fuzz["fuz"][k] = temp
68     return fuzz

```

```

8
9 crisplist = []
10 fuzzyInput = {
11     "pelayanan" : [],
12     "makanan" : []
13 }
14 l1 = {
15     "vwp" : [0, 0, 10, 20],
16     "wp" : [10, 25, 25, 40],
17     "lmp" : [30, 40, 40, 50],
18     "hmp" : [40, 55, 55, 70],
19     "sp" : [60, 70, 70, 80],
20     "vsp" : [70, 80, 80, 90],
21     "ep" : [80, 90, 100, 100]
22 }
23 l2 = {
24     "vwp" : [0, 0, 1, 2],
25     "wp" : [1, 2.5, 2.5, 4],
26     "lmp" : [3, 4, 4, 5],
27     "hmp" : [4, 5.5, 5.5, 7],
28     "sp" : [6, 7, 7, 8],
29     "vsp" : [7, 8, 8, 9],
30     "ep" : [8, 9, 10, 10]
31 }
32
33 for i in range(2, 102):

```

2.2.1 Field Pelayanan

No.	Nama Linguistik	Batas (a, b, c, d)
1.	Very weakly preferred (VWP)	(0, 0, 10, 20)
2.	Weakly preferred (WP)	(10, 25, 25, 40)
3.	Low moderately preferred (LMP)	(30, 40, 40, 50)
4.	High moderately preferred (HMP)	(40, 55, 55, 70)
5.	Strongly preferred (SP)	(60, 70, 70, 80)

6.	Very strongly preferred (VSP)	(70, 80, 80, 90)
7.	Extremely preferred (EP)	(80, 90, 100, 100)

2.2.1 Field Makanan

No.	Nama Linguistik	Batas (a, b, c, d)
1.	Very weakly preferred (VWP)	(0, 0, 1, 2)
2.	Weakly preferred (WP)	(1, 2.5, 2.5, 4)
3.	Low moderately preferred (LMP)	(3, 4, 4, 5)
4.	High moderately preferred (HMP)	(4, 5.5, 5.5, 7)
5.	Strongly preferred (SP)	(6, 7, 7, 8)
6.	Very strongly preferred (VSP)	(7, 8, 8, 9)
7.	Extremely preferred (EP)	(8, 9, 10, 10)

2.3. Aturan Inferensi

P\K	VWP	WP	LMP	HMP	SP	VSP	EP
VWP	VWP	VWP	WP	WP	LMP	HMP	HMP
WP	VWP	VWP	WP	LMP	LMP	HMP	SP
LMP	WP	WP	LMP	LMP	HMP	HMP	SP
HMP	WP	LMP	HMP	HMP	SP	SP	VSP
SP	LMP	LMP	HMP	HMP	SP	SP	VSP
VSP	HMP	HMP	HMP	SP	SP	VSP	VSP
EP	HMP	SP	SP	SP	VSP	VSP	EP

```

76
77 def MamdaniInference(id):
78     fuzzyOutput = {
79         "fuzOut" : {}
80     }
81     for k,v in fuzzyInput["pelayanan"][id]["fuz"].items():
82         for k1,v1 in fuzzyInput["makanan"][id]["fuz"].items():
83             sem = fuzzy_rules(k, k1)
84             derajat_rendah = min(v,v1)
85             fuzzyOutput["id"] = id + 1
86             if len(fuzzyOutput["fuzOut"]) == 0:
87                 fuzzyOutput["fuzOut"][sem] = derajat_rendah
88             else:
89                 if sem in fuzzyOutput["fuzOut"]:
90                     fuzzyOutput["fuzOut"][sem] = max(derajat_rendah, fuzzyOutput["fuzOut"][sem])
91                 else:
92                     fuzzyOutput["fuzOut"][sem] = derajat_rendah
93     return fuzzyOutput
94
95 def fuzzy_rules(a,b):
96     if a == 'vwp' and b == 'vwp':
97         return 'vwp'
98     elif a == 'vwp' and b == 'wp':
99         return 'vwp'
100     elif a == 'vwp' and b == 'lmp':
101         return 'wp'

```

2.4. Metode Defuzzifikasi

Metode defuzzifikasi yang digunakan adalah metode Center of Gravity (Mamdani Model). Fungsi keanggotaan yang kami gunakan adalah trapesium dan segitiga (diskrit) sehingga kami menggunakan versi fungsi sumasi dari metode ini, yaitu:

$$y^* = \frac{\sum y \mu_R(y)}{\sum \mu_R(y)}$$

Di mana y^* adalah nilai *crisp* dan $\mu_R(y)$ adalah derajat keanggotaan dari y .

2.5. Bentuk dan Batas Fungsi Keanggotaan Output (sesuai metode defuzzifikasi)

Input dari metode ini adalah fuzzy output hasil inferensi dengan nama dan batas berikut,

No.	Nama Linguistik Fuzzy Ouput	Batas (a, b, c, d)
1.	Very weakly preferred (VWP)	(0, 0, 10, 20)
2.	Weakly preferred (WP)	(10, 25, 25, 40)
3.	Low moderately preferred (LMP)	(30, 40, 40, 50)
4.	High moderately preferred (HMP)	(40, 55, 55, 70)
5.	Strongly preferred (SP)	(60, 70, 70, 80)
6.	Very strongly preferred (VSP)	(70, 80, 80, 90)
7.	Extremely preferred (EP)	(80, 90, 100, 100)

Perhatikan bahwa nilai fuzzy output dari suatu data bisa lebih dari satu jenis. Berdasarkan metode defuzzifikasi yang kami gunakan, diambil sebanyak n-angka untuk menentukan titik

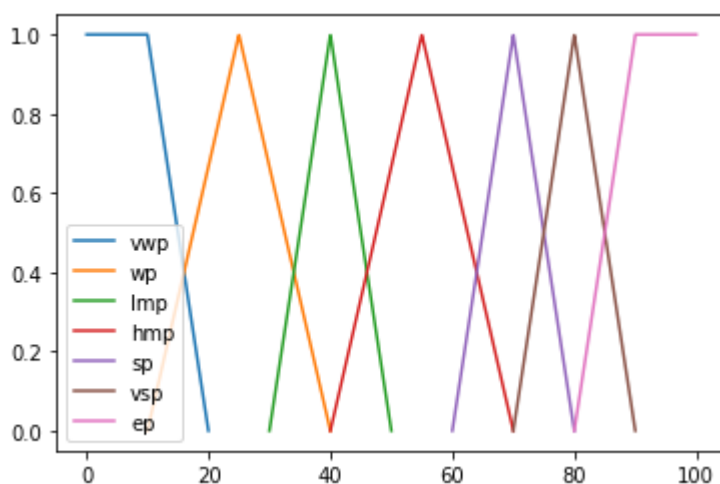
sembarang pada area agregasi hasil Clipping dari semua aturan fuzzy yang didapat. Lalu, setelah itu dimasukkan ke rumus sesuai dengan teori pada 2.4.

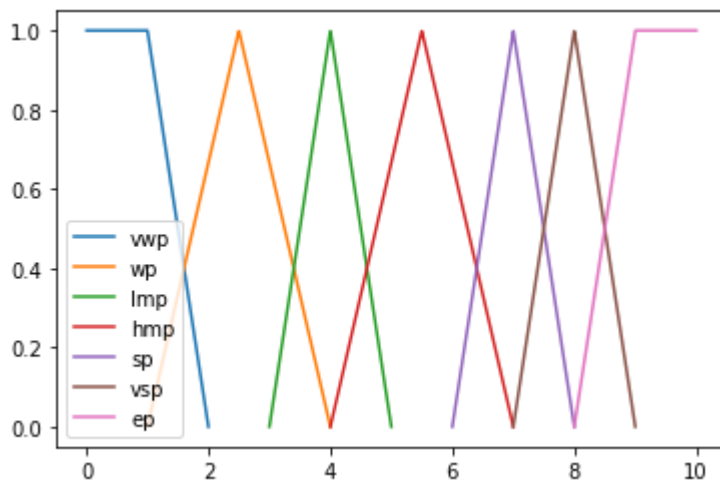
```
File Edit Selection View Go Run Terminal Help fuzzy.py - Visual Studio Code
fuzzy.py 2 X
C:\Users\Ryan Abdurrohmam\Desktop> fuzzy > fuzzy.py > ...
204
205 def defuzzification(x):
206     defuz = {}
207     for j in range (0, 100):
208         model = 0
209         luas = 0
210         #####
211         try :
212             if x[j]['fuzOut']['vwp']>0 :
213                 a = (5*x[j]['fuzOut']['vwp'])+(10*x[j]['fuzOut']['vwp'])+(15*x[j]['fuzOut']['vwp'])
214                 b = x[j]['fuzOut']['vwp']*3
215                 model = model + a
216                 luas = luas + b
217         except:
218             pass
219         #####
220         try :
221             if x[j]['fuzOut']['wp']>0 :
222                 a = (15*x[j]['fuzOut']['wp'])+(20*x[j]['fuzOut']['wp'])+(25*x[j]['fuzOut']['wp'])+(30*x[j]['fuzOut']['wp'])+(3
223                 b = x[j]['fuzOut']['wp']*6
224                 model = model + a
225                 luas = luas + b
226         except:
227             pass
228         #####
229         try :
```

3. HASIL DAN PEMBAHASAN

3.1 Proses *fuzzification*

Pada proses ini kita akan mengubah crisp input menjadi fuzzy input. Pada bagian 2, kita sudah mendefinisikan fungsi keanggotaan dari model kita. Secara grafis, fungsi keanggotaannya bisa digambar menjadi dua gambar berikut (untuk field pelayanan dan makanan)





Misal, pada id 1, kita akan mengubah crisp inputnya menjadi fuzzy input sebagai berikut,

- Nilai crisp pelayanan adalah 58. Nilai ini berada pada nilai linguistik High moderately preferred (HMP), sehingga variabel linguistiknya atau batas-batasnya adalah [40, 55, 55, 70] berturut-turut direpresentasi dengan a, b, c, dan d. Lalu kita masukkan ke fungsi $\frac{x-d}{c-d}$ karena $c \leq x \leq d$ maka $\frac{58-70}{55-70} = 0.8$
- Crisp makanan untuk id 1 adalah 7. Merupakan anggota dari Strongly preferred (SP) dengan batas [6, 7, 7, 8], sehingga nilai fuzzy inputnya adalah 1, karena berada pada rentang $b \leq x \leq c$.

3.2 Proses Inference

Prosen inferensi yang kami lakukan menggunakan model Mamdani, lebih tepatnya metode clipping. Berdasarkan metode inferensi yang telah disebutkan pada bagian metodologi, semisal kita memilih id 1, maka proses inferensinya sebagai berikut,

- Menggunakan aturan Conjunction dengan memilih derajat keanggotaan minimum dari nilai-nilai linguistik yang dihubungkan oleh operator AND dan melakukan Clipping pada fungsi keanggotaan trapesium untuk Nilai Kelayakan (Kualitas Restoran). Sehingga diperoleh:
IF Pelayanan is High moderately preferred (0.8) **AND** Makanan is Strongly preferred (1) **THEN** Nilai Kelayakan (Kualitas) is Strongly preferred (0.8)
- Menggunakan aturan Disjunction untuk memilih derajat keanggotaan maksimum. Namun, karena id 1 hanya memiliki satu derajat keanggotaan, maka fuzzy output untuk id 1 adalah Strongly preferred (0.8)

3.3 Proses Defuzzification

Pada proses ini, kita akan melakukan komposisi, yaitu agregasi hasil Clipping dari semua aturan fuzzy sehingga kita dapatkan satu fuzzy set tunggal. Pada kali ini kita akan menggunakan metode centroid atau model mamdani untuk melakukannya. Sesuai dengan yang dibahas pada bagian metodologi, maka kita akan mencari nilai crisp output (y^*). Misal, kita telah mendapatkan fuzzy output bernilai Strongly preferred (0.8). Maka kita ambil nilai sebarang pada daerah fungsi yang berkorespondensi lalu masukan pada rumus, sbb,

$$y^* = \frac{(65 + 70 + 75) * (0.8)}{3 * 0.8} = 69.99$$

Kita melakukan ketiga proses tersebut pada semua data yang ada pada file restoran.xlsx, sehingga hasil akhir yang diperoleh yaitu sepuluh restoran terbaik yaitu:

ID	Crisp Output (Nilai Kelayakan/Hasil Akhir)
24	90
42	90
79	87
69	86
6	80
22	80
25	80
31	80
54	80
20	79.09

4. SIMPULAN

Dalam sistem yang kami bangun, keseluruhannya mengikuti model yang diperkenalkan oleh Mamdani. Mulai dari penentuan fungsi keanggotaan yang berbentuk trapesium dan segitiga, dilanjutkan dengan proses fuzzification untuk mengubah nilai crisp yang ada pada data menjadi fuzzy input. Setelah itu, dilakukanlah inferensi untuk mengubah fuzzy input yang tadi dihasilkan menjadi fuzzy output. Nilai fuzzy ini memiliki rentan 0 s.d. 1. Derajat keanggotaan pada fuzzy output sangat dimungkinkan lebih dari satu. Untuk itu, kita lakukan defuzzification dengan metode centroid yaitu clipping sehingga menjadi satu set tunggal. Pada akhirnya, dari model fuzzy yang kita bangun, kita dapatkan 10 restoran terbaik yaitu 24, 42, 79, 69, 6, 22, 25, 31, 54, 20.

5. VIDEO

Ryan Abdurrohman : <https://youtu.be/Piib93tz1S0>

Muhammad Faiz Abdurrahman Djauhar : <https://youtu.be/nrgWm9nwFjY>

Azriel Naufal Aulia : https://youtu.be/DjJrgC2L_j0

DAFTAR PUSTAKA

Amindoust, A., Ahmed, S., Saghafinia, A., & Bahreininejad, A. (2012). Sustainable supplier selection: A ranking model based on fuzzy inference system. *Applied Soft Computing*, 12(6), 1668-1677. doi: 10.1016/j.asoc.2012.01.023

Suyanto, S. (2014). Artificial Intelligence: Searching-Reasoning-Planning-Learning (Revisi Kedua). Penerbit Informatika.