**ALVIN SETIAWAN 191011401509**

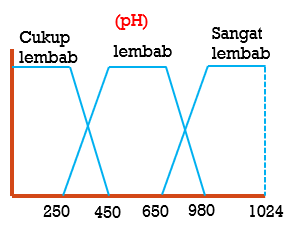
**1.Model Fuzzy**

Pemodelan merupakan proses memodelkan fuzzy. Diantaranya adalah menentukan parameter (variabel), himpunan keanggotaan, fungsi keanggotaan, variabel linguistik, dan inferensi fuzzy. Proses ini merupakan inti dari pembuatan sistem berbasis fuzzy. Adapun detail pemodelannya sebagai berikut :

1. **Variabel**

Variabel yang digunakan untuk membangun sistem kontrol penyiraman rumput diantaranya adalah :

1. Kelembaban tanah (Input)



**X** : Kelembaban tanah

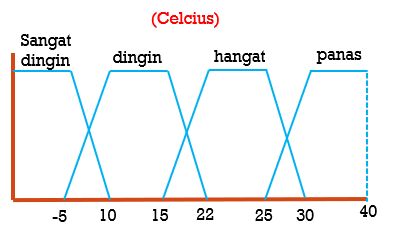
**T** : {cukup lembab, lembab, sangat lembab}

**U** : [1,1024]

**M** : cukup lembab, lembab, sangat lembab dengan nilai keanggotaannya  
(berupa kurvanya atau nilai keanggotaan μ).

Berdasarkan gambar diatas dapat direpresentasikan fungsi keanggotaannya sebagai berikut :

1. Suhu (Input)



**X** : Suhu

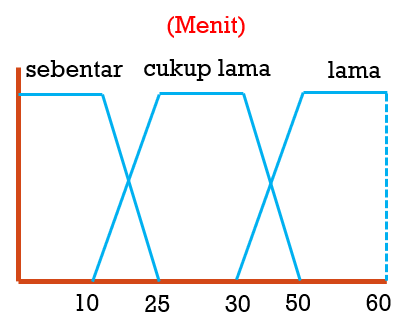
**T** : {sangat dingin, dingin, hangat, panas}

**U** : [-10,40]

**M** sangat dingin, dingin, hangat, panas dengan nilai keanggotaannya  
(berupa kurvanya atau nilai keanggotaan μ).

Berdasarkan gambar diatas dapat direpresentasikan fungsi keanggotaannya sebagai berikut :

1. Durasi Penyiraman (Output)



**X** : Suhu

**T** : {sebentar, cukup lama,lama}

**U** : [1,60]

**M** : sebentar, cukup lama,lama dengan nilai keanggotaannya (berupa kurvanya atau nilai keanggotaan μ).

1. **Rules**

|  |  |  |  |
| --- | --- | --- | --- |
| Rules  **IF** | Kelembabab | Suhu  **AND** | Output  **THEN** |
| 1 | cukup lembab | sangat dingin | cukup lama |
| 2 | cukup lembab | dingin | cukup lama |
| 3 | cukup lembab | hangat | lama |
| 4 | cukup lembab | panas | lama |
| 5 | lembab | sangat dingin | sebentar |
| 6 | lembab | dingin | sebentar |
| 7 | lembab | hangat | sebentar |
| 8 | lembab | panas | cukup lama |
| 9 | sangat lembab | sangat dingin | sebentar |
| 10 | sangat lembab | dingin | sebentar |
| 11 | sangat lembab | hangat | sebentar |
| 12 | sangat lembab | panas | sebentar |

1. **Implementasi**

Diasumsikan sebuah alat penyiraman berbasis IOT sedang melakukan penyiraman, diwaktu tersebut alat penyiraman mendeteksi kelembapan tanah **360pH** dan suhunya adalah **20** **derajat celcius**. Dengan demikian alat tersebut akan mengontrol durasi lama penyiraman sebagai berikut:

1. Fuzzifikasi

* Kelembaban tanah
* Suhu

1. Inferensi

IF lembab AND hangat THEN sebentar

MIN (0,55; 0,71) = **0,55**

IF lembab AND dingin THEN sebentar

MIN (0,55; 0,29) = **0,29**

IF cukup lembab AND hangat THEN lama

MIN (0,46; 0,71) = **0,46**

IF cukup lembab AND dingin THEN cukup lama

MIN (0,46; 0,29) = **0,29**

MAX(0,55;0,29) = **0,29** 🡪 sebentar

MAX(0,46) = **0,46** 🡪 lama

MAX(0,29) = **0,29** 🡪 cukup lama

1. Defuzzifukasi

Jadi durasi penyiraman hasil *controlling* dengan fuzzy adalah **29,97** atau jika dibulatkan menjadi **30 menit.**

1. **Coding**
2. import itertools
3. import numpy as np
4. kelembaban\_param = ["cukup lembab", "lembab","sangat lembab"]
5. dk\_kelembaban = [1, 250, 450, 650, 980, 1024]
6. temp\_param = ["sangat dingin","dingin","hangat","panas"]
7. dk\_temp = [-10, -5, 10, 15, 22, 25, 30, 40]
8. inp\_kelembaban = 360
9. inp\_temperatur   = 20
10. # Mencari nilai tengah dari interval bawah dan atas
11. def findCenter(nilai):
12. c = []
13. count = 0
14. center = 0
15. idx = 1
16. while idx <= len(nilai)-1:
17. if count == 2:
18. center = center / 2
19. c.append(center)
20. count = 0
21. center = 0
22. else:
23. center += nilai[idx]
24. count += 1
25. idx += 1
26. return c
27. c\_kelembaban = findCenter(dk\_kelembaban)
28. c\_temp = findCenter(dk\_temp)
29. # Fuzzification
30. def fuzzification(center, dk, input, param):
31. idx = 0
32. dic = {}
33. for i in range(len(center)):
34. batas\_kiri = 0
35. batas\_kanan = 0
36. fuzzi\_1 = 0
37. fuzzi\_2 = 0
38. for j in range(2):
39. idx+=1
40. batas\_kiri = dk[idx-1]
41. batas\_kanan = dk[idx]
42. if input < batas\_kiri:
43. kiri = (idx/2)-1
44. dic.update({param[int(kiri)]: 1})
45. return dic
46. break
47. if input >= batas\_kanan and input <= dk[idx+1]:
48. kiri = (idx/2)
49. dic.update({param[int(kiri)]: 1})
50. return dic
51. break
53. if input > batas\_kiri and input > batas\_kanan:
54. continue
55. else:
56. fuzzi\_1 = (batas\_kanan - input) / (batas\_kanan - batas\_kiri)
57. fuzzi\_2 = (input - batas\_kiri) / (batas\_kanan - batas\_kiri)
59. kanan = idx/2
60. kiri = (idx/2)-1
61. if input > center[i]:
62. dic.update({param[int(kanan)]: fuzzi\_2})
63. dic.update({param[int(kiri)]: fuzzi\_1})
64. else:
65. dic.update({param[int(kanan)]: fuzzi\_1})
66. dic.update({param[int(kiri)]: fuzzi\_2})
67. return dic
68. break
70. def getKeyVal(mydict):
71. key = []
72. val = []
73. for k,v in mydict.items():
74. key.append(k)
75. val.append(v)
76. return key,val
77. #Rules
78. def rules(inp1, inp2):
79. out = ""
80. if inp1 == "cukup lembab" and inp2 == "sangat dingin":
81. out = "cukup lama"
82. elif inp1 == "cukup lembab" and inp2 == "dingin":
83. out = "cukup lama"
84. elif inp1 == "cukup lembab" and inp2 == "hangat":
85. out = "lama"
86. elif inp1 == "cukup lembab" and inp2 == "panas":
87. out = "lama"
88. elif inp1 == "lembab" and inp2 == "sangat dingin":
89. out = "sebentar"
90. elif inp1 == "lembab" and inp2 == "dingin":
91. out = "sebentar"
92. elif inp1 == "lembab" and inp2 == "hangat":
93. out = "sebentar"
94. elif inp1 == "lembab" and inp2 == "panas":
95. out = "cukup lama"
96. elif inp1 == "sangat lembab" and inp2 == "sangat dingin":
97. out = "sebentar"
98. elif inp1 == "sangat lembab" and inp2 == "dingin":
99. out = "sebentar"
100. elif inp1 == "sangat lembab" and inp2 == "hangat":
101. out = "sebentar"
102. elif inp1 == "sangat lembab" and inp2 == "panas":
103. out = "sebentar"
105. return out
106. dic\_kelembaban = fuzzification(c\_kelembaban, dk\_kelembaban, inp\_kelembaban, kelembaban\_param)
107. dic\_temp = fuzzification(c\_temp, dk\_temp, inp\_temperatur, temp\_param)
108. # print(dic\_kelembaban)
109. # print(dic\_temp)
110. kelembabankey, kelembabanval = getKeyVal(dic\_kelembaban)
111. tempkey, tempval = getKeyVal(dic\_temp)
112. key = []
113. for x in itertools.product(kelembabankey, tempkey):
114. key.append(rules(x[0],x[1]))
115. val = []
116. for x in itertools.product(kelembabanval, tempval):
117. val.append(np.min(x))
118. print(key)
119. max\_data = {}
120. for i in range(len(key)):
121. for j in range(len(val)):
122. if key[i] == key[j]:
123. if max\_data.get(key[i]) != None and val[i] > max\_data.get(key[i]):
124. max\_data.update({key[i]: val[i]})
125. else:
126. max\_data.update({key[i]: val[i]})
127. # Defuzzifikasi
128. defuzzy = []
129. y = []
130. for k,v in max\_data.items():
131. y.append(v)
132. if k == "sebentar":
133. tmp = (25 - (v\*15))
134. # print(25, "-",v,"\*15 = ", tmp)
135. defuzzy.append(tmp)
136. elif k == "cukup lama":
137. tmp = (10 + (v\*15))
138. # print(10, "+",v,"\*15=", tmp)
139. defuzzy.append(tmp)
140. elif k == "lama":
141. tmp = (30 + (v\*20))
142. # print(30, "+",v,"\*20=", tmp)
143. defuzzy.append(tmp)
144. sum\_nilai = 0
145. sum\_y = 0
146. for i in range(len(defuzzy)):
147. sum\_nilai += (defuzzy[i]\*y[i])
148. sum\_y += y[i]
149. defuzifikasi = sum\_nilai/sum\_y
150. print("Durasi Penyiraman rumput dengan kelembaban",inp\_kelembaban,"pH & suhu", inp\_temperatur,"celcius adalah",defuzifikasi,"menit")
151. **Hasil Running**

