

LAPORAN TUGAS PROGRAM 2
ARTIFICIAL INTELLIGENCE
FUZZY LOGIC



Nama : I Putu Indra Aristya

NIM : 1301154219

Kelas : IF39-09

Mata Kuliah : Kecerdasan Buatan

Universitas Telkom

BANDUNG

2017

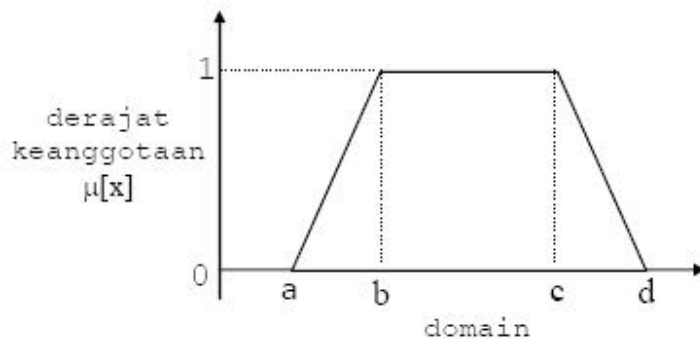
I. Deskripsi Masalah

Dalam tugas program-2 ini, kasus yang diberikan adalah menentukan dan mendesain bentuk fungsi keanggotaan, aturan-aturan (*rules*) dan hasil akhir dari sebuah data berita yang diberikan dengan menggunakan *Fuzzy Logic*. Data yang diberikan memiliki 2 data *input* dan 1 data *output*. Data *input* yang diberikan adalah tingkat emosi dan provokasi dari skala 1 sampai dengan 100 dan data *output* yang diberikan adalah *hoax* atau tidak *hoax*.

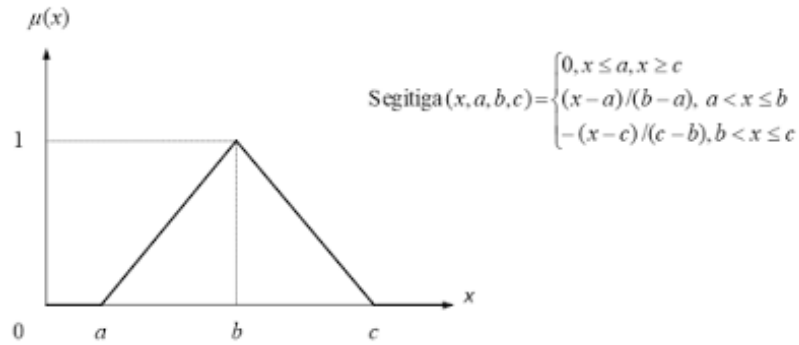
II. Landasan Teori

Fuzzy logic adalah salah satu metode yang dapat digunakan untuk merepresentasikan bahasa formal yang dapat dipahami oleh komputer. Dalam *fuzzy logic* ini terdapat beberapa istilah yang sering digunakan, yaitu *crisp set* (himpunan nilai asli dari sebuah ketentuan), *fuzzy set* (himpunan nilai *fuzzy* yang didapat dengan memproses *crisp* input), fungsi keanggotaan (grafik fungsi yang berisi batas nilai dan nilai linguistiknya). Dalam *fuzzy logic* ini terdapat 3 tahap yang harus dilalui, yaitu *Fuzzification*, *Inferences* dan *Defuzzification*.

Pada tahap *Fuzzification*, dilakukan pengubahan nilai input (*crisp* input) menjadi *fuzzy output*. Pengubahan dilakukan dengan menghitungnya sesuai dengan fungsi keanggotaan. Fungsi keanggotaan terdiri dari beberapa bagian, yaitu fungsi *sigmoid*, fungsi *phi*, fungsi segitiga dan fungsi trapesium. Dan dalam penyelesaian masalah ini saya menggunakan fungsi segitiga dan fungsi trapesium. Perhitungan yang ada dalam fungsi segitiga dan trapesium adalah sebagai berikut.



Gambar 1 Fungsi Keanggotaan Trapesium



Gambar 2 Fungsi Keanggotaan Segitiga

Fuzzy input terdiri atas nilai hasil perhitungan dengan nilai linguistik sesuai dengan yang ada pada grafik fungsi tersebut.

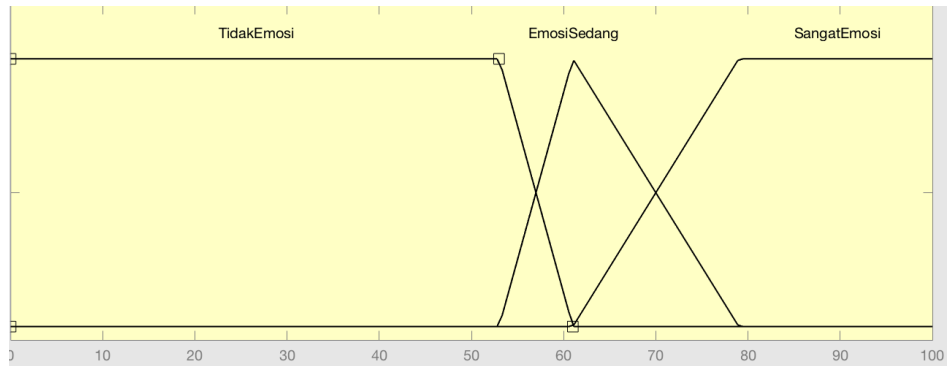
Pada tahap *Inferences*, dilakukan pengecekan hasil dari semua *fuzzy output* setiap fungsi keanggotaan input yang dipasangkan dengan setiap fungsi keanggotaan input yang lain. Pada proses ditentukan aturan-aturan yang berjumlah jumlah nilai linguistik fungsi keanggotaan *input* 1 dikali jumlah nilai linguistik fungsi keanggotaan input 2 dan seterusnya apabila terdapat lebih dari 2 fungsi keanggotaan input.

Pada tahap *Defuzzification*, dilakukan proses pengubahan *fuzzy input* dari hasil proses *Inferences* menjadi *crisp value*. Tahap ini menerapkan proses Mamdani atau Sugeno yang pada tugas ini saya menggunakan proses Sugeno.

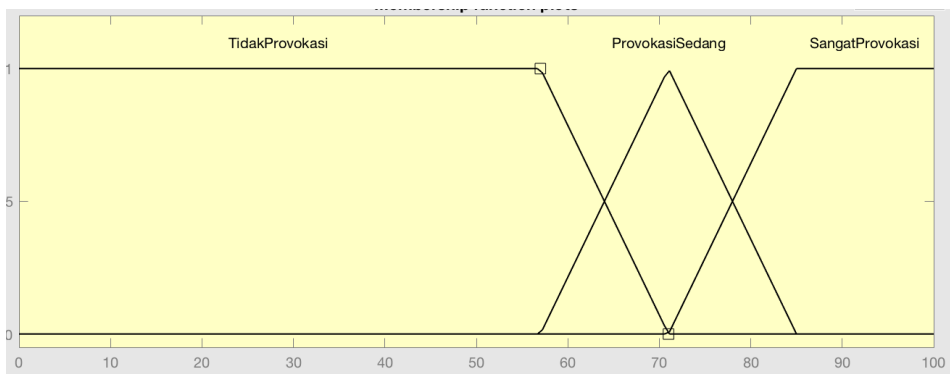
III. Desain *Fuzzy*

Untuk dapat menyelesaikan tugas program ini, ada beberapa hal yang saya desain untuk menyelesaikannya. Hal-hal tersebut adalah rancangan fungsi keanggotaan input, *rules*, dan fungsi keanggotaan output.

Fungsi Keanggotaan Input



Gambar 3 Fungsi Keanggotaan Input (Emosi)

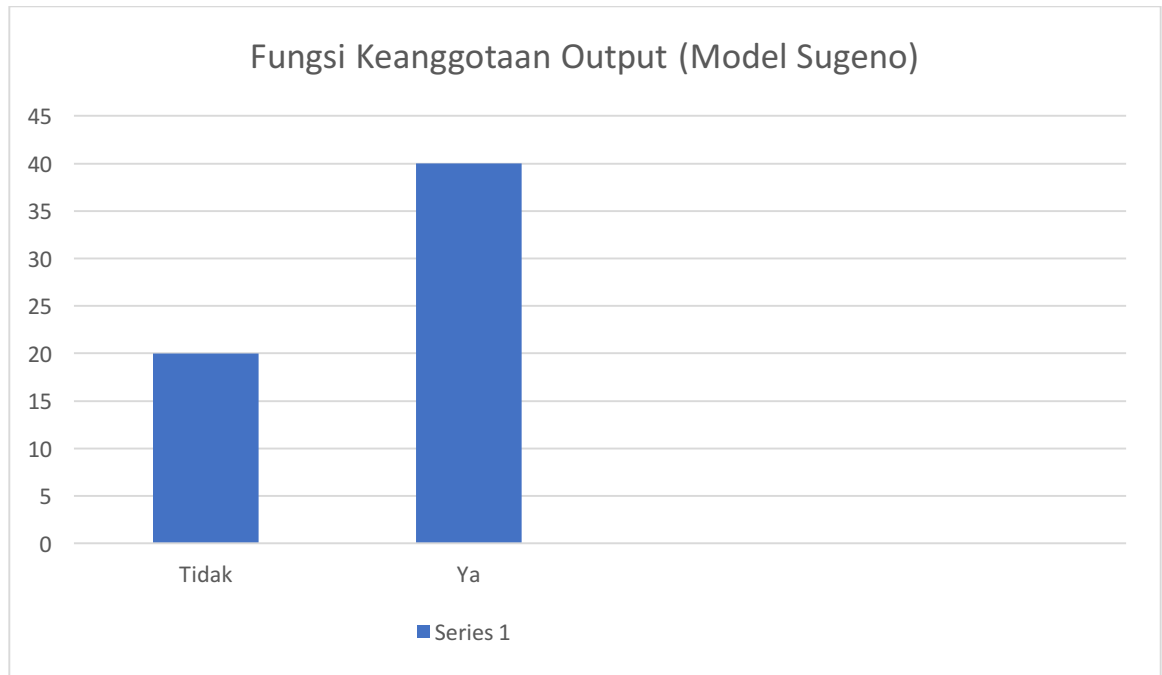


Gambar 4 Fungsi Keanggotaan Input (Provokasi)

Rules

Emosi Provokasi	Tidak Emosi	Emosi Sedang	Sangat Emosi
Tidak Provokasi	Tidak	Tidak	Tidak
Provokasi Sedang	Tidak	Tidak	Ya
Sangat Provokasi	Ya	Ya	Ya

Fungsi Keanggotaan Output



IV. Code Program

```
datasetEmosi =  
[97,36,63,82,71,79,55,57,40,57,77,68,60,82,40,80,60,50,100,11]  
datasetProvokasi =  
[74,85,43,90,25,81,62,45,65,45,70,75,70,90,85,68,72,95,18,99]  
datasetHasil =  
["Ya","Ya","Tidak","Ya","Tidak","Ya","Tidak","Tidak","Tidak","Tidak","Ya"  
,"Ya","Tidak","Ya","Tidak","Ya","Tidak",  
"Ya","Tidak","Ya"]  
  
datatestEmosi = [58,68,64,57,77,98,91,50,95,27]  
datatestProvokasi = [63,70,66,77,55,64,59,95,55,79]  
datatestHasil =  
["Tidak","Tidak","Tidak","Tidak","Tidak","Tidak","Tidak","Tidak","Tidak",  
"Tidak"]  
  
def FuzzyProg(a, b):  
    emosi = a  
    tidakEmosi = 0  
    emosiSedang = 0  
    sangatEmosi = 0  
  
    provokasi = b  
    tidakProvokasi = 0  
    provokasiSedang = 0  
    terProvokasi = 0
```

```

hoaxTemp = 0
hoax = 0
tidakhoaxTemp = 0
tidakhoax = 0

#Tahap Fuzzification
if (emosi <= 53):
    tidakEmosi = 1
    emosiSedang = 0
    sangatEmosi = 0
elif (emosi >=79):
    tidakEmosi = 0
    emosiSedang = 0
    sangatEmosi = 1
elif (emosi == 61):
    tidakEmosi = 0
    emosiSedang = 1
    sangatEmosi = 0
else:
    if (emosi > 53 and emosi < 61):
        tidakEmosi = -(emosi-61)/(61-53)
        emosiSedang = (emosi-53)/(61-53)
        sangatEmosi = 0
    elif (emosi > 61 and emosi < 79):
        tidakEmosi = 0
        emosiSedang = -(emosi-79)/(79-61)
        sangatEmosi = (emosi-61)/(79-61)

if (provokasi <= 57):
    tidakProvokasi = 1
    provokasiSedang = 0
    terProvokasi = 0
elif (provokasi >= 85):
    tidakProvokasi = 0
    provokasiSedang = 0;
    terProvokasi = 1
elif (provokasi == 71):
    tidakProvokasi = 0;
    provokasiSedang = 1;
    terProvokasi = 0;
elif (provokasi > 57 and provokasi < 71):
    tidakProvokasi = -(provokasi-71)/(71-57)
    provokasiSedang = (provokasi - 57)/(75-57)
    terProvokasi = 0
elif (provokasi > 71 and provokasi < 85):
    tidakProvokasi = 0
    provokasiSedang = -(provokasi-85)/(85-71)
    terProvokasi = (provokasi - 71)/(85-71)

#Tahap Inference
if (tidakEmosi != 0):
    if (tidakProvokasi != 0):
        tidakhoaxTemp = min(tidakEmosi,tidakProvokasi)
        if (tidakhoax < tidakhoaxTemp):
            tidakhoax = tidakhoaxTemp

```

```

        if (provokasiSedang != 0):
            tidakhoaxTemp = min(tidakEmosi, provokasiSedang)
            if (tidakhoax < tidakhoaxTemp):
                tidakhoax = tidakhoaxTemp
        if (terProvokasi != 0):
            hoaxTemp = min(tidakEmosi, terProvokasi)
            if (hoax < hoaxTemp):
                hoax = hoaxTemp

    if (emosiSedang != 0):
        if (tidakProvokasi != 0):
            tidakhoaxTemp = min(emosiSedang, tidakProvokasi)
            if (tidakhoax < tidakhoaxTemp):
                tidakhoax = tidakhoaxTemp
        if (provokasiSedang != 0):
            tidakhoaxTemp = min(emosiSedang, provokasiSedang)
            if (tidakhoax < tidakhoaxTemp):
                tidakhoax = tidakhoaxTemp
        if (terProvokasi != 0):
            hoaxTemp = min(emosiSedang, terProvokasi)
            if (hoax < hoaxTemp):
                hoax = hoaxTemp

    if (sangatEmosi != 0):
        if (tidakProvokasi != 0):
            tidakhoaxTemp = min(sangatEmosi, tidakProvokasi)
            if (hoax < tidakhoaxTemp):
                tidakhoax = tidakhoaxTemp
        if (provokasiSedang != 0):
            hoaxTemp = min(sangatEmosi, provokasiSedang)
            if (hoax < hoaxTemp):
                hoax = hoaxTemp
        if (terProvokasi != 0):
            hoaxTemp = min(sangatEmosi, terProvokasi)
            if (hoax < hoaxTemp):
                hoax = hoaxTemp

    #Tahap Defuzzification
    strhoax = str(hoax)
    strtidakhoax = str(tidakhoax)
    nilai1 = (hoax * 40)
    nilai2 = (tidakhoax * 20)
    jumlah = hoax + tidakhoax
    nilai = (nilai1 + nilai2)/jumlah

    if (nilai >= 22):
        kesimpulan = 'Ya'
    else:
        kesimpulan = 'Tidak'

    print(kesimpulan)
    return kesimpulan

print("-----")
print("DATA TRAIN")
print("-----")
count = 0

```

```

benar = 0
while count < 20:
    train = datasetHasil[count]
    if (FuzzyProg(datasetEmosi[count],datasetProvokasi[count]) ==
train):
        benar = benar + 1
        count = count + 1

print("")
print("Akurasi: "+str((benar/20)*100))
print("-----")
print("")
print("-----")
print("DATA TEST")
print("-----")
print("")

countTest = 0
while countTest < 10:
    datasetHasil[countTest] =
FuzzyProg(datatestEmosi[countTest],datatestProvokasi[countTest])
    countTest = countTest + 1

```

V. Hasil Keluaran Program

Berikut adalah hasil keluaran dari program yang dijalankan.

```

-----
DATA TRAIN
-----
Ya
Ya
Tidak
Ya
Tidak
Ya
Tidak
Tidak
Tidak
Tidak
Ya
Ya
Tidak
Ya
Ya
Ya
Ya
Tidak
Ya
Tidak
Ya
Ya

Akurasi: 95.0
-----

DATA TEST
-----
Tidak
Ya
Ya
Ya
Tidak
Ya
Ya
Ya
Ya
Tidak
Ya

Process finished with exit code 0

```

Gambar 5 Hasil Keluaran Program

VI. Kesimpulan

Fuzzy Logic dapat digunakan untuk penentuan suatu keputusan yang bersifat detail pada nilai linguistiknya. Dari hasil yang saya dapatkan, dengan fungsi keanggotaan input emosi, fungsi kanggotaan input provokasi dan rules pada inference, didapatkan 19 data benar (sesuai dengan *datatrain*) dari 20 data yang diberikan. Dengan akurasi 95%.

Daftar Pustaka

Suyanto. (2014). *Artificial Intelligence* (Vol. 2). Bandung: Informatika Bandung.