Daftar Isi

[Bab 1 Kajian Pustaka 3](#_Toc496877836)

[1.1 Pengertian Fuzzy Logic 3](#_Toc496877837)

[1.2 Konsep Dasar Fuzzy Logic 4](#_Toc496877838)

[1.3 Himpunan Fuzzy 5](#_Toc496877839)

[Gambar 2.3 Himpunan Konvensional (Sumber : Negnevitsky, 2005, p91). 6](#_Toc496877840)

[1.4 Fungsi Keanggotaan 6](#_Toc496877841)

[1.5 Operasi Himpunan Fuzzy 6](#_Toc496877842)

[1.6 Aturan Fuzzy 7](#_Toc496877843)

[Gambar 2.6 Fungsi Implikasi Min (Sumber ; Kusumadewi, 2003:180). 7](#_Toc496877844)

[Gambar 2.7 Fungsi implikasi dot (Sumber: Kusumadewi, 2003:180). 8](#_Toc496877845)

[1.7 Sistem Inferensi Fuzzy 8](#_Toc496877846)

[1.8 Aturan Program 10](#_Toc496877847)

[1.9 Spesifikasi Program 12](#_Toc496877848)

[1.10 Metode Pembuatan Program Fuzzy 12](#_Toc496877849)

[1.10.1 Fuzzification 13](#_Toc496877850)

[1.10.2 Inference 15](#_Toc496877851)

[1.10.3 Defuzzyfication 17](#_Toc496877852)

[1.10.4 Hasil 18](#_Toc496877853)

[Bab III Kesimpulan 21](#_Toc496877854)

[Daftar Pustaka 22](#_Toc496877855)

# Bab 1 Kajian Pustaka

## Pengertian Fuzzy Logic

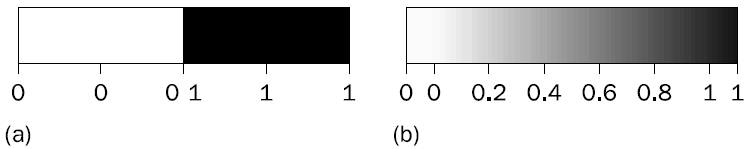
*Fuzzy logic* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke ruang *output* (Kusumadewi, 2003:153). *Fuzzy logic* dapat mengolah nilai yang tidak pasti berupa batasan, seperti “sangat”, “sedikit”, “kurang lebih”. Manusia dapat dengan mudah mengartikan kalimat “Saya pergi sebentar saja”, mungkin sebentar itu empat menit atau lima menit. Namun lain halnya dengan komputer, yang hanya menerima masukkan berupa bilangan *crisp*, seperti “Saya pergi 5 menit saja”. Komputer tidak mengerti nilai asli dari kata “sebentar”. Dengan *fuzzy logic*, komputer dapat mengolah ketidakpastian tersebut, sehingga dapat digunakan untuk memutuskan sesuatu yang membutuhkan kepintaran manusia dalam penalaran.

*Fuzzy logic* pertama kali diperkenalkan oleh Jan Lukasiewicz pada tahun 1920an sebagai teori kemungkinan. Logika kemungkinan ini memperluas jangkauan dari nilai kebenaran untuk semua bilangan riil pada interval antara 0 dan 1. Selanjutnya diteliti lebih lanjut oleh Max Black pada tahun 1930an dalam penelitiannya tentang ketidakjelasan (*vagueness*): sebuah latihan pada analisis logis. Pada tahun 1965, Professor dan kepala departemen teknik elektrik di Universitas California di Berkeley, Lotfi Zadeh, menemukan kembali, mengidentifikasi dan mengeksplorasi, dan mempromosikan dan berjuang untuk *fuzzy logic*. Professor Zadeh memperluas ruang kerja teori kemungkinan menjadi sistem logika matematika formal, dan konsep baru untuk mengaplikasikan istilah bahasa alami pada penelitiannya yaitu ‘*Fuzzy sets*’. Logika baru ini dinamakan *fuzzy logic*. *Fuzzy logic* banyak digunakan karena *fuzzy logic* mirip dengan cara berpikir manusia. Sistem *fuzzy logic* dapat merepresentasikan pengetahuan manusia dalam bentuk matematis dengan menyerupai cara berpikir manusia (Negnevitsky, 2005:89).

*Fuzzy logic* adalah suatu logika yang merepresentasikan cara berpikir manusia dalam bentuk matematis yang dapat mengolah ketidakpastian dan variabel-variabel linguistik.

## Konsep Dasar *Fuzzy Logic*

Seperti logika klasik, *fuzzy logic* berkaitan dengan kebenaran proposisi. Namun, proposisi di dunia nyata sering hanya sebagian benar. Selain itu, sering digunakan istilah-istilah yang tidak didefinisikan secara jelas. Contohnya, sulit untuk menggambarkan kebenaran “John sudah tua” bernilai benar atau salah jika John berumur 60 tahun. Dalam beberapa hal, John pada 60 tahun sudah cukup tua untuk memenuhi syarat untuk mendapat keuntungan warga senior di berbagai segi, tetapi dalam hal lain John tidak cukup tua karena dia tidak memenuhi syarat jaminan sosial. Jadi diperlukan nilai kebenaran dari “John sudah tua” untuk mendapat nilai antara [0,1], tidak hanya 0 atau 1 (Siler, 2005:36-37).



**Gambar 2.2 (a) Konsep logika Boolean, hanya terdiri 2 nilai dari 0 dan 1(b) konsep *fuzzy logic* terdiri dari banyak nilai** (Sumber : Negnevitsky, 2005:89).

Tidak seperti logika Boolean yang memiliki 2 nilai, *fuzzy logic* terdiri dari banyak nilai. *Fuzzy logic* menangani derajat keanggotaan dan derajat kebenaran. *Fuzzy logic* menggunakan nilai berkelanjutan antara 0 (sepenuhnya salah) dan 1 (sepenuhnya benar). Tidak hanya hitam dan putih, *fuzzy logic* mencakup spektrum warna, menandakan bahwa elemen-elemen bisa sebagian benar dan sebagian salah pada waktu yang sama (Negnevitsky, 2005:89).

Nilai keabuan pada *fuzzy logic* menunjukkan seberapa besar suatu elemen masuk dalam suatu kelompok. Dimana dalam kehidupan sehari-hari, pembagian elemen-elemen tertentu tidak dapat dinyatakan secara tegas, namun elemen satu dengan yang lain dapat berbagi kelompok.

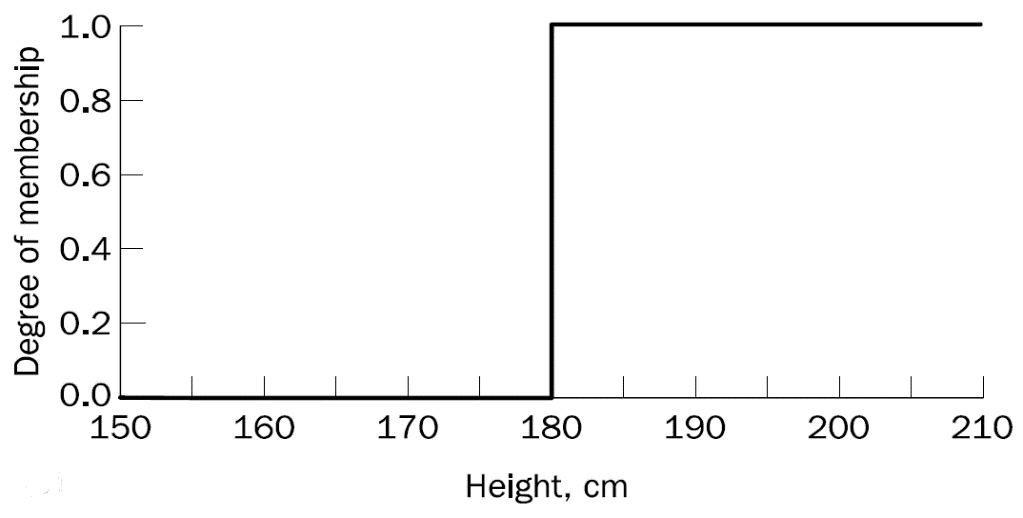
### Himpunan *Fuzzy*

Himpunan *fuzzy* adalah kumpulan prinsip matematik sebagai penggambaran pengetahuan berdasarkan derajat keanggotaan daripada menggunakan derajat *crisp* dari logika biner klasik (Negnevitsky, 2005:90).

Menurut Ross (2010:34), sebuah himpunan *fuzzy* adalah sebuah himpunan yang mengandung elemen-elemen yang mempunyai derajat keanggotaan yang bervariasi dalam himpunan. Ini berlawanan dengan himpunan klasik karena anggota dari sebuah himpunan klasik tidak mungkin menjadi anggota kecuali memiliki derajat keanggotaan penuh dalam himpunan. Karena elemen-elemen di sebuah himpunan *fuzzy* tidak perlu lengkap, maka elemen-elemen tersebut juga bisa masuk menjadi anggota himpunan *fuzzy* yang lain pada semesta yang sama.

Dalam himpunan konvensional, sebuah elemen *x* akan masuk dalam himpunan *A* jika memiliki nilai keanggotaan 1, dan tidak masuk dalam himpunan *A* jika memiliki nilai keanggotaan 0 (Konar, 2005:39).

Himpunan konvensional dapat dituliskan dalam bentuk matematis, sebagai berikut (Negnevitsky, 2005:91):



Tall

**Gambar 2.3 Himpunan Konvensional** (Sumber : Negnevitsky, 2005, p91).

## Fungsi Keanggotaan

Tingkat keanggotaan memetakan objek atau atributnya *(x)* ke bilangan riil positif pada interval [0,1]. Karena karakteristik pemetaannya seperti sebuah fungsi, maka disebut sebagai fungsi keanggotaan. Mengacu pada Konar (2005:41) definisi formal fungsi keanggotaan adalah

“Sebuah fungsi keanggotaan dikarakteristikan dengan pemetaan berikut:

Dimana *x* adalah sebuah bilangan riil yang mendeskripsikan sebuah objek atau atributnya dan *X* adalah semesta pembicaraan dan *A* adalah himpunan bagian dari *X*.”

## Operasi Himpunan *Fuzzy*

Mengacu pada Kusumadewi (2003:175-176), himpunan *fuzzy* memiliki 3 operasi *fuzzy* dasar yang diciptakan oleh Zadeh, yaitu :

1. Operator AND (*Intersection*)

Nilai dari hasil interseksi dari himpunan A dan B adalah dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan A dan B.

1. Operator OR (*Union*)

Nilai dari hasil penggabungan dari himpunan A dan B adalah dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan A dan B.

1. Operator NOT (*Complement*)

Nilai dari hasil komplemen dari himpunan A adalah dengan mengurangkan nilai keanggotaan elemen pada himpunan A dari 1.

## Aturan *Fuzzy*

Aturan *fuzzy* adalah sebuah pernyataan berkondisi dalam bentuk:

IF *x* is *A* THEN *y* is *B*

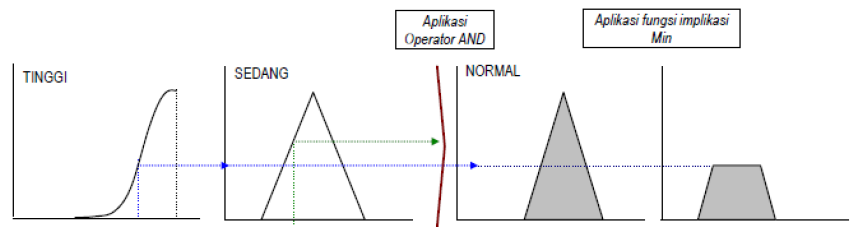
Dimana *x* dan *y* adalah variabel linguistik, dan *A* dan *B* adalah nilai linguistik yang ditentukan oleh himpunan-himpunan *fuzzy* pada semesta pembicaraan *X* dan *Y* berurutan (Negnevitsky, 2005:103).

1. **Relasi Implikasi *Fuzzy***

Menurut Kusumadewi (2003:179-180), secara umum terdapat 2 fungsi implikasi yang dapat digunakan, yaitu:

1. *Min* (*minimum*)

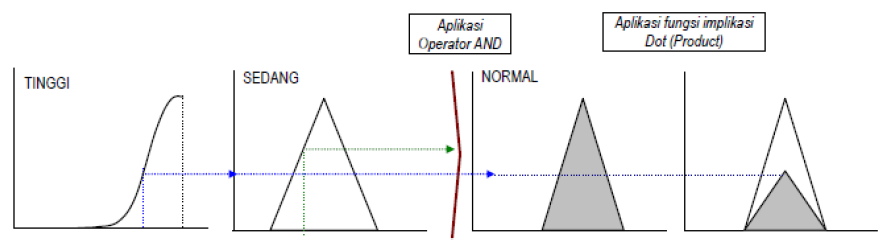
Fungsi ini akan memotong *output* himpunan *fuzzy*.



**Gambar 2.6 Fungsi Implikasi *Min*** (Sumber ; Kusumadewi, 2003:180).

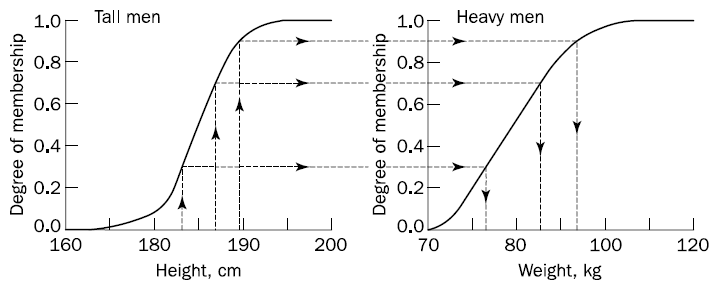
1. Dot (*product*)

Fungsi ini akan menskala *output* himpunan *fuzzy*.



**Gambar 2.7 Fungsi implikasi dot** (Sumber: Kusumadewi, 2003:180).

Penalaran pada sistem berbasis aturan, jika anteseden bernilai benar, maka konsekuen juga bernilai benar. Pada sistem *fuzzy*, dimana anteseden adalah pernyataan *fuzzy*. Jika antesenden bernilai benar pada beberapa derajat keanggotaan, maka konsekuennya juga bernilai benar pada derajat yang sama (Negnevitsky, 2005:104). Penalaran ini disebut dengan penalaran monoton. Mengacu pada Kusumadewi (2003:177), penalaran monoton ini sudah jarang digunakan, dimana nilai *output* dapat diestimasi secara langsung berdasar pada derajat keanggotaan dari antesendennya.



**Gambar 2.8 Penalaran monoton dari tinggi badan ke berat badan** (Negnevitsky, 2005:105).

## Sistem Inferensi *Fuzzy*

Negnevitsky (2005:106) mengatakan, inferensi *fuzzy* dapat didefinisikan sebagai proses pemetaan dari *input* yang diberikan ke dalam *output*, menggunakan teori dari himpunan *fuzzy*.

Sistem inferensi *fuzzy* adalah kerangka kerja matematis berdasarkan pada konsep teori himpunan *fuzzy*, aturan *fuzzy*, dan perhitungan *fuzzy*. Dimana sistem inferensi *fuzzy* mengambil *input* baik *crisp* ataupun *fuzzy*, dan diubah menjadi nilai *fuzzy* untuk dimasukkan kedalam himpunan *fuzzy*. Setelah himpunan *fuzzy* didapatkan dilakukan agregasi terhadap aturan yang telah didefinisikan. Hasil dari agregasi aturan tersebut akan diubah nilainya menjadi nilai *crisp*.

**Bab II Cara Kerja Program**

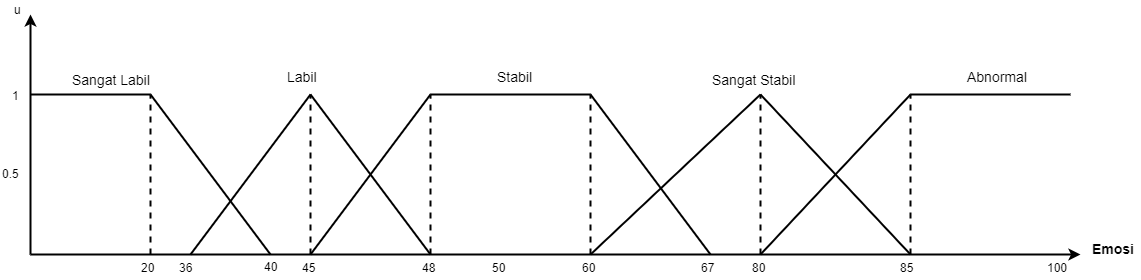
## Aturan Program

Diberikan suatu himpunan data berisi 30 berita dengan dua atribut: Emosi dan Provokasi yang bernilai 0 sampai 100, serta atribut kelas *Hoax* yang bernilai “Ya” dan “Tidak”, seperti diilustasikan pada tabel berikut. Gunakan dua puluh berita pertama, B01 sampai B20, sebagai acuan untuk membangun sebuah sistem penalaran berbasis *fuzzy logic* untuk mendeteksi apakah sepuluh berita yang belum diketahui kelasnya, B21 sampai B30 (Data Testing), adalah berita bohong (*hoax*) atau bukan.

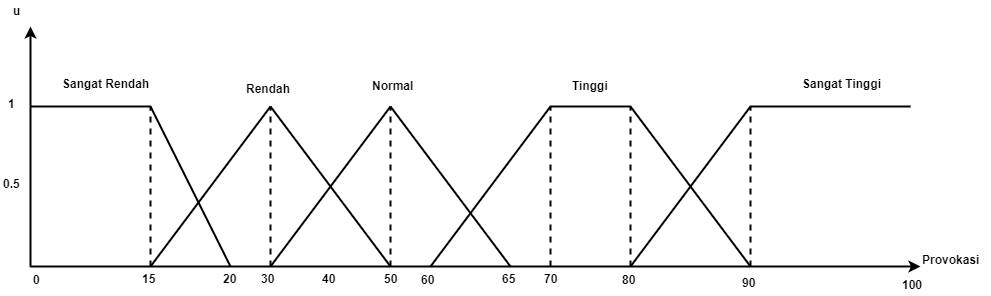
|  |  |  |  |
| --- | --- | --- | --- |
| **Berita** | **Emosi** | **Provokasi** | ***Hoax*** |
| B01 | 97 | 74 | Ya |
| B02 | 36 | 85 | Ya |
| B03 | 63 | 43 | Tidak |
| B04 | 82 | 90 | Ya |
| B05 | 71 | 25 | Tidak |
| B06 | 79 | 81 | Ya |
| B07 | 55 | 62 | Tidak |
| B08 | 57 | 45 | Tidak |
| B09 | 40 | 65 | Tidak |
| B10 | 57 | 45 | Tidak |
| B11 | 77 | 70 | Ya |
| B12 | 68 | 75 | Ya |
| B13 | 60 | 70 | Tidak |
| B14 | 82 | 90 | Ya |
| B15 | 40 | 85 | Tidak |
| B16 | 80 | 68 | Ya |
| B17 | 60 | 72 | Tidak |
| B18 | 50 | 95 | Ya |
| B19 | 100 | 18 | Tidak |
| B20 | 11 | 99 | Ya |
| B21 | 58 | 63 |  |
| B22 | 68 | 70 |  |
| B23 | 64 | 66 |  |
| B24 | 57 | 77 |  |
| B25 | 77 | 55 |  |
| B26 | 98 | 64 |  |
| B27 | 91 | 59 |  |
| B28 | 50 | 95 |  |
| B29 | 95 | 55 |  |
| B30 | 27 | 79 |  |

Dari soal yang telah diberikkan pada tugas program, kami mendefinisikkan dengan gambar fungsi keanggotaan sebagai berikut

1. **Fungsi Keanggotaan Emosi**



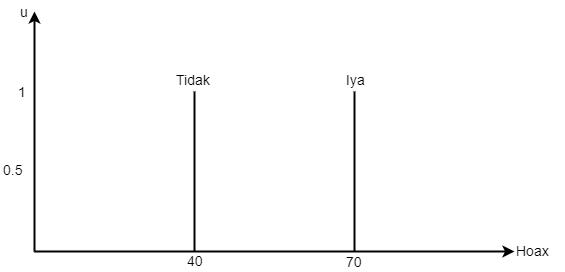
1. **Fungsi Keanggotaan Provokasi**



Dari fungsi keanggotaan tersebut, dapat disimpulkan fuzzy rulessebagai berikut yang telah saya definisikkan dari gambar berdasarkan dataset berita B01 sampai B020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Abnormal | Tidak | Tidak | Tidak | Iya | Iya |
| Sangat Stabil | Tidak | Tidak | Tidak | Iya | Iya |
| Stabil | Tidak | Tidak | Tidak | Tidak | Iya |
| Labil | Tidak | Tidak | Tidak | Tidak | Tidak |
| Sangat Labil | Tidak | Tidak | Tidak | Iya | Iya |
|  | Sangat Rendah | Rendah | Normal | Tinggi | Sangat Tinggi |

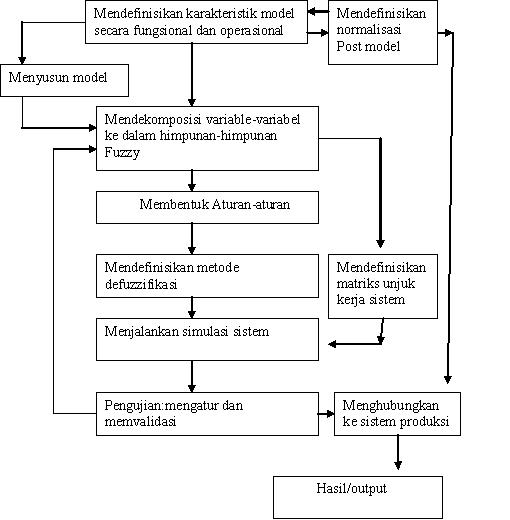
Dari Rules tersebut saya membuat program ini dengan metode sugeno untuk mendefinisikkan berita dalam kategori hoax atau tidak, berikut gambarnya



## Spesifikasi Program

1. Program Simulated Annealing ini dibangun menggunakan Bahasa pemograman python dengan interpreter 3.6
2. IDE yang digunakan dalam membangun program ini adalah PyChram Prefisional 2017.1
3. Program ini dapat dijalankan di system operasi windows atau linux

## Metode Pembuatan Program Fuzzy



### Fuzzification

#### Fuzzy Input

emosional = []  
provokasi = []  
akhirdefuzzyficatin = 0

kode diatas adalah array yang digunakkan untuk menampung inputan

**def trapesium**(x, a, b, c, d):  
 **if** (x <= a **and** x >= d):  
 **return** 0  
 **elif** (a < x < b):  
 **return** (x - a) / float(b - a)  
 **elif** (b <= x <= c):  
 **return** 1  
 **elif** (c < x < d):  
 **return** -(x - d) / float(d - c)

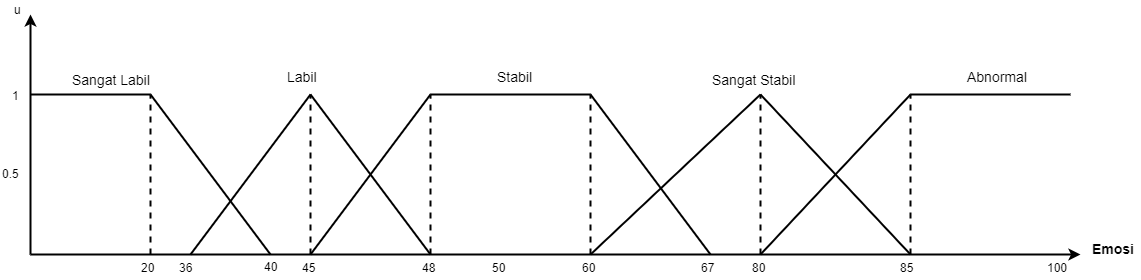
Fungsi kode program diatas adalah fungsi membership trapesium, dengan mengikuti kaidah rules fuzzy

**def segitiga**(x, a, b, c):  
 **if** (x <= a **and** x >= c):  
 **return** 0  
 **elif** (a < x <= b):  
 **return** (x - a) / float(b - a)  
 **elif** (b < x <= c):  
 **return** -(x - c) / float(c - b)

fungsi baris kode diatas adalah fungsi membership segitiga, dengan mengikuti kaidah rules fuzzy

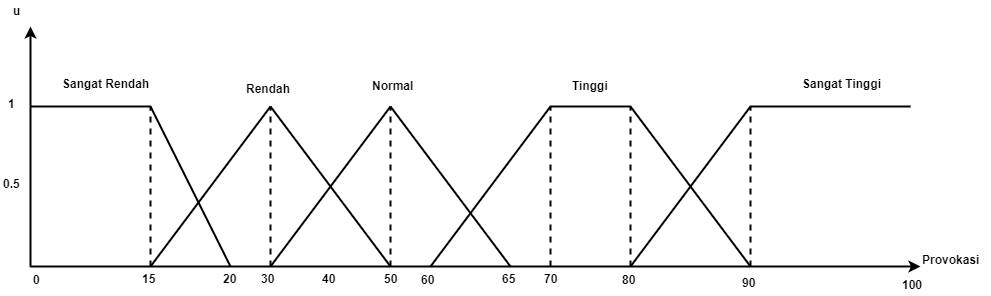
**def membershipemosi**(inputemosi):  
 **if** (inputemosi >= 0 **and** inputemosi < 40):  
 emosional.append(["Sangat Labil", trapesium(inputemosi, 0, 0, 20, 40)])  
 **if** (inputemosi > 36 **and** inputemosi < 48):  
 emosional.append(["Labil", segitiga(inputemosi, 36, 45, 48)])  
 **if** (inputemosi > 45 **and** inputemosi < 67):  
 emosional.append(["Stabil", trapesium(inputemosi, 45, 48, 60, 67)])  
 **if** (inputemosi > 60 **and** inputemosi < 85):  
 emosional.append(["Sangat Stabil", segitiga(inputemosi, 60, 80, 85)])  
 **if** (inputemosi > 80 **and** inputemosi <= 100):  
 emosional.append(["Abnormal", trapesium(inputemosi, 80, 85, 100, 100)])

fungsi baris kode diatas adalah untuk menentukan membership function Emosi, dengan Inputan seperti fungsi keanggotaan dibawah ini, yang sudah didefinisikkan dari soal



**def membershipprovokasi**(inputprovokasi):  
 **if** (inputprovokasi >= 0 **and** inputprovokasi < 20):  
 provokasi.append(["Sangat Rendah", trapesium(inputprovokasi, 0, 0, 15, 20)])  
 **if** (inputprovokasi > 15 **and** inputprovokasi < 50):  
 provokasi.append(["Rendah", segitiga(inputprovokasi, 15, 30, 50)])  
 **if** (inputprovokasi > 30 **and** inputprovokasi < 65):  
 provokasi.append(["Normal", segitiga(inputprovokasi, 30, 50, 65)])  
 **if** (inputprovokasi > 60 **and** inputprovokasi < 90):  
 provokasi.append(["Tinggi", trapesium(inputprovokasi, 60, 70, 80, 90)])  
 **if** (inputprovokasi > 80 **and** inputprovokasi <= 100):  
 provokasi.append(["Sangat Tinggi", trapesium(inputprovokasi, 80, 90, 100, 101)])

fungsi baris kode diatas adalah untuk menentukan membership function Provokasi, dengan Inputan seperti fungsi keanggotaan dibawah ini, yang sudah didefinisikkan dari soal



### Inference

#### Fuzzy Rules

**def fuzzyrules**(emo, pro):  
 emosi = emo[0]  
 provo = pro[0]  
 # Kondisi Sangat Labil  
  
 **if** (emosi == "Sangat Labil" **and** provo == "Sangat Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Sangat Labil" **and** provo == "Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Sangat Labil" **and** provo == "Normal"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Sangat Labil" **and** provo == "Tinggi"):  
 **return** ["Iya", min(emo[1], pro[1])]  
 **if** (emosi == "Sangat Labil" **and** provo == "Sangat Tinggi"):  
 **return** ["Iya", min(emo[1], pro[1])]  
  
 # Kondisi Labil  
 **if** (emosi == "Labil" **and** provo == "Sangat Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Labil" **and** provo == "Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Labil" **and** provo == "Normal"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Labil" **and** provo == "Tinggi"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Labil" **and** provo == "Sangat Tinggi"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
  
 # Kondisi Stabil  
 **if** (emosi == "Stabil" **and** provo == "Sangat Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Stabil" **and** provo == "Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Stabil" **and** provo == "Normal"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Stabil" **and** provo == "Tinggi"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Stabil" **and** provo == "Sangat Tinggi"):  
 **return** ["Iya", min(emo[1], pro[1])]  
  
 # Kondisi Sangat Stabil  
 **if** (emosi == "Sangat Stabil" **and** provo == "Sangat Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Sangat Stabil" **and** provo == "Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Sangat Stabil" **and** provo == "Normal"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Sangat Stabil" **and** provo == "Tinggi"):  
 **return** ["Iya", min(emo[1], pro[1])]  
 **if** (emosi == "Sangat Stabil" **and** provo == "Sangat Tinggi"):  
 **return** ["Iya", min(emo[1], pro[1])]  
  
 # Kondisi Abnormal  
 **if** (emosi == "Abnormal" **and** provo == "Sangat Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Abnormal" **and** provo == "Rendah"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Abnormal" **and** provo == "Normal"):  
 **return** ["Tidak", min(emo[1], pro[1])]  
 **if** (emosi == "Abnormal" **and** provo == "Tinggi"):  
 **return** ["Iya", min(emo[1], pro[1])]  
 **if** (emosi == "Abnormal" **and** provo == "Sangat Tinggi"):  
 **return** ["Iya", min(emo[1], pro[1])]

Baris kode program diatas adalah fungsi menentukkan aturan fuzzy untuk masalah pendefinisian berita hoax atau tidak, dari dataset yang sudah diberikkan dari B01 sampai B020, sengan aturan fuzzy yang sudah dibuat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Abnormal | Tidak | Tidak | Tidak | Iya | Iya |
| Sangat Stabil | Tidak | Tidak | Tidak | Iya | Iya |
| Stabil | Tidak | Tidak | Tidak | Tidak | Iya |
| Labil | Tidak | Tidak | Tidak | Tidak | Tidak |
| Sangat Labil | Tidak | Tidak | Tidak | Iya | Iya |
|  | Sangat Rendah | Rendah | Normal | Tinggi | Sangat Tinggi |

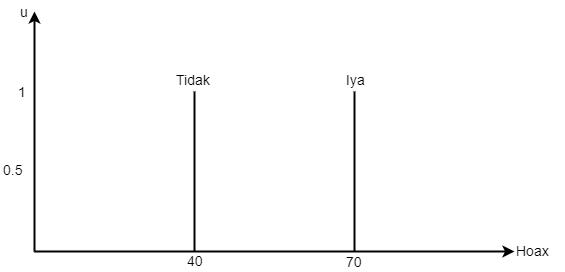
**def inference**():  
 hasil = [0, 0]  
 **for** i **in** range(len(emosional)):  
 **for** j **in** range(len(provokasi)):  
 p = fuzzyrules(emosional[i], provokasi[j])  
 **if** (p[0] == "Tidak" **and** hasil[0] < p[1]):  
 hasil[0] = p[1]  
 **elif** (p[0] == "Iya" **and** hasil[1] < p[1]):  
 hasil[1] = p[1]  
 **return** hasil

Fungsi baris kode program diatas alah untuk memperhitungkan aturan yang ada dalam basis pengetahuan. Dengan menggunakan aturan Disjunction(v) dengan memilih derajat keanggotaan maxium dari nilai-nilai linguistik yang dihubungkan oleh v. untuk melakukkan proses composition, yaitu agregasi hasil Clipping dari semua aturan fuzzy sehingga kita dapatkan satu fuzzy set tunggal.

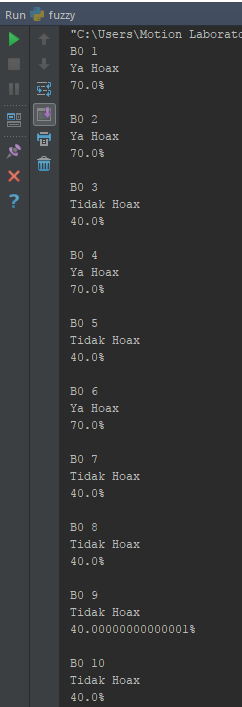
### Defuzzyfication

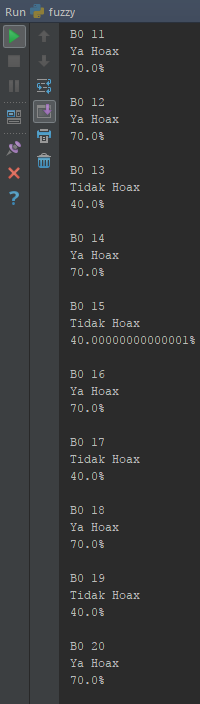
Prose defuzzyfication pada program ini adalah menggunakan model sugeno

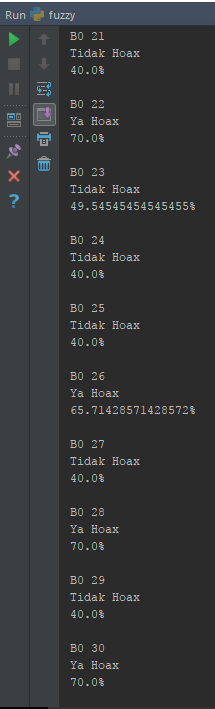
**def defuzzy**(inferens):  
 bobot = [40, 70]  
 **return** float(((inferens[0] \* bobot[0]) + (inferens[1] \* bobot[1])) / (inferens[0] + inferens[1]))



### Hasil







# Bab III Kesimpulan

Dari program fuzzy yang telah dibuat berdasarkan soal yang telah diberikan, Dengan mengunakan dua puluh berita pertama, B01 sampai B20, sebagai acuan untuk membangun sebuah sistem penalaran berbasis *fuzzy logic* untuk mendeteksi apakah sepuluh berita yang belum diketahui kelasnya, B21 sampai B30 (Data Testing), adalah berita bohong (*hoax*) atau bukan. Saya memperoleh hasil kategori berita B21-B30 sebagai berikut

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Berita | Emosi | Provokasi | Hoax | Definisi Program |
| B01 | 97 | 74 | Ya | Ya Hoax |
| B02 | 36 | 85 | Ya | Ya Hoax |
| B03 | 63 | 43 | Tidak | Tidak Hoax |
| B04 | 82 | 90 | Ya | Ya Hoax |
| B05 | 71 | 25 | Tidak | Tidak Hoax |
| B06 | 79 | 81 | Ya | Ya Hoax |
| B07 | 55 | 62 | Tidak | Tidak Hoax |
| B08 | 57 | 45 | Tidak | Tidak Hoax |
| B09 | 40 | 65 | Tidak | Tidak Hoax |
| B10 | 57 | 45 | Tidak | Tidak Hoax |
| B11 | 77 | 70 | Ya | Ya Hoax |
| B12 | 68 | 75 | Ya | Ya Hoax |
| B13 | 60 | 70 | Tidak | Tidak Hoax |
| B14 | 82 | 90 | Ya | Ya Hoax |
| B15 | 40 | 85 | Tidak | Tidak Hoax |
| B16 | 80 | 68 | Ya | Ya Hoax |
| B17 | 60 | 72 | Tidak | Tidak Hoax |
| B18 | 50 | 95 | Ya | Ya Hoax |
| B19 | 100 | 18 | Tidak | Tidak Hoax |
| B20 | 11 | 99 | Ya | Ya Hoax |
| B21 | 58 | 63 | Tidak | Tidak Hoax |
| B22 | 68 | 70 | Ya | Ya Hoax |
| B23 | 64 | 66 | Tidak | Tidak Hoax |
| B24 | 57 | 77 | Tidak | Tidak Hoax |
| B25 | 77 | 55 | Tidak | Tidak Hoax |
| B26 | 98 | 64 | Ya | Ya Hoax |
| B27 | 91 | 59 | Tidak | Tidak Hoax |
| B28 | 50 | 95 | Ya | Ya Hoax |
| B29 | 95 | 55 | Tidak | Tidak Hoax |
| B30 | 27 | 79 | Ya | Ya Hoax |

# Daftar Pustaka

**Suyanto (2014).** Artificial Intelligence Searching-Reasoning-Planning-Learning.Informatika, Bandung.

Kusumadewi. S dan H. Purnomo. (2004). Aplikasi Logika Fuzzy Untuk Mendukung Keputusan. Graha Ilmu, Yogyakarta.

Marimin. (2005). Teori dan aplikasi sistem pakar dalam tehnologi manajerial. IPB – Press, Bogor.

Sri Kusumadewi, (2002). Analisis  dan Desain Sistem Fuzzy menggunakan Tool Box Matlab, edisi pertama. Penerbit Graha Ilmu, Jakarta.

Sri Kusumadewi, (2003). Artificial Intelligence (Teknik dan Aplikasinya), edisi pertama. Penerbit Graha Ilmu, Jakarta.

Turban, E. (1988). Decision Support and Expert System. MacMillan Publishing Company, New York.

Widodo,T.S (2005). Sistem Neuro Fuzzy. Graha Ilmu, Yogyakarta.