

# Trading Decision dengan Fuzzy Logic

Membangun model sederhana  
dengan Python

cr. nbilaasvv

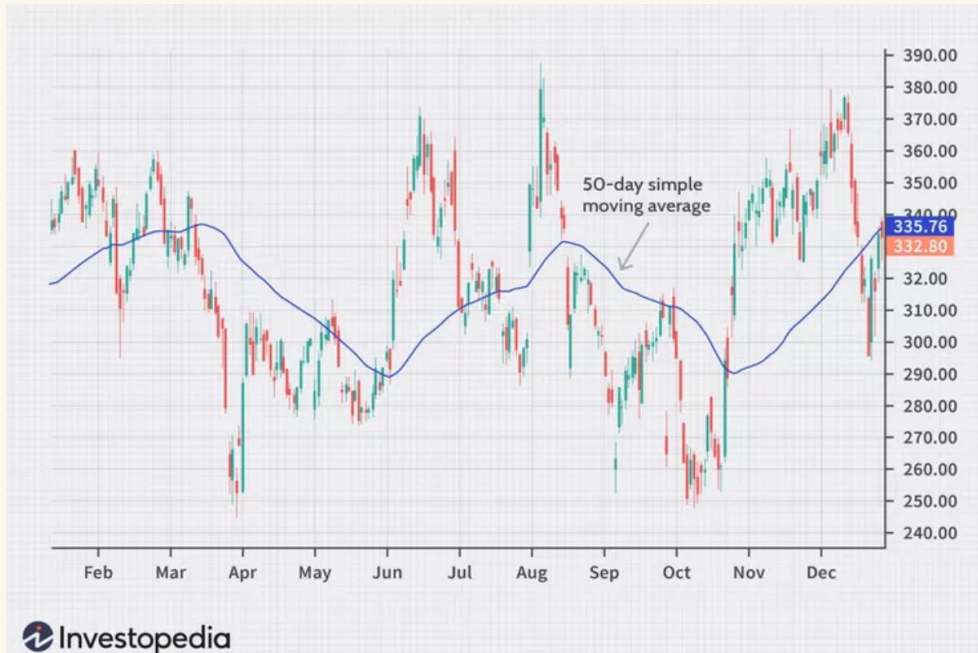


# Trading

<b>Trading</b>	Trading saham adalah kegiatan membeli dan menjual saham dalam jangka waktu yang relatif pendek, seperti dalam hitungan hari, jam, bahkan menit
<b>Trader Saham</b>	Trader saham merupakan pihak yang melakukan trading saham
<b>Saham</b>	Saham merupakan surat berharga yang menyatakan bukti kepemilikan sebuah perusahaan atau badan usaha, sehingga pihak tersebut memiliki klaim atas pendapatan perusahaan, klaim atas aset perusahaan, dan berhak hadir dalam Rapat Umum Pemegang Saham (RUPS)
<b>Moving Average (MA)</b>	indikator yang mengukur harga rata-rata suatu saham selama periode waktu tertentu
<b>Relative Strength Index (RSI)</b>	indikator momentum yang mengukur kekuatan dan kelemahan relatif dari pergerakan harga saham
<b>Trading Strength</b>	kekuatan umum tren saham

# Moving Average

- Moving Average adalah indikator yang mengukur harga rata-rata suatu saham selama periode waktu tertentu.
- Satuan (Rupiah/Dolar) dan berbentuk index.



Sumber : <https://tinyurl.com/mcjrcpjd>

# Relative Strength Index

- RSI adalah indikator momentum yang mengukur kekuatan dan kelemahan relatif dari pergerakan harga saham.
- Tidak memiliki satuan dan berbentuk index.



Sumber : <https://tinyurl.com/52y3a4sr>

# Mendefinisikan RULES

Di definisikan dua input berupa **Moving Average (MA)** dan **Relative Strength Index (RSI)** dan output **Trading Signal (TS)**



dengan rules sebagai berikut:

1. Jika **Moving Average (MA)** rendah dan **Relative Strength Index (RSI)** juga rendah, maka keputusannya adalah **Sell (Jual)** Saham
2. Jika **Moving Average (MA)** tinggi dan **Relative Strength Index (RSI)** juga tinggi, maka keputusannya adalah **Buy (Beli)** Saham



## Tujuan

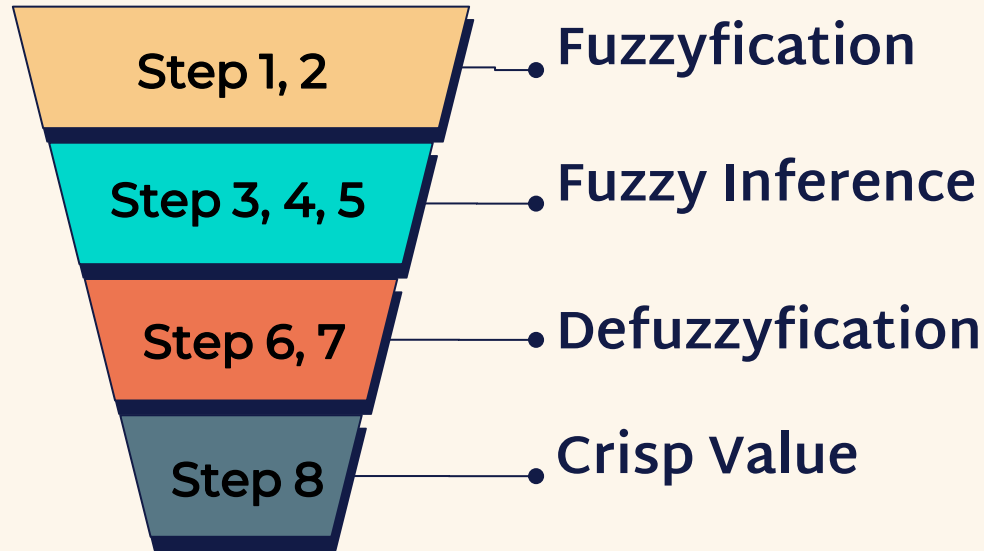
Menganalisis ribuan sekuritas pada sistem trading secara real time dan memberikan informasi berupa peluang trading terbaik kepada investor



## Manfaat

Mengurangi ruang lingkup keputusan untuk trading

# Fuzzy Logic dengan Metode Mamdani



# Langkah Model Fuzzy





# Fuzzyfication

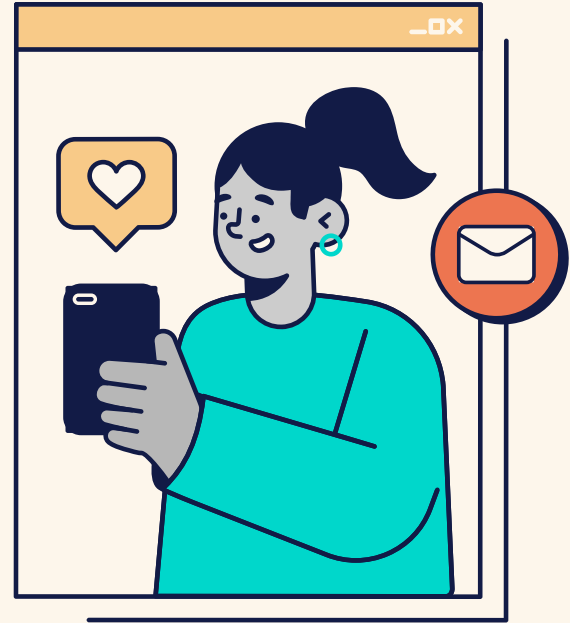


# Input Values

MISALKAN:

```
1  #input values
2  MA = 40
3  RSI = 80
4
5  print("Moving Average :", MA)
6  print("Relative Strength Index:", RSI)
```

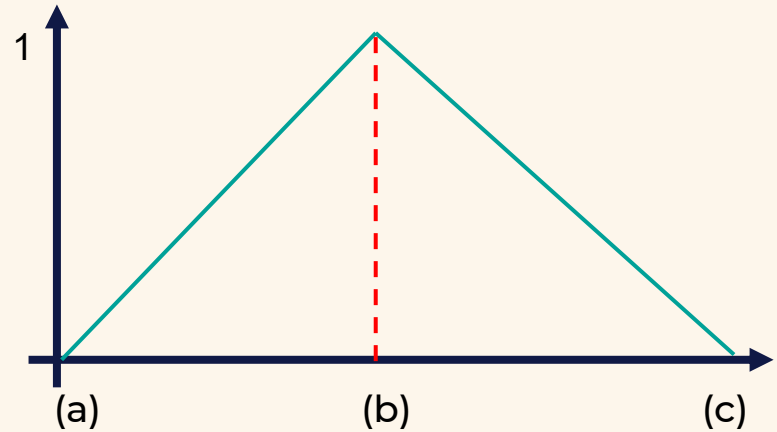
```
Moving Average : 40
Relative Strength Index: 80
```



# Mendefinisikan Membership Function

## Triangle

```
1  #function untuk fuzzyfication
2  def triangular(x, a, b, c):
3      if x <= a or x >= c:
4          return 0
5      elif a < x <= b:
6          return (x - a) / (b - a)
7      elif b < x < c:
8          return (c - x) / (c - b)
9      else:
10         return 0
```

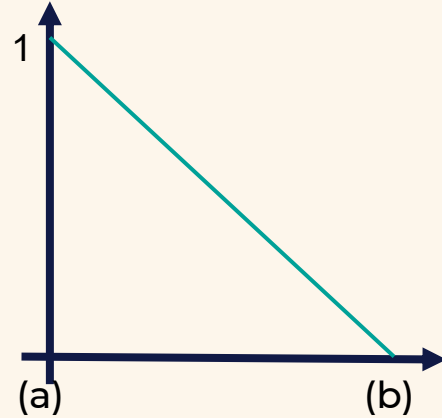


$$f(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

# Mendefinisikan Membership Function

## Linear Decrease

```
#function untuk fuzzyfication
def linearDecreasing(x,alpha, beta):
    if x<alpha:
        return 1
    if alpha<x and x<=beta:
        return (beta - x)/(beta - alpha)
    else:
        return 0
```



$$f(x; a, b) = \frac{(b - x)}{(b - a)}$$

# Fuzzy Linguistik Variabel

Low : Index Rendah

High : Index Tinggi

Fuzzy Rules:

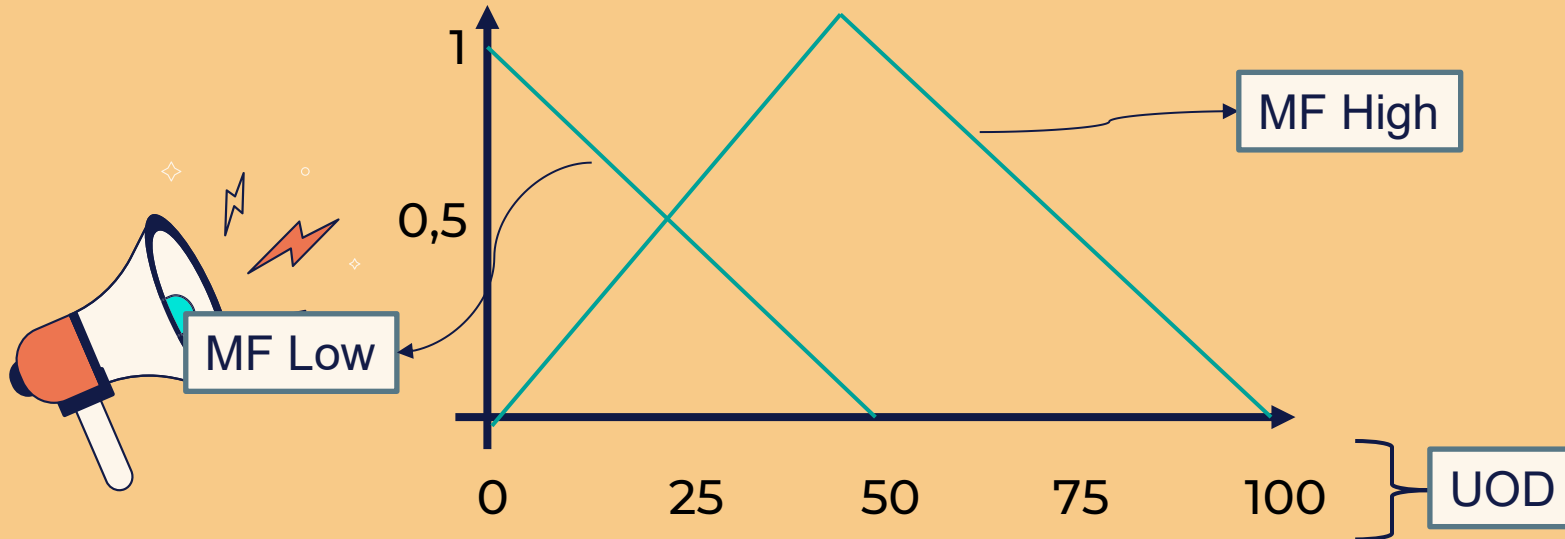
R1	If MA is Low	And	If RSI is Low	Then	TS is SS
R2	If MA is High	And	If RSI is High	Then	TS is SB

# Partition

Partitions:

- \* low =  $[0,0,50]$
- \* high =  $[0,50,100]$

UOD : Universe of Discourse  $[0,100]$



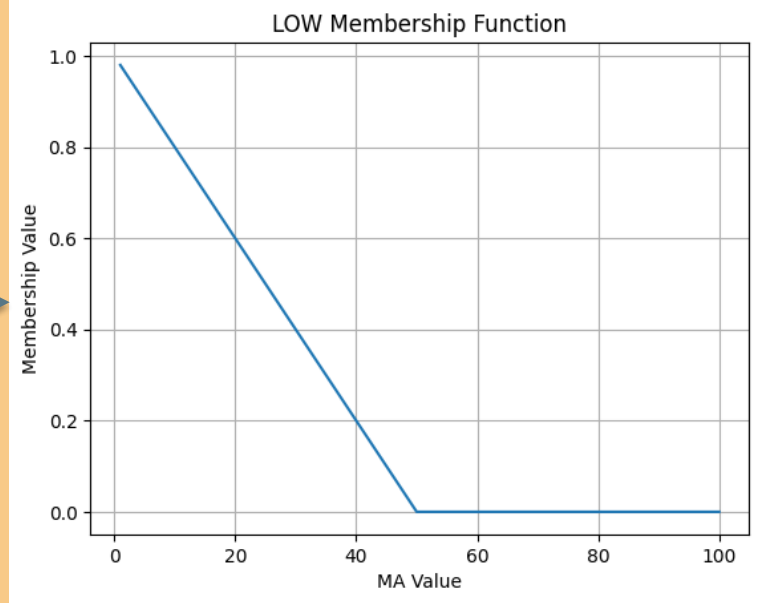
## Melakukan Partition untuk setiap MF per Variabel

```
[ ] 1 #fuzzy values untuk semua fuzzy sets  
    2 ma_low, ma_high = partition(MA)  
    3 rsi_low, rsi_high = partition(RSI)
```



# Low Membership Function

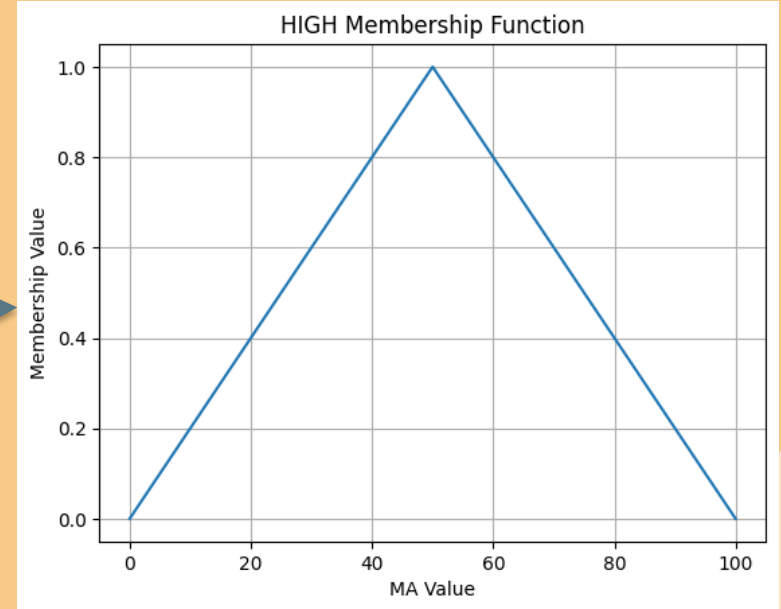
```
1 #fuzzy partition
2 def partition(x):
3     low = 0; high = 0
4
5     if x>0 and x<50:
6         low = triangular(x,0,0,50)
7
8     if x>0 and x<100:
9         high = triangular(x,0,50,100)
10
11 return low, high
```



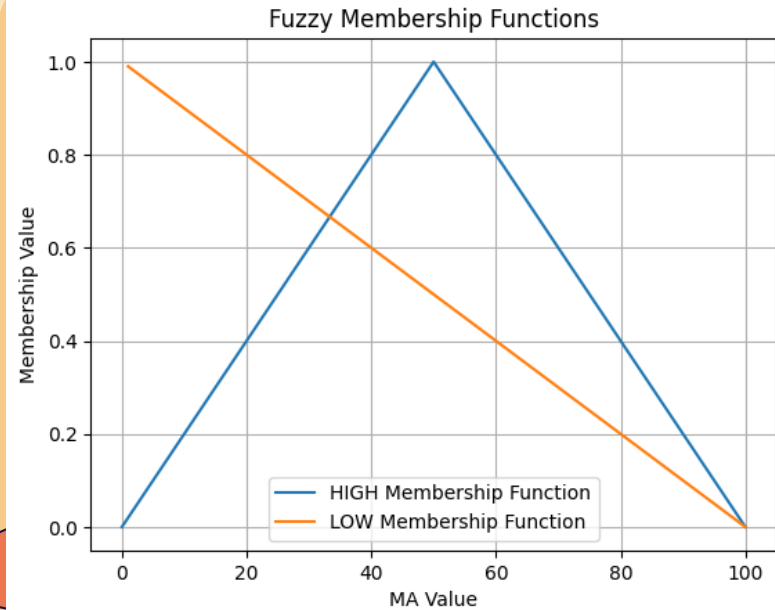


# High Membership Function

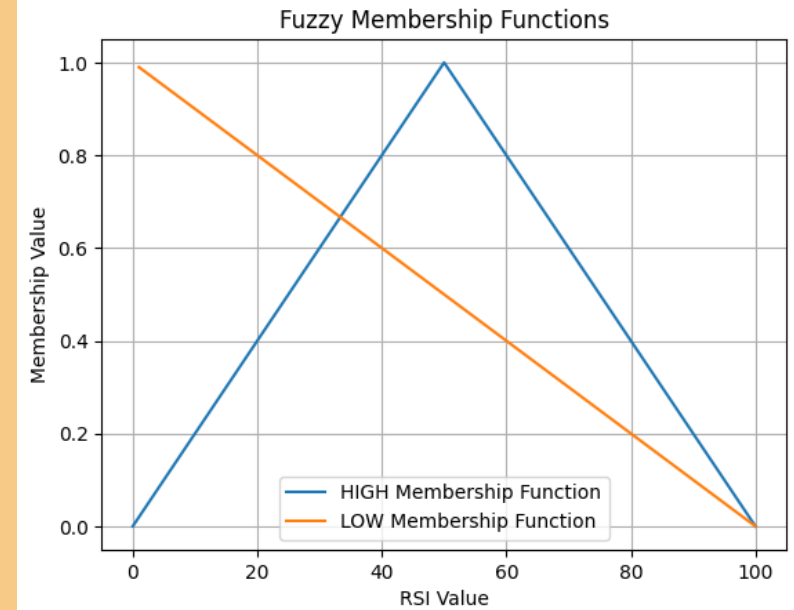
```
1 #fuzzy partition
2 def partition(x):
3     low = 0; high = 0
4
5     if x>0 and x<50:
6         low = triangular(x,0,0,50)
7
8     if x>0 and x<100:
9         high = triangular(x,0,50,100)
10
11 return low, high
```



## MF untuk Variabel MA

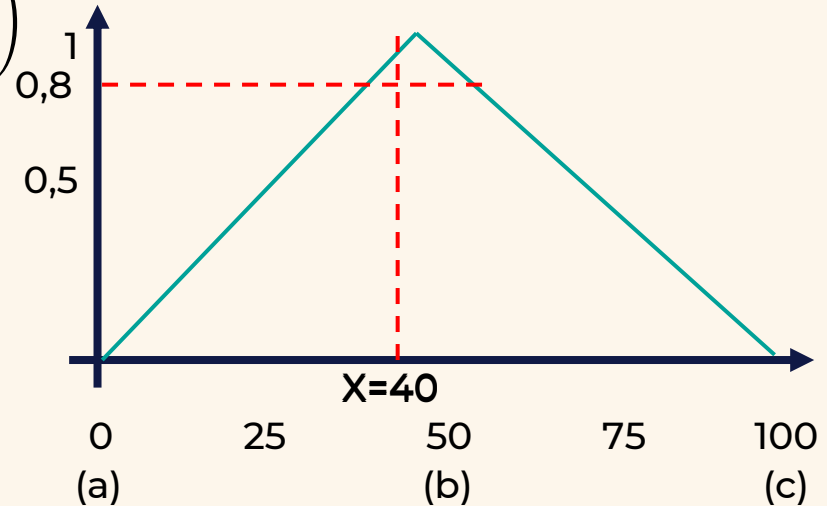


## MF untuk Variabel RSI



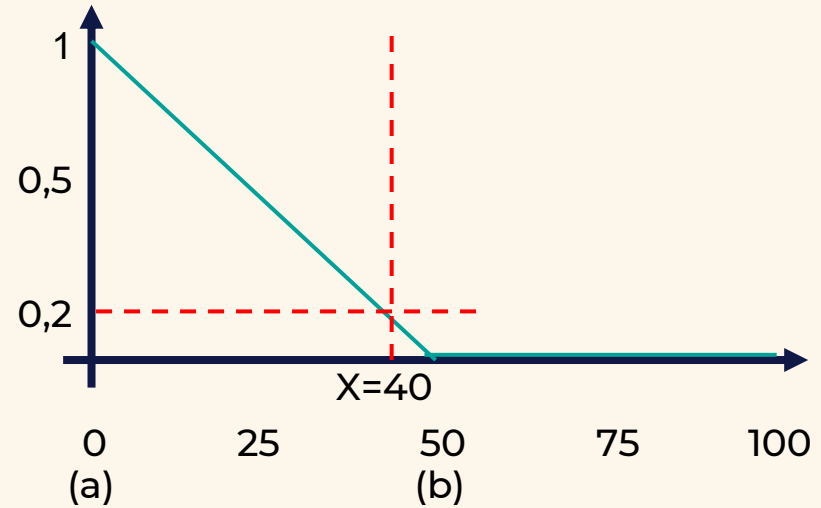
## Menghitung Membership Function MA High

$$\begin{aligned} f(40; 0, 50, 100) &= \max \left( \min \left( \frac{40 - 0}{50 - 0}, \frac{100 - 40}{100 - 50} \right), 0 \right) \\ &= \max \left( \min \left( \frac{40}{50}, \frac{60}{50} \right), 0 \right) \\ &= \max \left( \frac{40}{50}, 0 \right) \\ &= \frac{40}{50} \\ &= 0,8 \end{aligned}$$



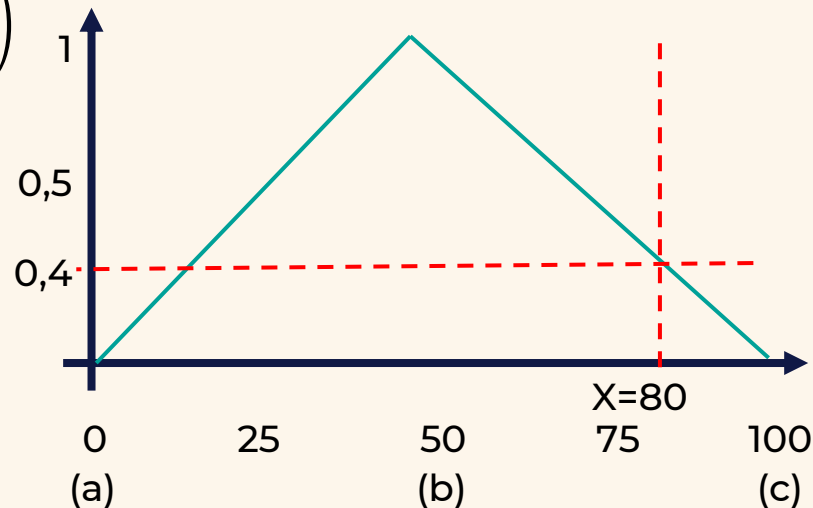
## Menghitung Membership Function MA Low

$$\begin{aligned} f(40; 0, 50) &= \frac{(50 - 40)}{(50 - 0)} \\ &= \frac{10}{50} \\ &= 0,2 \end{aligned}$$



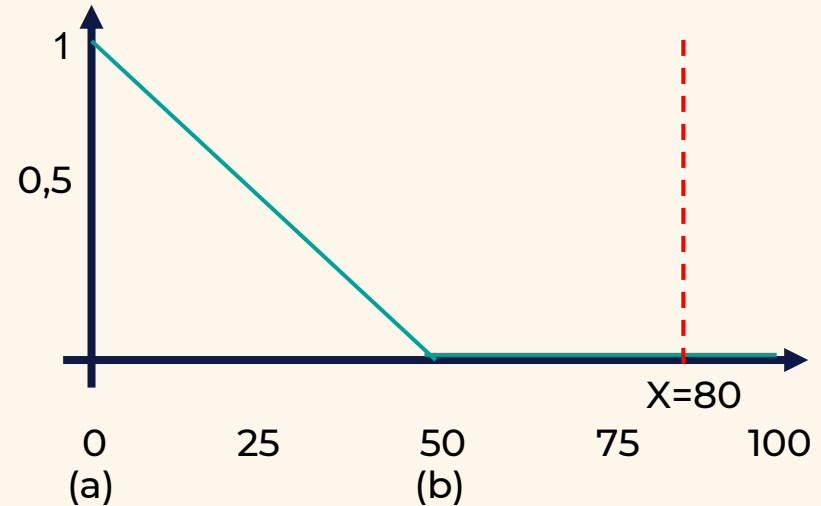
## Menghitung Membership Function RSI High

$$\begin{aligned} f(80; 0, 50, 100) &= \max \left( \min \left( \frac{80 - 0}{50 - 0}, \frac{100 - 80}{100 - 50} \right), 0 \right) \\ &= \max \left( \min \left( \frac{80}{50}, \frac{20}{50} \right), 0 \right) \\ &= \max \left( \frac{20}{50}, 0 \right) \\ &= \frac{20}{50} \\ &= 0,4 \end{aligned}$$



## Menghitung Membership Function RSI Low

X= 80 tidak memiliki keanggotaan pada MF RSI low



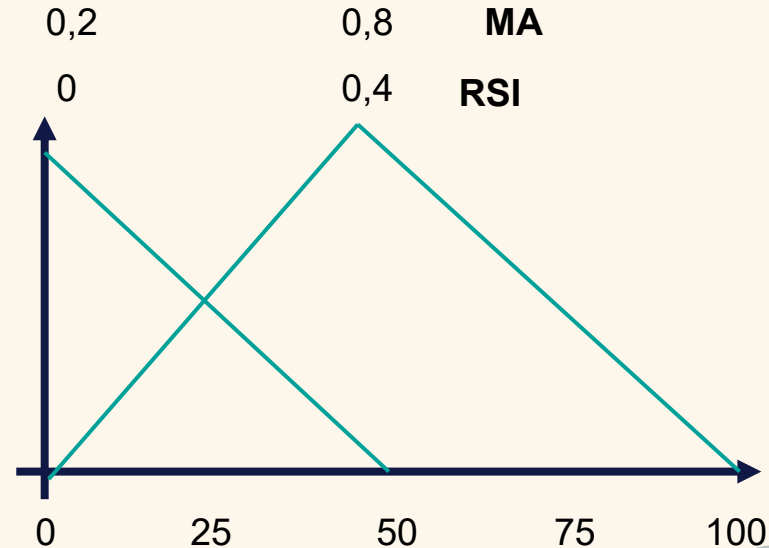
# Menunjukkan nilai Crisp Sets

```
[ ] 1 import numpy as np

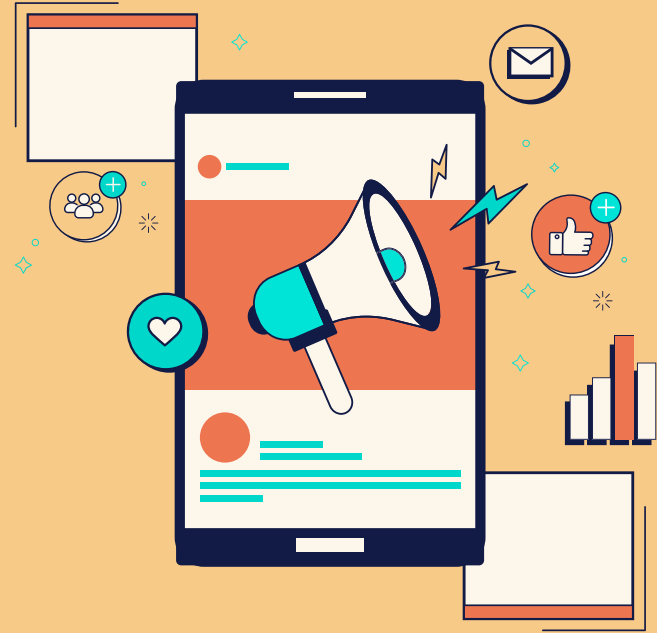
[ ] 1 #menunjukkan fuzzy values untuk semua fuzzy sets
2 output = [[ma_low, ma_high]
3           |[rsi_low, rsi_high]]
4
5 print("Fuzzy values dari crisp inputs")
6 print('MA', 'RSI')
7 print(np.round(output,2))
```

```
Fuzzy values dari crisp inputs
MA RSI
[[0.2 0.8]
 [0.  0.4]]
```

Sehingga :



# Fuzzy Inference





# Penerapan Rules pada Fuzzy Sets

R1	If MA is Low (0, 2)	And	If RSI is Low (0)	Then	TS is SS (0)
R2	If MA is High (0,8)	And	If RSI is High (0,4)	Then	TS is SB (0,4)

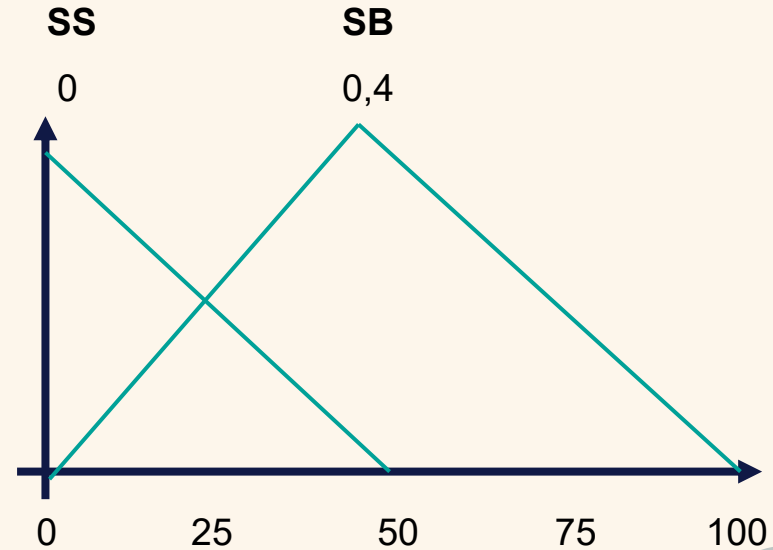
Dipilih minimum  
dari setiap  
membership  
functions

```
[ ] 1 #penerapan rules pada fuzzy sets
2 def rule(ma_low, ma_high
3         ,rsi_low, rsi_high):
4     SS = min(ma_low, rsi_low)
5     SB = min(ma_high, rsi_high)
6     return SS, SB
7
8 SS, SB = rule(ma_low, ma_high, rsi_low, rsi_high)
```

# Menunjukkan output fuzzy values ✨

```
1 #Menunjukkan fuzzy values untuk semua rules
2 output_rules = [SS, SB]
3 print("Fuzzy output:")
4 print('SS', 'SB')
5 print(np.round(output_rules,2))
```

```
Fuzzy output:
SS SB
[0.  0.4]
```



A man in a blue suit and a woman in a white shirt are working together at a desk. The man is seated, looking at a laptop screen, while the woman stands behind him, pointing at a document. The desk is cluttered with papers, a calculator, and a small potted plant. A large window in the background shows a cityscape. The text 'De-Fuzzyfication' is overlaid in a yellow rounded rectangle.

# De-Fuzzyfication

# Proses De-Fuzzyfikasi



Menentukan  
Membership  
function

De-Fuzzyfication



Menghitung  
area dibawah  
kurva fuzzy



Menghitung  
Titik Pusat

# De-Fuzzyfikasi SS

Definisi de-fuzzification untuk SS dan SB

Definisikan bahwa setiap area dan titik pusat = 0

Menentukan parameter MF

Menghitung area dibawah kurva

Menghitung pusat

```
#De-fuzzifications
def defuzzification(SS, SB):
    # area = daerah segitiga
    # c = pusat
    areaSS = 0; areaSB = 0;
    cSS = 0; cSB = 0

    if SS != 0:
        # parameter Fuzzy membership function
        a = 0
        b = 0
        c = 50

        # menghitung area triangle membership function SS
        x1 = (SS * (b - a) + a)
        x2 = (c - SS * (c - b))
        d1 = (c - a)
        d2 = x2 - x1
        areaSS = (1 / 2) * SS * (d1 + d2)
        cSS = (1 / 3) * (a + x1 + x2)
```

# De-Fuzzyfikasi SB

Menentukan parameter MF

Menghitung area dibawah kurva

Menghitung pusat

```
if SB != 0:
    # parameter Fuzzy membership function
    a = 0
    b = 50
    c = 100

    # menghitung area triangle membership function SB
    x1 = (SB * (b - a) + a)
    x2 = (c - SB * (c - b))
    d1 = (c - a)
    d2 = x2 - x1
    areaSB = (1 / 2) * SB * (d1 + d2)
    cSB = (1 / 3) * (a + x1 + x2)
```

# Menghitung Degree of Membership

Menghitung Momen

```
1 # Menghitung derajat keanggotaan untuk setiap himpunan fuzzy putput
2 #N =  $\sum (\text{Degree of Membership} * \text{Crisp Value})$ 
3 numerator = areaSS*cSS + areaSB*cSB
4
5 #D =  $\sum (\text{Degree of Membership})$ 
6 denominator = areaSS + areaSB
```

Menghitung Luas

# Menghitung Crisp Value

```
[ ] 1  # Menghitung defuzzifikasi menggunakan metode rata rata tertimbang
    2  #menentukan crisp value
    3  if denominator !=0:
    4      #Crisp Value = N / D
    5      crisp_output = numerator / denominator
    6  else:
    7      crisp_output = 0 # definisikan value ketika tidak ada area
    8      print("tidak ada hasil")
    9
   10  return crisp_output
```

$$Z^* = \frac{\int \mu(z)z \, dz}{\int \mu(z)}$$

$$\text{Crisp Value} = \frac{\text{Momen}}{\text{Luas}}$$



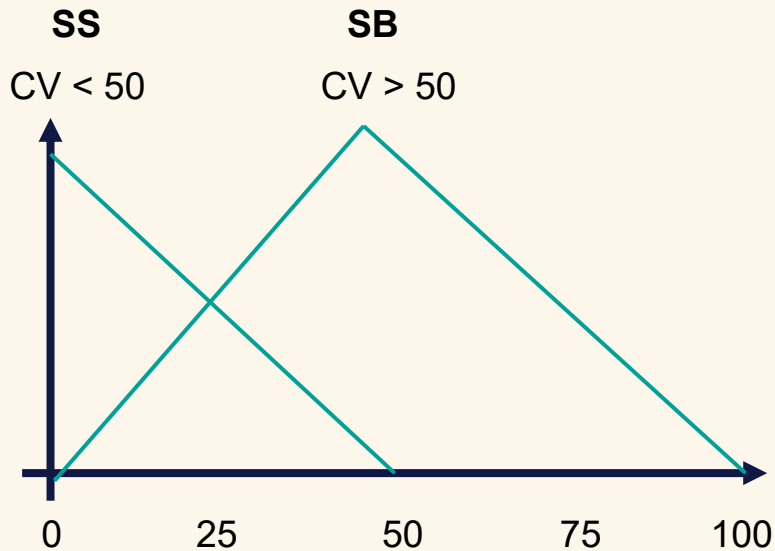
# Menghitung Hasil De-Fuzzyfikasi

```
54  
55 # nilai crisp final (hasil defuzzifikasi)  
56 crisp_value = defuzzification(SS, SB)  
57 print("Crisp Value:", crisp_value)
```

```
Crisp Value: 33.33333333333333
```

Didapatkan Crisp Value sebesar 33,3333

# Membuat keputusan dengan Crisp Value




```
def rules(crisp_value):  
    if crisp_value < 25:  
        return "SS" # Sell Signal  
    elif 25 <= crisp_value < 50:  
        return "SS" # Sell Signal  
    elif 50 <= crisp_value < 75:  
        return "SB" # Buy Signal  
    else:  
        return "SB" # Buy Signal
```

Crisp Value sebesar 33,3333 < 50, sehingga keputusannya adalah SS (Sell Signal)

# De-fuzzyfication -> Rules

```
# terapkan defuzzification function pada rules  
crisp_value = defuzzification(SS, SB)
```



The diagram illustrates the process of defuzzification. It shows two lines of code. The first line, `crisp_value = defuzzification(SS, SB)`, is highlighted with a brown border. An orange arrow points from the `defuzzification` function to a text box on the right. A blue arrow points from the `crisp_value` variable to the `rules` function in the second line of code, `trading_signal = rules(crisp_value)`, which is also highlighted with a brown border.

Mendefinisikan defuzzifikasi sebagai crisp value

```
trading_signal = rules(crisp_value)
```



# Keputusan Trading

```
15  
16 # Menunjukkan keputusan Trading Signal (output)  
17 print("Trading Signal:", trading_signal)  
18
```

```
Trading Signal: SS
```

Keputusan dari defuzzifikasi menunjukkan  
untuk SS (Signal Sell)

# Kesimpulan

Berdasarkan algoritma fuzzy logic dengan input Moving Average = 40 dan Relative Strength Index = 80, didapatkan nilai tegas ("crisp value") sebesar 33,33 sehingga didapatkan keputusan Sell Signal (SS) yang artinya pengguna trading saham disarankan untuk menjual sahamnya.



# Resources

Perhitungan dari Model Fuzzy dengan Metode Tsukamoto ini menggunakan alat Python yang dilakukan dengan menggunakan Google Collab. Code dari perhitungan ini dapat diakses pada link berikut :

<https://colab.research.google.com/drive/1PAJAZdCX1qieLV5c2n4RcOkMrnduibdO?usp=sharing>