

Laporan Tugas 2

CCH3F3 Kecerdasan Buatan

I. Deskripsi Masalah

Terdapat data penghasilan/bulan dan hutang 100 kepala keluarga. Diharuskan mencari 20 kepala keluarga paling optimum untuk mendapatkan Bantuan Langsung Tunai (BLT). Pencarian dilakukan berdasarkan data penghasilan dan hutang yang dihitung menggunakan algoritma *Fuzzy Logic* untuk memilih 20 orang dengan nilai potensi paling besar.

II. Strategi Penyelesaian Masalah

a. Pemilihan Bahasa Pemrograman

Bahasa pemrograman yang digunakan Python, karena memiliki *library* yang cukup luas dan implementasi *import/export* file yang cenderung lebih mudah dibanding Bahasa pemrograman lain (yang saya ketahui).

b. Parameter yang Digunakan

Linguistic Variable yang dibuat

Input :

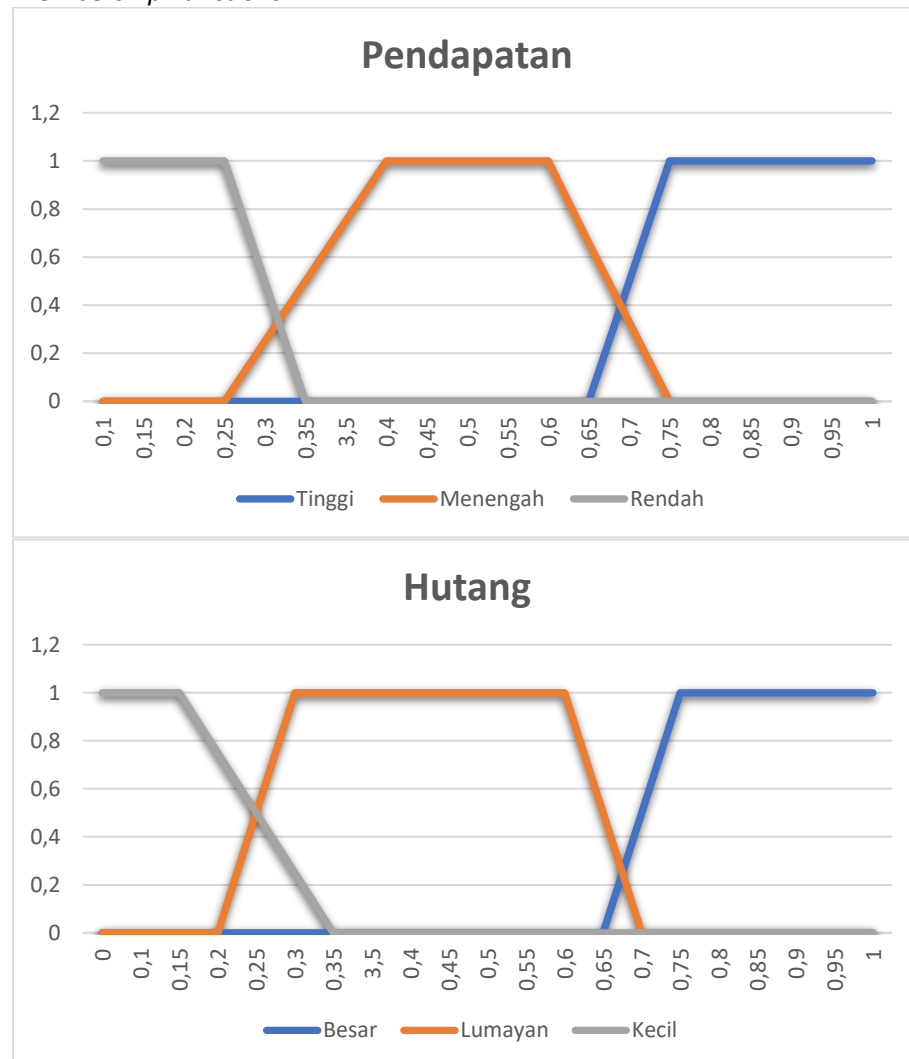
Pendapatan : Tinggi, Menengah, Rendah

Hutang : Besar, Lumayan, Kecil

Output :

Potensi : Ya, Mungkin, Tidak

Membership Functions



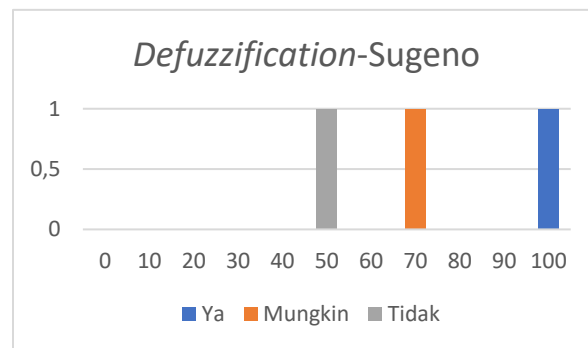
Range sumbu Y (0-1) menandakan poin *membership* dari setiap titik. Poin nol menandakan titik tersebut tidak memiliki poin pada sumbu yang ditentukan. Poin satu menandakan kepemilikan penuh dari setiap titik. Sumbu X menandakan persentase terhadap pendapatan/hutang. Nol menandakan pendapatan/hutang terendah, dan satu menandakan pendapatan/hutang tertinggi. Dengan demikian, penentuan poin pendapatan/hutang dilakukan dengan rumus : $((x_{\max} - x_{\min}) * \text{persentase}) + x_{\min}$, untuk x = pendapatan/hutang.

Fuzzy Rules

Pendapatan	Hutang	Potensi
Tinggi	Besar	Mungkin
Tinggi	Lumayan	Tidak
Tinggi	Sedikit	Tidak
Menengah	Besar	Ya
Menengah	Lumayan	Mungkin
Menengah	Sedikit	Tidak
Rendah	Besar	Ya
Rendah	Lumayan	Ya
Rendah	Sedikit	Mungkin

c. Defuzzification Method

Metode *defuzzification* yang digunakan Takagi-Sugeno-Style. Pemilihan metode tersebut untuk meminimalisir kesalahan pembuatan algoritma, yang dapat menyebabkan hasil bias.



III. Hasil Terbaik Program

20 Kepala Keluarga yang Mendapatkan BLT :

```
D:\Kuliah\Semester 5\AI\Tugas\Tugas 2\Kirim>python program.py
[93, 100.0]
[77, 100.0]
[74, 100.0]
[62, 100.0]
[61, 100.0]
[59, 100.0]
[53, 100.0]
[49, 100.0]
[36, 100.0]
[34, 100.0]
[23, 100.0]
[21, 100.0]
[15, 100.0]
[9, 100.0]
[47, 99.99999999999999]
[3, 99.99999999999999]
[41, 98.59922178988329]
[44, 98.32642836496774]
[88, 97.04268677703242]
[64, 96.57075285770594]
20
```