**Algoritma FCM**

*Fuzzy C-Means Clustering (FCM)*, atau dikenal juga sebagai *fuzzy* Isodata merupakan salah satu metode clustering yang merupakan bagian dari metode *Hard K-Means*.FCM menggunakan model pengelompokan *fuzzy* sehingga data dapat menjadi anggota dari semua kelas atau cluster terbentuk dengan derajat atau tingkat keanggotaan yang berbeda antara 0 hingga 1. Tingkat keberadaan data dalam suatu kelas atau cluster ditentukan oleh derajat keanggotaannya.

Algoritma *Fuzzy C-Means* (FCM) adalah sebagai berikut:

1. Input data yang akan di cluster X, berupa matriks berukuran n x m (n = jumlah sampel data, m = atribut setiap data). Xij = data sampel ke-i (i=1,2,...,n), atribut ke-j (j=1,2,...,m).
2. Tentukan:

* Jumlah cluster = c;
* Pangkat = w;
* Maksimum iterasi = MaxIter;
* Error terkecil yang diharapkan =
* Fungsi objektif awal = Po = 0;
* Iterasi awal = t = 1;

1. Bangkitkan bilangan random dengan menggunakan kodingan:

|  |
| --- |
| def generate\_initial\_partition\_matrix(n, c):      partition\_matrix = []      for i in range(n):          row = [random.random() for \_ in range(c)]          # Normalisasi setiap baris sehingga jumlah probabilitas dalam setiap baris menjadi 1          row\_sum = sum(row)          row = [value / row\_sum for value in row]          partition\_matrix.append(row)      return partition\_matrix |

1. Hitung pusat cluster ke-k: Vkj, dengan k = 1,2,...,c; dan j = 1,2,...,m (Yan, 1994).



1. Hitung fungsi objektif pada iterasi ke-t, Pt (Yan, 1994):



1. Hitung perubahan matriks partisi menggunakan kodingan:

def generate\_initial\_partition\_matrix(n, c):

    partition\_matrix = []

    for i in range(n):

        row = [random.random() for \_ in range(c)

        # Normalisasi setiap baris sehingga jumlah probabilitas dalam setiap baris menjadi 1

        row\_sum = sum(row)

        row = [value / row\_sum for value in row]

        partition\_matrix.append(row)

    return partition\_matrix

1. Cek kondisi berhenti:

* Jika: ( |Pt - Pt-1| <      ) (Melebihi Batas Toleransi) atau (t > MaxIter) maka berhenti;
* Jika tidak: t = t+1, ulangi langkah ke-4.