LAPORAN

CLUSTERING DATASET *SEED*

DENGAN ALGORITMA K-MEANS

Oleh

Rahandi Noor Pasha	**05111640000054**
Dandy Naufaldi	**05111640000011**
Muhammad Alam Cahya N	**05111640000134**

KECERDASAN KOMPUTASIONAL F

Pembina

DR. CHASTINE FATICHAH, M.KOM

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018

DAFTAR ISI

METODE YANG DIGUNAKAN

K-means merupakan salah satu metode clustering non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih cluster. Metode ini mempartisi data ke dalam cluster sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokan ke dalam cluster yang lain.

Langkah-langkah dari K-means ialah sebagai berikut

1. Menentukan (K) secara random untuk centroid
2. Memasukkan setiap item pada centroid yang terdekat (menggunakan Euclidean Distance).
3. Pindahkan tiap centroid ke mean dari data yang termasuk clusternya.
4. Mengulangi step 2 dan 3 hingga konvergen.

Kelebihan dan kekurangan K-means algorithm

Kelebihan	**Kekurangan**
Simpel dan mudah dimengerti	Harus menentukan jumlah cluster sebelumnya.
Data secara otomatis masuk pada cluster	Semua data harus masuk pada cluster
	Sensitif terhadap data yang abnormal (*outlier*).

IMPLEMENTASI

1. Inisialisasi centroid
    - Secara random

```
numpy.random.seed(self.seed)
idx = numpy.random.choice(range(len(data)), size=(self.n_cluster))
centroid = data[idx]
```

      Mengambil K-data secara random untuk dijadikan centroid awal

    - Dengan K-Means++

```
numpy.random.seed(self.seed)
centroid = [int(numpy.random.uniform()*len(data))]
for _ in range(1, self.n_cluster):
    dist = []
    dist = [min([numpy.inner(data[c]-x, data[c]-x) for c in centroid])
            for i, x in enumerate(data)]
    dist = numpy.array(dist)
    dist = dist / dist.sum()
    cumdist = numpy.cumsum(dist)

    prob = numpy.random.rand()
    for i, c in enumerate(cumdist):
        if prob > c and i not in centroid:
            centroid.append(i)
            break
centroid = numpy.array([data[c] for c in centroid])
```

      Mengambil 1 index data secara random untuk 1 centroid pertama, lalu memilih index data lain dengan probabilitas sesuai jarak data dengan centroid terdekat yang data terhadap total jarak semua data dengan centroid terdekat masing-masing.

2. Menghitung jarak data ke centroid

```
distances = []
for c in self.centroid:
    distance = numpy.sum((data - c) * (data - c), axis=1)
    distances.append(distance)

distances = numpy.array(distances)
distances = distances.T
return distances
```

   Menghitung Euclidian distance (tanpa akar) dari semua data ke semua centroid

3. Memasukkan data ke dalam cluster

```
def _assign_cluster(self, distance: numpy.ndarray):
    """Assign cluster to data based on minimum distance to centroids

    Parameters
    ----------
    distance : numpy.ndarray
        Distance from each data to each centroid

    """
    cluster = numpy.argmin(distance, axis=1)
    return cluster
```

Memasukkan data ke dalam cluster berdasarkan jarak ke centroid yang terkecil

4. Update centroid

```
def _update_centroid(self, data: numpy.ndarray, cluster: numpy.ndarray):
    """Update centroid from means of each cluster's data

    Parameters
    ----------
    data : numpy.ndarray
        Data matrix to get mean from
    cluster : numpy.ndarray
        Cluster label for each data

    """
    centroids = []
    for i in range(self.n_cluster):
        idx = numpy.where(cluster == i)
        centroid = numpy.mean(data[idx], axis=0)
        centroids.append(centroid)
    centroids = numpy.array(centroids)
    return centroids
```

Mengupdate nilai centroid berdasarkan mean dari data yang masuk ke dalam cluster.

DATASET

A. Dataset Cluster

a. Sumber Dataset

https://archive.ics.uci.edu/ml/datasets/seeds

b. Tujuan Dataset

Mengkluster biji-bijian ke dalam tiga macam gandum yaitu Kama, Rosa, dan Canadian.

c. Jumlah Record Dataset

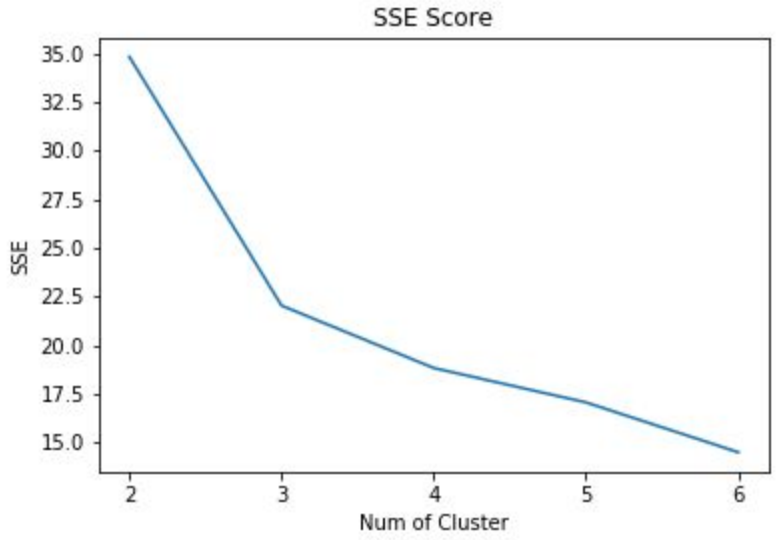
Terdapat 210 data dengan rincian 70 record setiap varietasnya

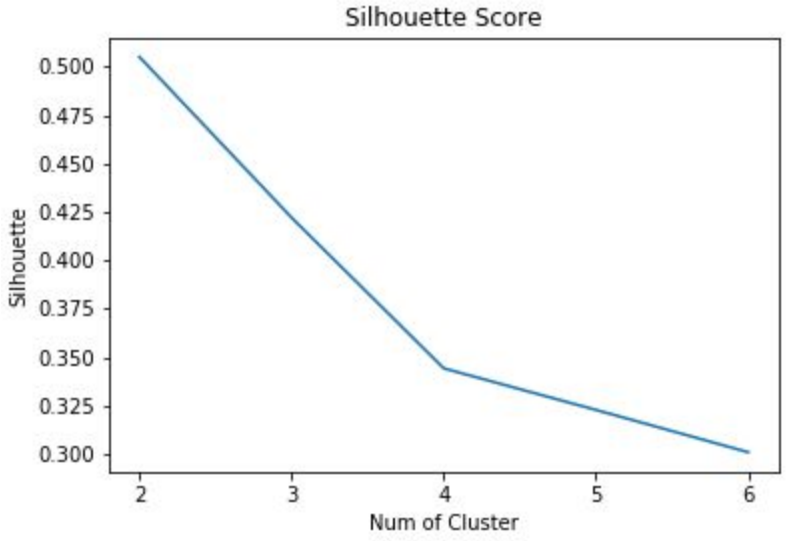
d. Atribut & Target Dataset

Nama	Tipe Data
Area A	continuous
Perimeter P	continuous
Compactness C	continuous
Length of kernel	continuous
Width of kernel	continuous
Asymmetry coefficient	continuous
length of kernel groove	continuous

UJI COBA

1. **Pengujian metode inisialisasi centroid**
2. **Pengujian untuk menentukan nilai - K yang baik**





Dengan mengutamakan skor SSE, maka besar K yang baik adalah 3 karena terbentuk '*elbow*' pada hasil plot

Nilai K	Silhouette	SSE
2	0.505114	34.813268

3	0.422105	22.024363
4	0.344288	18.816756
5	0.322823	17.049704
6	0.300981	14.487756

3. Pengujian akurasi hasil prediksi kelas untuk K = 3

Hasil Akurasi Terhadap Dataset "Seed"

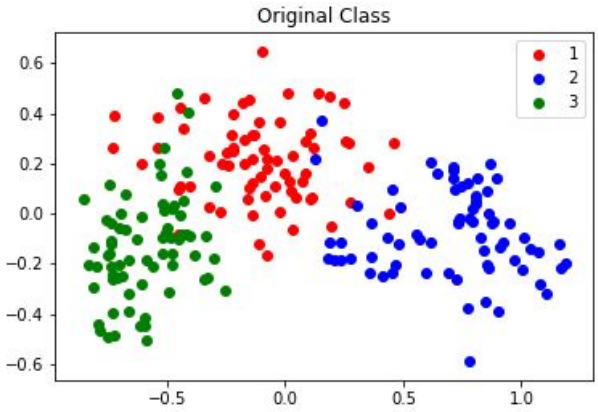
Dataset	Akuasi
Seed	0.890476190476

Hasil Akurasi Tiap Kluster

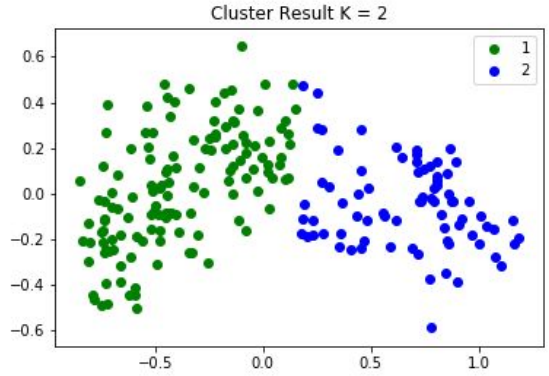
Kelas	Akurasi
1	0.840579710145
2	0.96875
3	0.87012987013

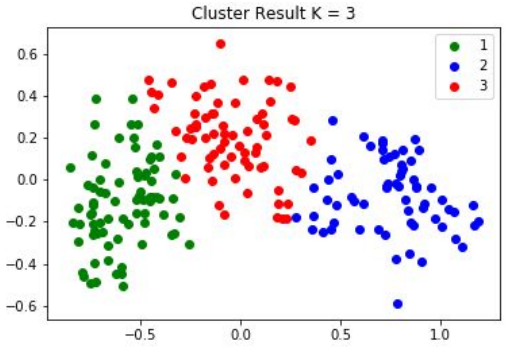
Plot

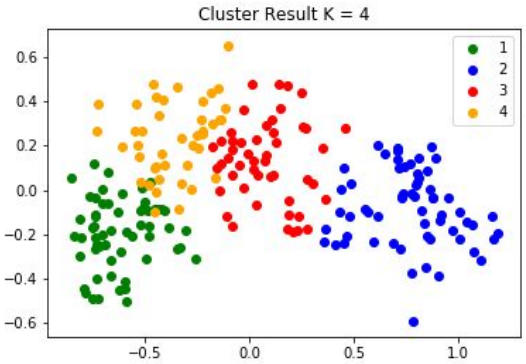
Kelas Asli

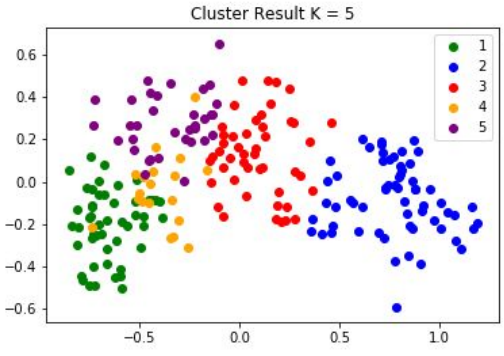


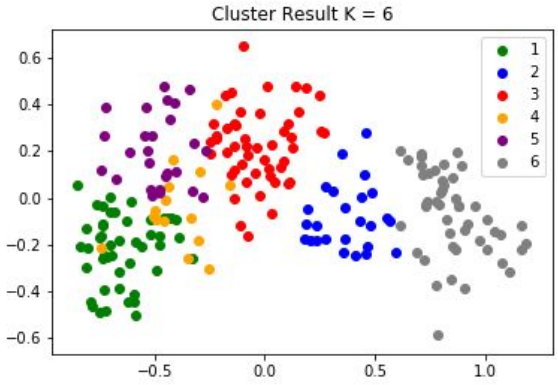
Cluster K = 2 hingga 6










Cluster Result K = 6
1
2
3
4
5
6
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.5
0.0
0.5
1.0