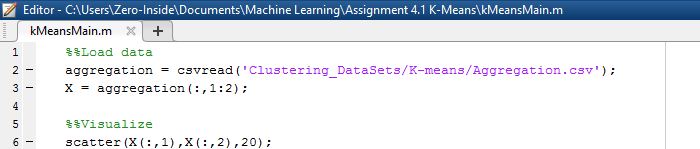
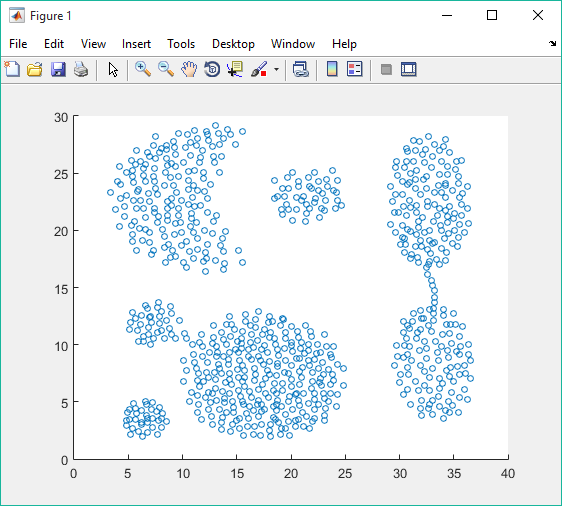
**K-Means Clustering**

1. Partitional Clustering
2. Load Data & Visualize

* kMeansMain.m



* Output

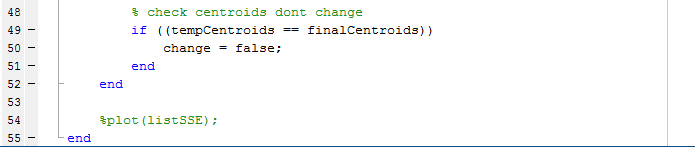
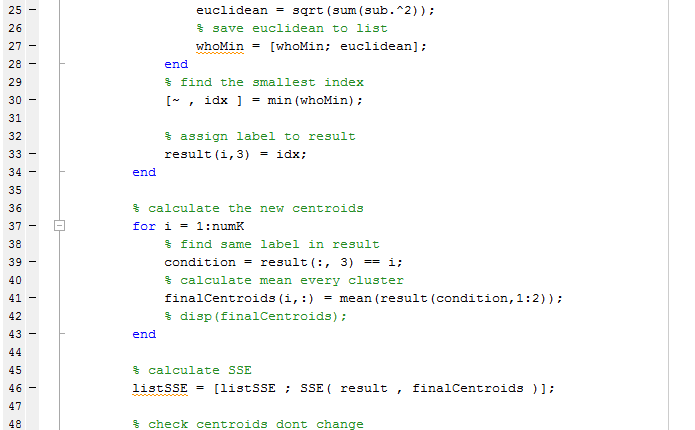
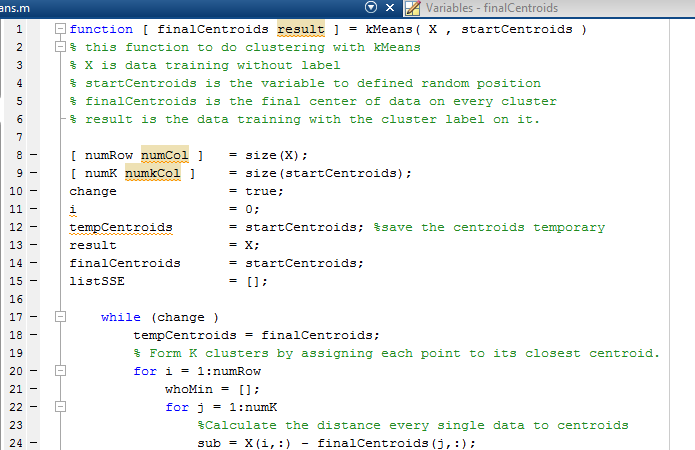


Analisis Program :

Pada tugas ini saya menggunakan dataset aggregation dengan nama file Aggregation.csv. kemudian meload data tersebut ke dalam variable X untuk ditampilkan. Menggunakan fungsi scatter menghasilkan visualisasi seperti output tanpa label class.

1. Apply K-Means
2. kMeans Function

* kMeans.m

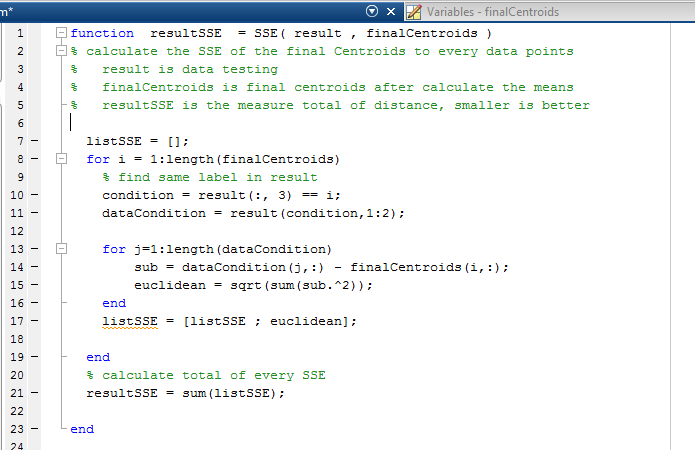


Analisis Program

Function kMeans dengan parameter input X sebagai data training tanpa label dan startCentroids dimana berisi nilai acak dari centroids pada masing-masing cluster. Dengan langkah berikut menglompokkan data sehingga terbentuk K buah cluster dengan titik centroid dari setiap cluster merupakan titik centroid yang telah dipilih sebelumnya dengan menghitung jarak terdekat setiap data point ke setiap centroid, kemudian memberi label datapoint tersebut dengan label yang paling kecil, Perbaharui nilai titik centroid dengan menghitung rata-rata dari setiap cluster. Ulangi langkah tersebut sampai centroid tidak lagi berubah.

1. SSE Function

* SSE.m

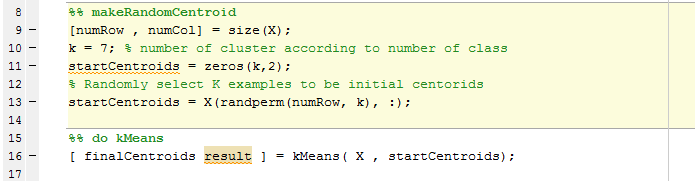


* Analisis Program

Fungsi SSE dengan parameter input berupa result sebagai data yang akan dihitung SSEnya dan finalcentroid sebagai centroid baru hasil proses k-means. Perhitungan SSE dilakukan pada setiap class dan pada class tersebut semua data dihitung jarak euclidean dari centroid itu class itu sendiri. Setelah mendapatkan jarak setiap class antar masing-masing centroidnya kemudian dijumlahkan dengan fungsi sum. Fungsi ini akan mengembalikkan nilai SSE yang akan digunakan untuk analisis performa k-means.

1. Run K-Means Algorithm

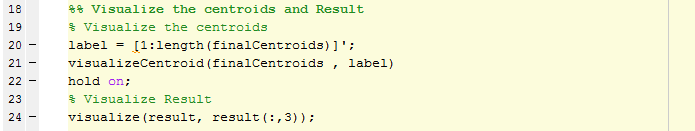
* kMeansMain.m



Analisis Program

Untuk menjalankan fungsi kMeans, terlebih dahulu menentukan centroids awal berupa random data points dari dataset dan banyak cluster yaitu K.

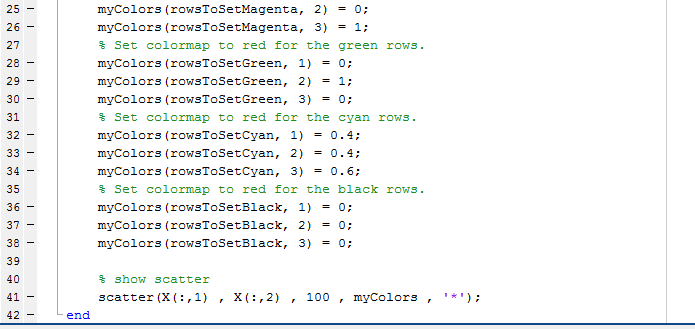
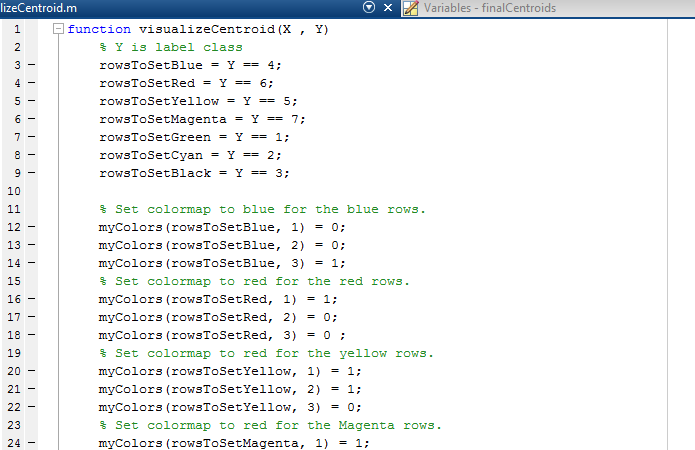
* kMeansMain.m



Analisis Program

Baris ke-20 untuk memberi label dari 1 sampai 7 (sesuai dengan banyak cluster). Baris 21 memvisualisasikan final centroids dengan scatter. Pada baris 24 memvisualisasikan result hasil dari kMeans.

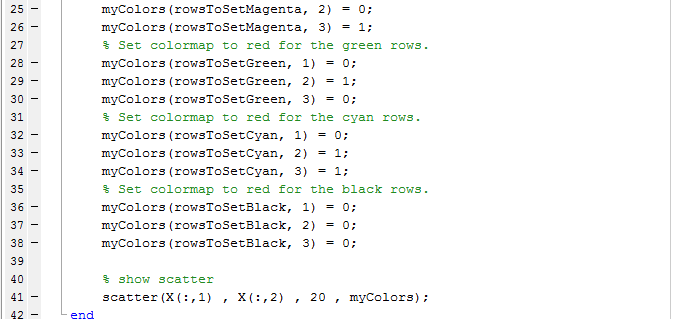
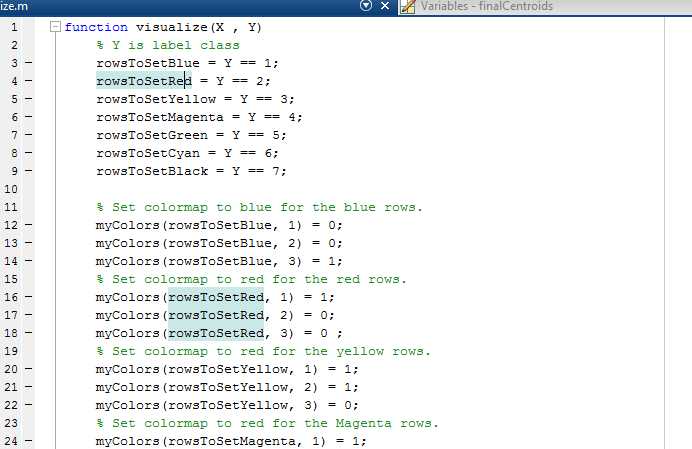
* visualizeCentroid.m



Analisis Program

Dengan menggunakan scatter, memvisualisasikan dataset lebih mudah. Terlebih dahulu meload dataset dan membagi data train dan data class label. Kemudian memanggil funsi visualize dengan parameter data dan label. Pada fungsi visualize(x,y) masing-masing data label class disesuaikan dengan data warna RGB. Kemudian memanggil scatter(param1,param2,param3,param4) dengan param1 dan param2 adalah kolom pertama dari data train dan kolom kedua dari data train, kemudian param3 adalah size dari ukuran plot, dan param4 adalah data color.

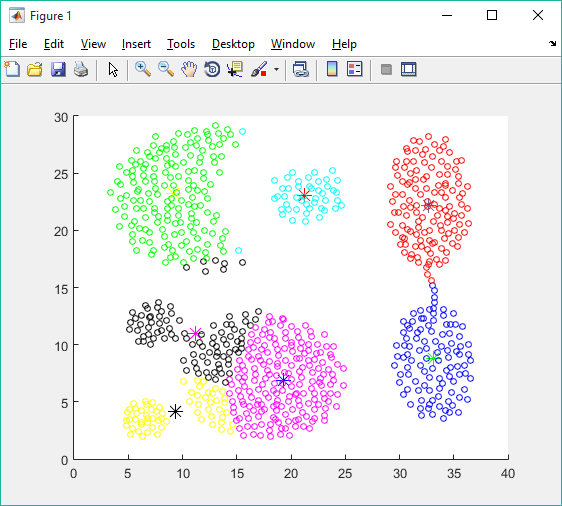
* visualize.m



Analisis Progam

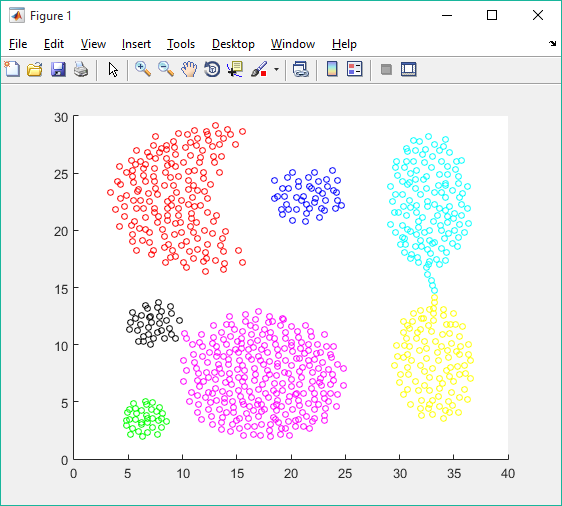
Sama seperti visualizeCentroids.

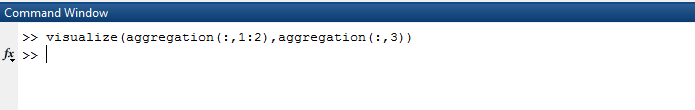
* Output



1. Visualize from training data.

* Output





Analisis Program

Dengan menjalankan visualize ditampilkan scatter plot terhadap data training dengan memasukkan label data asli.

1. Compare Result

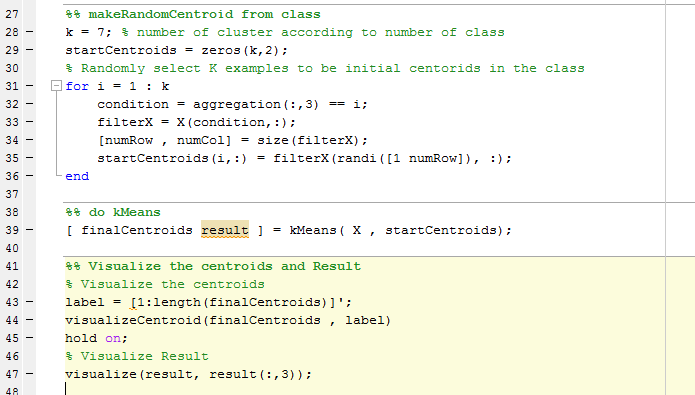
|  |  |
| --- | --- |
| Visualize of training Data | Visualize of k-Means data |
|  |  |

Analisis

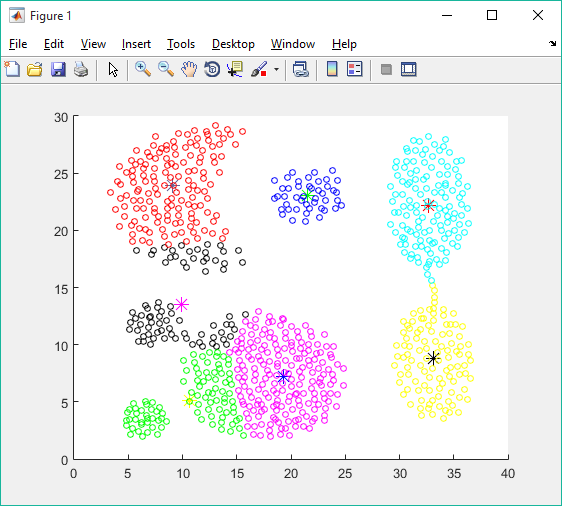
Hasil dari algoritma k-Means yang dijalankan terhadap data training tanpa label akan menghasilkan cluster(kelompok) data seperti table pada kolom ke 2. Terlihat bahwa ada beberapa data yang dikelompokkan melewati class yang seharusnya. Kemudian beberapa warna class juga ada yang berbeda. Karena initial centroids acak yang mungkin bernilai dengan class terdekat sehingga warna class berubah.

1. Do K-Means with initial centroids on each class and visualize it.

* kMeansMain.m



* Output



Analisis Program

Baris 31 sampai 36 akan menginisalisasi start centroids dengan masing-masing class. Untuk setiap class dipilih secara random datapoint yang ada. Kemudian melakukan kMeans dan menampilkannya dengan scatter.

1. Compare result

|  |  |
| --- | --- |
| Visualize of training Data | Visualize of k-Means data |
|  |  |

Analisis

Dengan menjalankan visualize dari data yang sudah dilakukan algoritma K-Means didapat hasil sesuai kolom ke-2 pada table. Diperoleh data class yang sesuai dengan yang asli tetapi ada missclassification pada kelompok data yang kecil. Seperti warna hitam dan warna hijau. Karena data yang ungu dan merah tersebar luas, maka titik centroids pada warna hitam dan warna hijau akan menganggap data-points terdekat dari mereka adalah bagian dari class yang tersebar luas karena melihat jarak terdekat dari antar cluster.

1. Compare result with random datapoints selected and random selected on each class as initial centroids.

|  |  |
| --- | --- |
| Visualize of k-Means data with random datapoints selected. | Visualize of k-Means data with random selected on each class |
|  |  |

Analisis

Hasil yang diperoleh terlihat perbedaan warna antar cluster, cluster yang menggunakan initial centroids pada random data sesuai dengan classnya akan sesuai dengan hasil warna pada training data dan jika initial centroids menggunakan random datapoints dari seluruh data training maka tidak menjamin warna yang akan dihasilkan sama dengan training data beserta labelnya karena ada unsur acak tersebut. Selain warna, lebih akurat memilih initial centroids dari setiap class karena initial centroids menentukan performasi K-Means untuk mengclusterkan dengan tepat setiap datapoints terhadap centroidsnya lebih dekat dengan kelompok classnya maka dengan mudah dikelompokkan sesuai classnya.