Jurnal Kajian Ilmiah (JKI)
e-ISSN: 2597-792X, ISSN: 1410-9794
Vol. 20 No. 1 (Januari 2020), Halaman: 103 – 118
Terakreditasi Peringkat 5 (SINTA 5) sesuai SK RISTEKDIKTI Nomor. 3/E/KPT/2019

Penggunaan Matlab dan Python dalam Klasterisasi Data

Herlawati [1,*], Rahmadya Trias Handayanto [2]

[1] Fakultas Teknik, Universitas Bhayangkara Jakarta Raya; Jl. Raya Perjuangan, Marga Mulya, Bekasi Utara, Jawa Barat 17121. Telp: 021-88955882, 889955883, e-mail: herlawati@ubharajaya.ac.id
[2] Fakultas Teknik, Universitas Islam 45; Jl. Cut Meutia No. 83, Bekasi Timur, Bekasi, Jawa Barat 17113. Telp: 021-8801027, 8802015, Fax: 021-8801192; e-mail: rahmadya.trias@gmail.com

* Korespondensi: e-mail: herlawati@ubharajaya.ac.id

Abstract

Organizations need to dig through the data clustering process, both past data and data from the internet. Sometimes the data has to be re-clustered to match the actual conditions. Therefore, it is necessary to prepare clustering support equipment. In this study the K-Means method was chosen for comparing two technical computational languages, i.e. Matlab and Python which are currently in great demand by researchers and can be used by organizations for a clustering process. This study showed both Matlab and Python have enough libraries (libraries) and toolboxes to help users in data clastering as well as graphics presentation. The test results show that the two programming languages are capable of carrying out the clustering process with two clusters; cluster 1 with a center point at coordinates (1.24, 1.34) and cluster 2 with a center point at coordinates (3.1, 3.07) and are presented by a cluster distribution plot.

Keywords: *Clusterization, K-Means, Matlab, Python.*

Abstrak

Organisasi perlu menggali data lewat proses klasterisasi data, baik data lampau maupun data dari internet. Terkadang data harus dilakukan klasterisasi ulang untuk mencocokan dengan kondisi yang sebenarnya. Oleh karena itu perlu dipersiapkan peralatan pendukung klasterisasi. Dalam penelitian ini metode K-Means dipilih untuk membandingkan dua bahasa komputasi teknis yaitu Matlab dan Python yang sekarang ini banyak diminati para peneliti yang dan dapat digunakan oleh organisasi yang membutuhkan proses klasterisasi. Hasil dari penelitian ini menunjukan baik Matlab maupun Python memiliki cukup pustaka (*library*) dan toolbox dalam membantu pengguna mengklasterisasi data, mempresentasikan grafik. Hasil pengujian menunjukan kedua Bahasa pemrograman mampu menjalankan proses klasterisasi berupa klaster 1 yang memiliki titik pusat yang berada pada koordinat (1.24, 1.34) dan klaster 2 dengan titik pusat yang berada pada koordinat (3.1, 3.07) disertai dengan plot sebaran klasternya.

Kata kunci: Klasterisasi, K-Means, Matlab, Python.

Submitted: **14 Desember 2019**; Revised: **28 Desember 2019**; Accepted: **11 Januari 2020**; Published: **25 Januari 2020**

1. Pendahuluan

Masalah klasterisasi merupakan masalah penting yang perlu dilakukan dalam setiap organisasi mengingat tersedianya data yang melimpah baik data lampau maupun yang berasal dari internet. Terkadang perlu mengklasterisasi ulang data yang sudah memiliki label untuk menguji apakah klasifikasi tersebut sesuai dengan kondisi real di lapangan. Sebagai bagian dari *unsupervised learning,* klasifikasi memerlukan data yang baik agar diperoleh hasil yang sesuai selain juga pemilihan bahasa pemrograman yang banyak dijumpai saat ini. Sebagai perbandingan, dalam penelitian ini dua bahasa komputasi terkenal ditunjukan penggunaannya yaitu Matlab dan Python. Matlab yang sejak dulu memiliki komunitas tersendiri dalam komputasi kini kian menjangkau ke segala bidang, sementara Python, sebagai bahasa dengan tren penggunaan yang meningkat mencuri perhatian para pengguna baru. Sebagai bahan referensi berikut ini hal-hal yang perlu dipahami sebelum melakukan proses klasterisasi.

A. Analisa Klaster (*Cluster Analysis*)

Analisa klaster (dikenal juga dengan istilah *data clustering*) adalah metode yang digunakan untuk membagi rangkaian data menjadi beberapa grup berdasarkan kesamaan-kesamaan yang telah ditentukan sebelumnya. Jadi secara umum dapat dikatakan bahwa (Gorunescu, 2011): 1) Data dalam satu klaster memiliki tingkat kesamaan yang tinggi, dan 2) Data dalam klaster yang berbeda memiliki tingkat kesamaan yang rendah.

Karena itu perlu diketahui teknik-teknik yang digunakan untuk mengukur tingkat kesamaan, antara lain: 1) *Minowski Distance* (Masuk dalam kelompok ini Manhattan, Eulidean, dan Chebysev) , 2) *Tanimoto Measure, 3) Pearson's r Measure, 4) Mahalanobis Measure.*

Contoh-contoh penerapan analisa klaster dapat dijumpai saat ini. Berikut ini beberapa diantaranya: 1) **Segmentasi Pasar**. Adalah pengklasteran data yang membagi pelanggan menjadi grup-grup tertentu yang akan mempermudah bagian penjualan (marketing) dalam memasarkan produk-produknya, seperti rumah, kendaraan, dan sebagainya. 2) **Pengklasteran Dokumen**. Dokumen-dokumen yang memiliki kemiripan yang sama, misalnya politik, ekonomi, dan bidang lainnya dikumpulkan dalam satu grup. Manfaat yang diperoleh adalah kemudahan dalam mencari, mengorganisir, dan mensuplai data-data yang akan dimanfaatkan oleh pengguna pada bidang yang sesuai. 3) **Pengklasifikasian Penyakit**. Penyakit tertentu dapat didetiksi dari gejala-gejala yang menyertainya. Oleh karena itu pengklasifikasian penyakit berdasarkan gejala sangat membantu para praktisi kesehatan dalam aktivitas kesehariannya, sehingga perlakuan yang tepat dapat diterapkan untuk tiap kasus penyakit tertentu. 4) **Pengklasifikasian dalam Biologi**. Biologi sangat membutuhkan proses klasifikasi, misalnya dalam bioinformatika untuk mencari gen-gen terbaik berdasarkan kelas-kelas yang terbentuk.

B. Pencarian Hukum Asosiasi (*Association Rule Discovery*)

Dalam satu grup, tiap anggota memiliki hubungan satu dengan yang lain, juga terhadap grup-grup yang lain. Keterhubungan itu selanjutnya dibuat hukum/aturan yang dapat dijadikan patokan kebijakan dari pengguna. Sebagai contoh data yang berasal dari penjualan supermarket. Misalnya berdasarkan data yang ada ternyata banyak pembelian beberapa botol minuman bir disertai dengan pembelian popok bayi. Sehingga pengaturan letak posisi stan penjualan minuman bir sebaiknya berdekatan dengan stan penjualan popok bayi. Contoh yang lain adalah pada aplikasi berbasis web, dimana kecenderungan suatu user mengakses situs tertentu setelah suatu situs dia akses dapat dijadikan patokan suatu perusahaan dalam menawarkan produk ke user tersebut. Kecenderungan user dapat diketahui dari aktivitasnya menelusuri dunia maya, misal setelah dia mengakses situs A akan cenderung mengakses situs B, dan seterusnya. (Widodo et al., 2013)

C. Pencarian Pola Berurutan (*Sequential Pattern Discovery*)

Banyak sistem yang cara kerjanya berurutan seperti prosedur-prosedur tertentu, akses ke suatu halaman situs, protein dalam rangkaian Asam Dioksi Ribonukleat (DNA), dan sebagainya. Dengan memiliki riwayat data pada sistem-sistem tersebut kita dapat mengetahui pola-pola yang dihasilkannya. Tugas perancang data mining adalah mengetahui pola-pola berurutan dari data-data yang tersebar tersebut. Contoh aplikasi-aplikasi yang diterapkan misalnya: **Kesatu,** Dalam suatu pasar swalayan diketahui data transaksi nasabah yang belanja di situ. Jika ditelusuri, seorang nasabah akan diketahui rangkaian urutan belanja dalam rentang waktu tertentu. **Kedua,** Dalam dunia kedokteran beberapa penyakit memiliki gejala-gejala yang berurutan. Misalnya demam berdarah dimulai dengan demam yang disertai bercak-bercak tertentu pada kulit, dilanjutkan dengan fase kritis dimana demam mula mereda, dan seterusnya. **Ketiga,** Dalam dunia meteorologi, rangkaian cuaca dapat digunakan untuk memprediksi iklim yang terjadi dan dapat memprediksi adanya badai, tsunami, dan bencana lainnya yang melibatkan perubahan cuaca.

Klasterisasi memiliki perbedaan dengan klasifikasi. Jika pada klasifikasi model dilatih agar mampu mengarahkan data masuk ke kelas tertentu, pada klasterisasi model diminta membuat kelas-kelas berupa klaster berdasarkan kemiripannya. Karena ketika pelatihan melibatkan data yang sudah memiliki kelas, klasifikasi masuk dalam kategori pembelajaran terpandu (*supervised learning*). Klasterisasi berbeda dengan klasifikasi dari sisi data, dimana pada metode itu data tidak memiliki kelas (sering diistilahkan dengan label atau target), sehingga klasterisasi masuk dalam kategori pembelajaran tak terpandu (*unsupervised learning*).

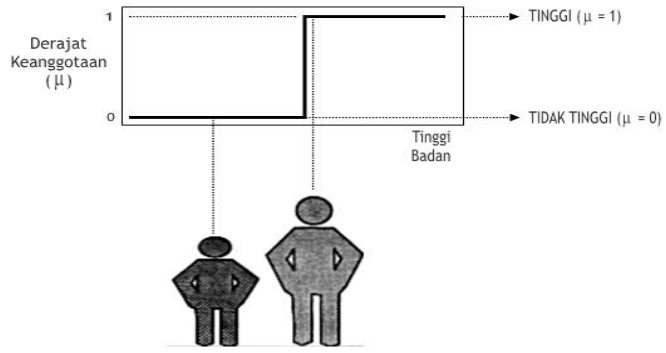
Banyak metode klasterisasi yang digunakan saat ini, antara lain: Fuzzy C-Means Clustering, K-Means, K-Medoids, CLARA, dan lain-lain. Banyak literatur yang membedakan klasterisasi berdasarkan metode partisinya, hirarki, densitas, dan grid. Buku ini fokus ke

Submitted: **14 Desember 2019**; Revised: **28 Desember 2019**; Accepted: **11 Januari 2020**; Published: **25 Januari 2020**

klasterisasi berdasarkan metode partisinya, terutama yang menggunakan metode-metode berbasis mesin pembelajaran (*machine learning*) yang merupakan bidang yang paling banyak berkembang saat ini. Secara garis besar, dua metode partisi yakni klasterisasi kasar (*Hard Clustering*) dan klasterisasi halus (*Soft Clustering*) banyak diterapkan saat ini. Contoh klasterisasi kasar adalah K-Means dan turunan-turunannya, sementara yang klasterisasi halus adalah Fuzzy C-Means dengan konsep Fuzzy yang diterapkannya.

Klasterisasi kasar membandingkan satu kelas dengan kelas lainnya melalui mekanisme biasa yang tidak mengkonversi angka utuh (*Crisp*) menjadi kabur (*Fuzzy*). Jika kita membuat aturan jika seorang siswa memiliki nilai lebih besar dari 80 maka akan memperoleh nilai “A”. Seorang siswa memiliki nilai 79,9 maka siswa tersebut memiliki nilai “B” karena di bawah 80 nilainya. Jika Anda lihat dari sisi keadilan tentu saja sistem tersebut terlihat kaku (walaupun terasa lebih adil). Tetapi dari sisi manusiawi, tentu saja kita tidak boleh melupakan aspek nilai 0,1 apa yang membuat siswa tersebut gagal memperoleh nilai “A”.

Atau kasus tinggi badan dimana batas ketinggian dimana seseorang dikatakan tinggi dan dikatakan rendah sangat tegas (Gambar 1). Pembagian antara kelas orang yang tinggi dengan kelas orang yang rendah sangat tegas. Ketegasan terlihat dari batas pemisah yang tegak lurus, hanya ada dua kemungkinan suatu harga akan jatuh pada kategori tinggi atau jatuh pada kategori rendah.

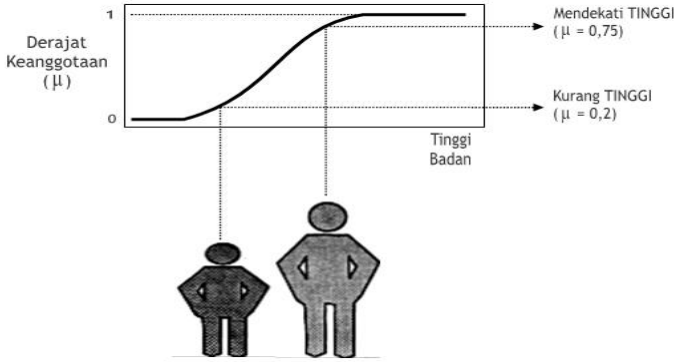


Sumber: (Widodo et al., 2013)

Gambar 1. Pembagian Tegas Pada Sistem Non-Fuzzy

Sebaliknya perhatikan gambar 2 di bawah ini, dimana antara orang yang masuk kategori tinggi dan rendah tidak memiliki batas yang tegas. Orang sebelah kanan tidak disebut tinggi melainkan dikatakan mendekati tinggi dengan derajat keanggotaan μ sebesar 0,75. Sebaliknya orang yang disebelah kanan tidak hanya dikatakan rendah, tetapi bisa juga dikatakan kurang tinggi dengan derajat keanggotaan μ sebesar 0,2. Manfaat dari adanya kekaburan dalam bentuk derajat keanggotaan μ adalah ketika kita tidak hanya memperhatikan tinggi atau rendahnya seseorang dari ketinggian saja, misalnya dengan memperhatikan faktor

usia, berat badan atau jenis kelamin. Seorang wanita yang masuk kategori rendah bisa saja dikatakan tinggi karena dia seorang wanita yang memang secara umum tinggi badan wanita di bawah pria.



Sumber: (Widodo et al., 2013)

Gambar 2. Klasifikasi Berdasarkan Logika Kabur

Pada mulanya klasterisasi bermaksud mengelompokan data menjadi klaster-klaster yang khas. Tiap klaster harus memiliki anggota yang mirip satu sama lain dan berbeda dengan anggota klaster lainnya. Namun hasil pengelompokan tersebut dapat dimanfaatkan untuk mengelompokan satu data baru. Kebanyakan dimanfaatkan dengan klasifikasi lewat titik pusat klaster yang dihasilkan. Klasifikasi jenis ini biasanya karena data yang akan digunakan untuk klasifikasi belum memiliki label/kelas, sehingga perlu dilakukan proses klasterisasi. Tentu saja jika data telah memiliki label/kelas dan dari sumber yang terpercaya, tidak perlu proses klasterisasi, melainkan langsung melatih model. (Miyamoto et al., 2008)

Beberapa penelitian terkait dengan Klasterisasi K-Means telah dilakukan, salah satunya klasterisasi warna. Warna dan ukuran adalah salah satu fitur terpenting untuk klasifikasi kematangan buah yang akurat. Petani usaha kecil menggunakan evaluasi manual melalui pengamatan visual untuk mengklasifikasikan kematangan pick mereka. yang menurut FAMA ada enam indeks jatuh tempo. Proses berulang itu membosankan dan rentan terhadap kesalahan manusia. Makalah ini berfokus pada identifikasi kematangan buah mangga. Raspberry Pi adalah komputer kecil, yang cukup kuat untuk menjalankan algoritma pemrosesan gambar dipilih untuk sistem ini. Algoritma pemrosesan gambar yang dikembangkan mampu menentukan ukuran buah dan menerapkan pengelompokan K-means untuk menentukan warna buah. (Mustaffa & Mohd Khairul, 2017).

Klasterisasi dapat juga diterapkan untuk mendukung strategi perusahaan listrik. Untuk menggambarkan perilaku pelanggan secara akurat dan memandu departemen listrik untuk menyesuaikan strategi pembangkit listrik secara efektif, algoritma K-means plus clustering berdasarkan Python diusulkan untuk mengklasifikasikan data konsumsi daya di Taiyuan. Dengan mengekstraksi data listrik perusahaan, nomor clustering K yang paling cocok

Submitted: **14 Desember 2019**; Revised: **28 Desember 2019**; Accepted: **11 Januari 2020**; Published: **25 Januari 2020**

ditemukan. Algoritma K-means plus clustering mengklasifikasikan data konsumsi listrik dan akhirnya mendapatkan lima jenis pengguna yang berbeda. Dan kemudian, kondisi ekonomi rumah tangga pengguna dianalisis. Telah diverifikasi bahwa algoritma K-Means plus clustering lebih cepat daripada K-means dan hasil clustering lebih akurat. (Zao et al., 2017).

Penelitian berikutnya membahas klasterisasi data di internet. Dengan meningkatnya data multi-media melalui Internet, kueri dengan contoh deteksi istilah terucapkan (QbE-STD) telah menjadi penting dalam menyediakan mekanisme pencarian untuk menemukan kueri yang diucapkan dalam audio lisan. Algoritma pencarian audio harus efisien dalam hal kecepatan dan memori untuk menangani file audio besar. Secara umum, pendekatan yang berasal dari algoritma Dynamic Time Warping (DTW) yang terkenal menderita masalah skalabilitas. Untuk mengatasi masalah seperti itu, algoritma DTW (IR-DTW) berbasis Pengambilan Informasi telah diusulkan baru-baru ini. IR-DTW meminjam teknik dari komunitas Pengambilan Informasi untuk mendeteksi daerah yang lebih mungkin mengandung permintaan yang diucapkan dan kemudian menggunakan DTW standar untuk mendapatkan waktu mulai dan akhir yang tepat. Salah satu kelemahan IR-DTW adalah waktu yang dibutuhkan untuk pengambilan titik referensi serupa untuk titik kueri yang diberikan. Dalam makalah ini kami mengusulkan metode untuk meningkatkan kinerja pencarian algoritma IR-DTW menggunakan teknik berbasis clustering. Metode yang diusulkan telah menunjukkan perkiraan kecepatan 2400X. (Mantena & Anguera, 2013). Bahkan dapat juga diterapkan untuk sebuah sistem *load balancing* (Pangestu et al., 2018) dengan sebuah *classifier* tertentu.

Dengan banyaknya manfaat diterapkannya proses klasterisasi di sebuah organisasi maka diperlukan peralatan pendukung seperti bahasa pemrograman, metode yang cocok, dan kemudahan-kemudahan lainnya. Penelitian ini mencoba membandingkan dua bahasa pemrograman yang layak untuk diterapkan untuk klasterisasi data dengan kasus khusus pendukung keputusan penerimaan beasiswa dengan metode K-Means, sebagai alternatif metode lainnya seperti *Analytic Hierarchy Process* (AHP) (Khasanah et al., 2015).

2. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini, terdiri dari:

A. Mencari Pusat Klaster

Suatu data termasuk dalam klaster tertentu ketika jaraknya terdekat dengan pusat klasternya dibandingkan jarak terhadap pusat klaster yang lain. Sebelum aplikasi SPK berbasis FCM dibuat, terlebih dahulu kita mencari pusat klaster data yang akan digali.

B. Klasterisasi K-Means dengan Python

Klasterisasi merupakan salah satu klasifikasi tak terpandu (*unsupervised classification*). Ciri khas klasifikasi tak terpandu adalah pada data latih tidak tersedia target atau label yang

berisi informasi kelas tiap-tiap tuple. Tahapan dalam klasterisasi dengan K-Means antara lain: 1) Menentukan jumlah klaster "K" yang akan dibagi. 2) Menentukan data k sebagai titik pusat (*centroid*) awal tiap klaster. 3) Mengelompokan data ke dalam K cluster sesuai dengan titik pusat yang telah ditentukan sebelumnya. 4) Memperbaharui pusat klaster dan mengulangi langkah ketiga sampai nilai dari titik centroid tidak berubah. Penentuan suatu tuple masuk klaster mana adalah dengan menggunakan jarak. Biasanya menggunakan Euclidean dan Manhattan.

C. Menyiapkan Dataset

Dengan menggunakan Jupyter Notebook. Buat sel baru dan impor pustaka-pustaka yang diperlukan. Jika sudah eksekusi sel tersebut dan pastikan tidak ada pesan kesalahan.

D. Memanggil K-Means

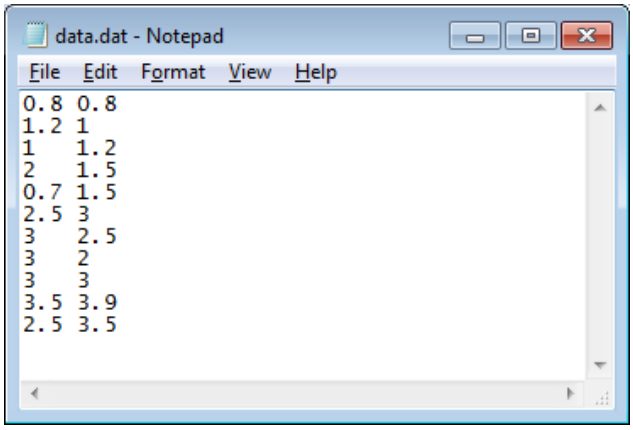
K-Means dapat diakses dari pustaka "Scikit-Learning" atau dalam pustaka Python dikenal dengan "Sklearn". Huruf "K" pada K-Means bermakna jumlah klaster pada data yang akan dikelompokan. Berikut ini lanjutan dari kode sebelumnya.

3. Hasil dan Pembahasan

A. Mencari Pusat Klaster

Suatu data termasuk dalam klaster tertentu ketika jaraknya terdekat dengan pusat klasternya dibandingkan jarak terhadap pusat klaster yang lain. Sebelum aplikasi SPK berbasis FCM dibuat, terlebih dahulu kita mencari pusat klaster data yang akan digali. Dengan meggunakan Matlab, arahkan ***Current Directory*** ke folder kerja kemudian lakukan langkah berikut:

Pertama, Buka notepad, misalnya kita memiliki data ipk dan tingkat kemiskinan mahasiswa. Jangan lupa simpan dengan ekstensi ***dat*** agar bisa diolah oleh Matlab.



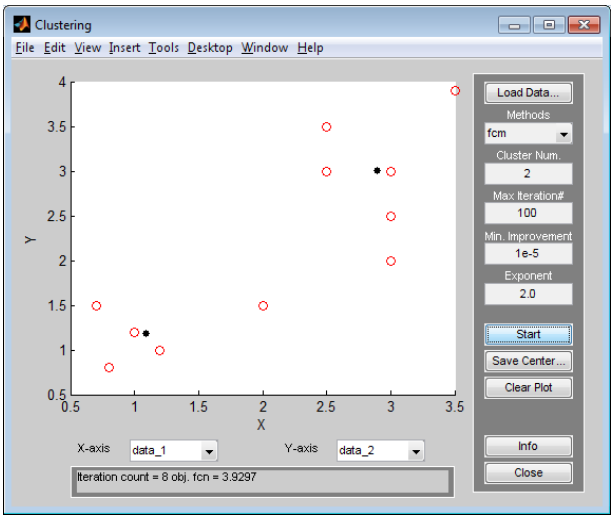
Sumber: Hasil Penelitian (2020)

Gambar 3. Data IPK Mahasiswa dan Tingkat Kemiskinan

Submitted: **14 Desember 2019**; Revised: **28 Desember 2019**; Accepted: **11 Januari 2020**; Published: **25 Januari 2020**

Kedua, Jika data sudah dibuat dan diletakkan di folder yang sama dengan folder kerja Matlab, ketik Load Data untuk mengambil data tersebut di Command Window.

Ketiga, Ketik findcluster di Command Window untuk memanggil jendela Clustering yang akan kita gunakan untuk mencari titik pusat klaster. Tekan Load Data untuk memanggil data mahasiswa yang akan kita cari titik pusat klasternya. Jika data yang baru dibuat terletak di folder yang sama, maka akan muncul di jendali 'Load Data'. Klik ganda file-nya sehingga muncul grafiknya.



Sumber: Hasil Penelitian (2020)

Gambar 4. Jendela Clustering

Keempat, Di sebelah kanan jendela banyak isian yang harus kita isi. Untuk metode 'Methods' pilih fcm. Jumlah klaster Cluster Num jika kita akan membagi dua kategori misalnya dapat beasiswa dan tidak dapat beasiswa, maka kita isi 2. Tetapi bisa saja kita mengisi lebih dari dua, misalnya kita akan membagi menjadi tidak dapat beasiswa, beasiswa setengah dan dapat beasiswa (beasiswa 100 persen).

Kelima, Setelah tombol 'Start' ditekan, Matlab akan menghitung pusat klaster yang jika sudah selesai akan muncul dua titik hitam. Titik hitam itu merupakan pusat klaster. Simpan dengan menekan tombol 'Save Center'. Beri nama, misalnya 'center.dat'. Titik pusat klaster ini yang kita jadikan alat penentu keputusan dari data baru yang akan dicari kecenderungannya.

B. Klasterisasi K-Means dengan Python

Klasterisasi merupakan salah satu klasifikasi tak terpandu (*unsupervised classification*). Ciri khas klasifikasi tak terpandu adalah pada data latih tidak tersedia target atau label yang berisi informasi kelas tiap-tiap tuple.

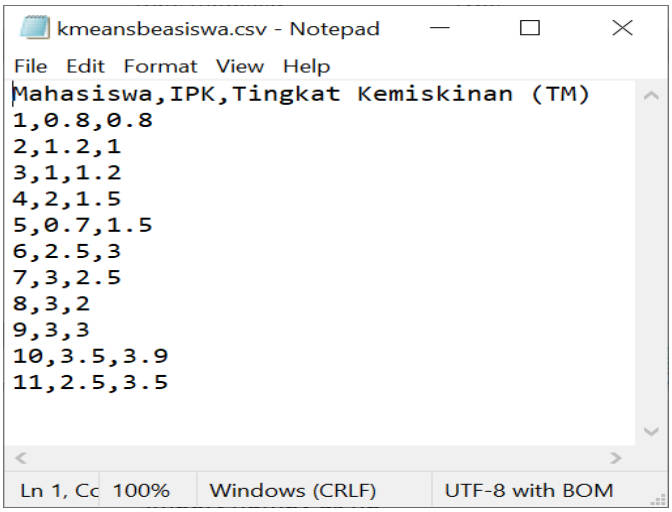
Penentuan suatu tuple masuk klaster mana adalah dengan menggunakan jarak. Biasanya menggunakan Euclidean dan Manhattan. Untuk mempraktekannya kita bisa menggunakan data beasiswa, tetapi tanpa target/label.

Tabel 1. Data Mahasiswa untuk training dengan Python

Mahasiswa	IPK	Tingkat Kemiskinan (TM)
1.	0.8	0.8
2.	1.2	1
3.	1	1.2
4.	2	1.5
5.	0.7	1.5
6.	2.5	3
7.	3	2.5
8.	3	2
9.	3	3
10.	3.5	3.9
11.	2.5	3.5

Sumber: Hasil Penelitian (2020)

Tulis data tersebut di Excel dan simpan dengan nama “kmeansbeasiswa.csv” yang nanti akan kita oleh dengan algoritma K-Means. Pastikan file ketika dibuka dengan text editor, misalnya notepad, antar kolom terpisah dengan koma.



Sumber: Hasil Penelitian (2020)

Gambar 5. Tampilan Notepad untuk kmeansbeasiswa

CATATAN: Jika antar kolom terpisah dengan titik koma “;”, set ulang Excel Anda. Lihat bab mengenai “mengelola data dengan python” pada bagian “mengimpor data scv”.

C. Menyiapkan Dataset

1. Kembali buka Jupyter Notebook. Beri nama, misalnya “kmeansbeasiswa.ipynb”. Buat sel baru dan impor pustaka-pustaka yang diperlukan. Jika sudah eksekusi sel tersebut dan pastikan tidak ada pesan kesalahan.

```
#Impor Pustaka
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
```

2. Berikutnya adalah proses mengimpor dataset dari file “kmeansbeasiswa.csv”.

```
#Mengambil data dari file csv
```

Submitted: **14 Desember 2019**; Revised: **28 Desember 2019**; Accepted: **11 Januari 2020**; Published: **25 Januari 2020**

```
datasiswa=pd.read_csv('kmeansbeasiswa.csv')
df=pd.DataFrame(datasiswa,columns=['Mahasiswa','IPK','Tingkat Kemiskinan (TM)'])
X=np.asarray(datasiswa)
x_train=X[:,1:]
x_train
```

Pastikan ketika dieksekusi, data yang akan dilatih (x_train) muncul.

D. Memanggil K-Means

K-Means dapat diakses dari pustaka “Scikit-Learning” atau dalam pustaka Python dikenal dengan “Sklearn”. Huruf “K” pada K-Means bermakna jumlah klaster pada data yang akan dikelompokan. Berikut ini lanjutan dari kode sebelumnya.

Pertama, Buat sel baru dan panggil K-Means dengan kode berikut.

```
#Proses klasterisasi dengan K-Means
kmeans = KMeans(n_clusters=2)
kmeans.fit(x_train)
```

Silahkan pelajari lebih lanjut parameter-parameter K-Means di situs resmi https://scikit-learn.org. Eksekusi sel tersebut hingga dihasilkan model K-Means berikut ini. Pastikan tidak ada kesalahan yang muncul.

```
#Proses pengklusteran dengan K-Means
kmeans = KMeans(n_clusters=2)
kmeans.fit(x_train)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means+
+', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precomput
e_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

Sumber: Hasil Penelitian (2020)

Gambar 6. Tampilan Proses Pengklusteran dengan K-Means

Kedua, Proses “fitting” di atas menghasilkan pusat center (*centroid*) dan label/target untuk tiap-tiap tuple data beasiswa.

```
pusat=kmeans.cluster_centers_
labels=kmeans.labels_
print(pusat)
print(labels)
```

Ketiga, Berikutnya adalah visualisasi hasil dengan menggunakan grafik/plot. Buat satu sel baru khusus untuk mengelola grafik.

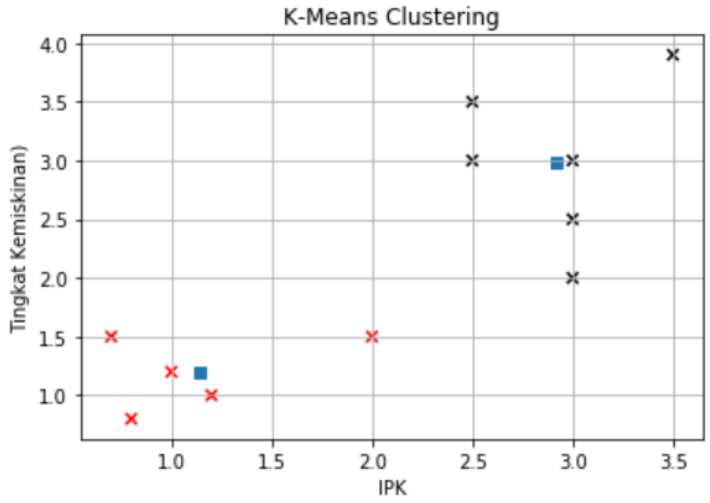
```
#Visualisasi Data
absis=[x_train[:,0]]
ordinat=[x_train[:,1]]
target=labels+absis-absis
plt.scatter(absis, ordinat, c=target, cmap='flag', marker='x')
#Label
plt.xlabel("IPK")
plt.ylabel("Tingkat Kemiskinan)")
plt.title("K-Means Classification")
plt.scatter(pusat[:,0],pusat[:,1], marker='s')
plt.grid()
```

Absis dan ordinat akan dipetakan ke grafik. Target akan memberikan warna yang berbeda untuk klaster yang berbeda.

CATATAN: Di sini ada sedikit modifikasi agar dihasilkan format yang sama dengan absis dan ordinat lewat sedikit manipulasi untuk menghindari pesan kesalahan: ValueError: 'c' argument has 11 elements, which is not acceptable for use with 'x' with size 11, 'y' with size 11. Silahkan menggunakan teknik lain jika ada, misalnya konversi array ke list, dan lain-lain.

Pada Gamabr 7, di sini ada dua warna yang mewakili klaster (beasiswa atau tidak). Titik dengan “marker” kotak menggambarkan *centroid* tiap klaster. Jika ada data baru, tinggal diukur mana jarak yang terdekat terhadap dua titik pusat tersebut untuk memutuskan klaster data tersebut.

```
#Label
plt.xlabel("IPK")
plt.ylabel("Tingkat Kemiskinan)")
plt.title("K-Means Clustering")
plt.scatter(pusat[:,0],pusat[:,1], marker='s')
plt.grid()
```



Sumber: Hasil Penelitian (2020)

Gambar 7. Tampilan Grafik/Plot dengan K-Means

Berikut contoh kasus yang lain jika mempunyai data seperti pada Tabel 2 berikut ini, yang harus dipecahkan permasalahannya yaitu: 1) Tunjukkan titik-titik pusat masing-masing klasternya dan 2) Jika seorang mahasiswa memiliki IPK = 3,8 dan TM = 2,4 maka termasuk klaster yang mana.

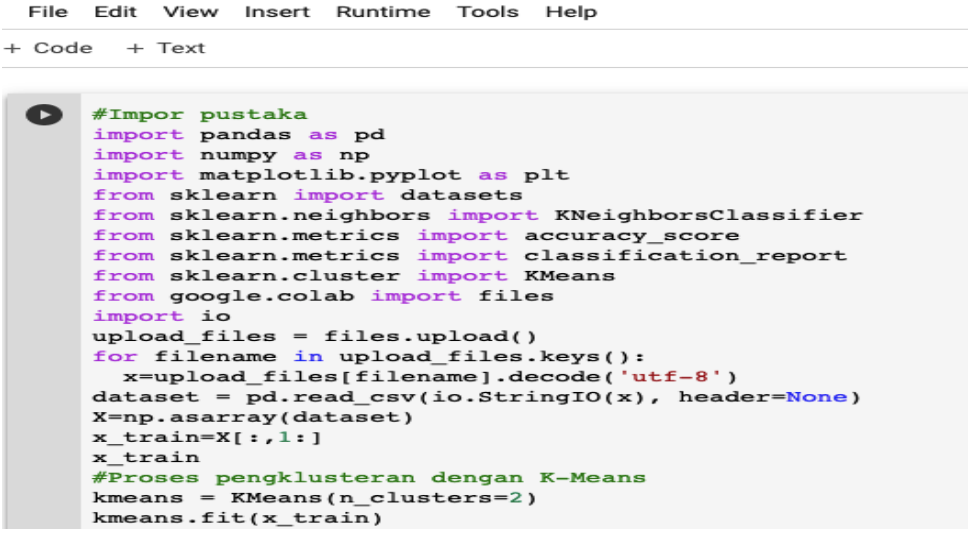
Submitted: **14 Desember 2019**; Revised: **28 Desember 2019**; Accepted: **11 Januari 2020**; Published: **25 Januari 2020**

Tabel 2. Kelompok mahasiswa berdasarkan IPK dan Tingkat Kemiskinan (TM)

Mahasiswa	IPK	Tingkat Kemiskinan (TM)
1	0.8	0.8
2	1.2	1.7
3	1	1.2
4	2.5	1.5
5	0.7	1.5
6	2.5	3
7	3	2.5
8	3.4	2.7
9	3	3.2
10	3.5	3.9
11	2.5	3.5

Sumber: Hasil Penelitian (2020)

Tahap pertama, kunjungi Google Colab pada halaman colab.research.google.com, kemudian buat Notebook baru dengan klik **New Notebook,** Jika sudah akan muncul notebook baru, ubah judul pada notebook tersebut dengan format namafile.ipynb (contoh: **beasiswa.ipynb**). Lanjutkan dengan Klik **+Code** dan pastekan code yang telah diberikan sebelumnya, berikut adalah tampilan potongan kode tersebut.



Sumber: Hasil Penelitian (2020)

Gambar 8. Tampilan input source code di colab.research.google.com

Berikut source code lengkapnya.

```
#Impor pustaka
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.cluster import KMeans
from google.colab import files
import io
upload_files = files.upload()
for filename in upload_files.keys():
  x=upload_files[filename].decode('utf-8')
dataset = pd.read_csv(io.StringIO(x), header=None)
X=np.asarray(dataset)
x_train=X[:,1:]
x_train
#Proses pengklusteran dengan K-Means
kmeans = KMeans(n_clusters=2)
kmeans.fit(x_train)
pusat=kmeans.cluster_centers_
labels=kmeans.labels_
print(pusat)
```

```
            print(labels)
            #Visualisasi Data
            absis=[x_train[:,0]]
            ordinat=[x_train[:,1]]
            target=labels+absis-absis
            plt.scatter(absis, ordinat, c=target, cmap='flag',
marker='x')
            #Label
            plt.xlabel("IPK")
            plt.ylabel("Tingkat Kemiskinan (TM)")
            plt.title("K-Means Clustering")
            plt.scatter(pusat[:,0],pusat[:,1], marker='s')
            plt.grid()
```

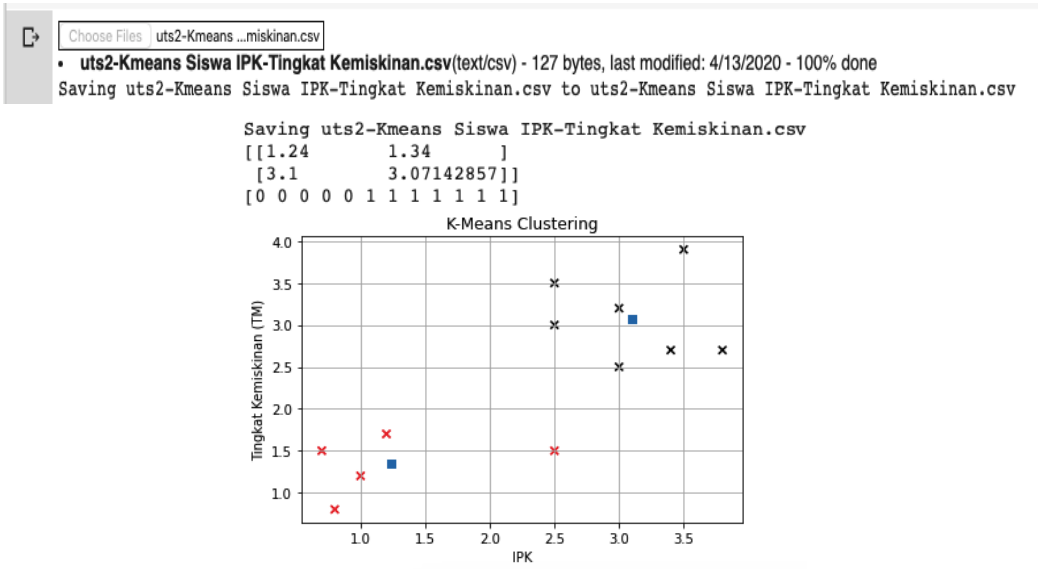
Tahap kedua, Siapkan data pendukung berupa file dengan isian data yang akan kita buat K-Means Clustering tersebut dengan format CSV. Data tersebut diambil dari data tabel pada soal dengan penulisan atau format: *nomor*, *ipk*, *tingkat kemiskinan* (pisahkan dengan koma). Pada baris nomor 12, tambahkan data sesuai pada soal yakni dengan IPK=3.8 dan TM=2.7 agar kita dapat mengetahui siswa tersebut masuk ke klaster mana. Tampilannya terlihat pada Gambar 9.

1	1,0.8,0.8
2	2,1.2,1.7
3	3,1,1.2
4	4,2.5,1.5
5	5,0.7,1.5
6	6,2.5,3
7	7,3,2.5
8	8,3.4,2.7
9	9,3,3.2
10	10,3.5,3.9
11	11,2.5,3.5
12	12,3.8,2.7

Sumber: Hasil Penelitian (2020)

Gambar 9. Tampilan input data di MS. Excel

Tahap Ketiga, Simpan data tersebut dengan format csv dengan namafile.csv contoh: **Tingkat Kemiskinan.csv**. Selanjutnya running code yang sudah dituliskan di Google Colab, maka setelah berjalan program akan meminta file untuk di upload. Klik tombol **Choose Files** dan upload file **Tingkat Kemiskinan.csv** yang sudah dibuat sebelumnya.

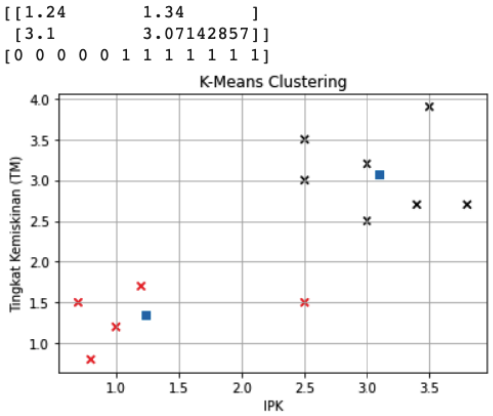


Sumber: Hasil Penelitian (2020)

Gambar 10. Tampilan hasil K-Means Clustering

Submitted: **14 Desember 2019**; Revised: **28 Desember 2019**; Accepted: **11 Januari 2020**; Published: **25 Januari 2020**

Kemudian program akan memproses dari data yang sudah diupload tersebut. Maka akan muncul grafik K-Means Clustering yang memiliki 2 klustering tersebut. Titik Pusat ditandai dengan titik berwarna biru. DiperolehTitik Pusat masing-masing klaster terlihat pada Gambar 11.



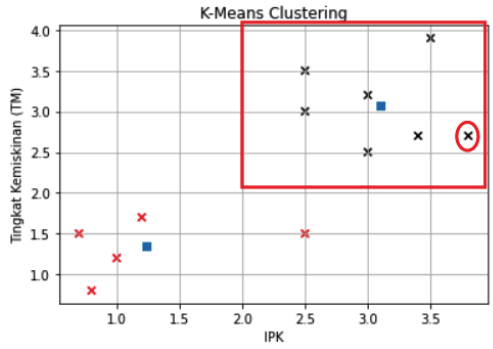
Sumber: Hasil Penelitian (2020)

Gambar 11. Tampilan Titik-titik Pusat K-Mens Clustering

Dari hasil tersebut, diketahui terdapat 2 klaster yang ditandai dengan masing-masing titik pusatnya berwarna biru. **Klaster 1** memiliki titik pusat yang berada pada koordinat (1.24, 1.34) yang setiap anggota klasternya ditandai dengan titik x berwarna merah. **Klaster 2** memiliki titik pusat yang berada pada koordinat (3.1, 3.07) yang setiap anggota klasternya ditandai dengan titik x berwarna hitam.

Jika ada mahasiswa dengan nilai IPK = 3,8 dan Tingkat kemiskinan =2,7 maka hasil dari klasterisasi terlihat pada Gambar 12.

Jadi diperoleh mahasiswa yang memiliki nilai IPK=3.8 dan TM=2.7 masuk ke dalam klaster 2 yang memiliki titik pusat pada koordinat (3.1, 3.07). Hal tersebut dapat dilihat pada titik x yang berwarna hitam. Koordinat siswa tersebut ditandai dengan lingkaran berwarna merah pada gambar 12.



Sumber: Hasil Penelitian (2020)

Gambar 12. Tampilan hasil K-means Clustering untuk mahasiswa yang baru dtambahkan

4. Kesimpulan

Berdasarkan data hasil uji yang diperoleh di dalam penelitian, kedua Bahasa pemrograman mampu mengklasterisasi data. Baik Matlab maupun Python memiliki cukup pustaka (*library*) dan toolbox dalam membantu pengguna mengklasterisasi data, mempresentasikan grafik. Untuk mengetahui struktur algoritma pun dapat dilakukan dengan membuka fungsi m-file maupun *py bawaan fungsi klasterisasi tersebut. Salah satu keunggulan Python adalah dukungan dari perusahaan raksasa Google dalam menyediakan Jupyter Notebook online-nya untuk pemrograman yang tidak berbayar. Di sisi lain Matlab memiliki toolbox-toolbox terstandar yang dapat digunakan sebagai "alat ukur" kinerja algoritma yang diusulkan yang mempermudah peneliti lain mengujinya sehingga banyak digunakan dalam tulisan ilmiah internasional. Perlu penelitian lebih lanjut mengenai penerapan praktisnya baik dalam aplikasi web maupun desktop dengan pilihan-pilihan *framework* yang tersedia.

Daftar Pustaka

Gorunescu, F. (2011). *Data Mining – Concepts, Models, and Techniques*. Springer-Verlag Berlin Heidelberg.

Khasanah, F. N., Grafika, J., Ugm, N., Grafika, J., Ugm, N., Grafika, J., Ugm, N., & High, A. S. (2015). *Fuzzy MADM for Major Selection at Senior High School*. 41–45.

Mantena, G., & Anguera, X. (2013). Speed improvements to Information Retrieval-based dynamic time warping using hierarchical K-Means clustering. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. https://doi.org/10.1109/ICASSP.2013.6639327

Miyamoto, S., Ichihashi, H., & Honda, K. (2008). *Algorithms for Fuzzy Clustering – Methods c-Means Clustering with Application*. Springer-Verlag Berlin Heidelberg.

Mustaffa, I. B., & Mohd Khairul, S. F. Bin. (2017). Identification of fruit size and maturity through fruit images using OpenCV-Python and Rasberry Pi. *International Conference on Robotics, Automation and Sciences (ICORAS)*, *27-29 Nov*. https://doi.org/10.1109/ICORAS.2017.8308068

Pangestu, Y., Setiyadi, D., & Khasanah, F. N. (2018). Metode Per Connection Classifier Untuk Implementasi Load Balancing Jaringan Internet. *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, *6*(1), 1–8. https://doi.org/10.33558/piksel.v6i1.1389

Widodo, P. P., Handayanto, R. T., & Herlawati. (2013). *Penerapan Data Mining dengan Matlab*. Informatika.

Zao, Z., Wang, J., & Liu, Y. (2017). User Electricity Behavior Analysis Based on K-Means Plus

Submitted: **14 Desember 2019**; Revised: **28 Desember 2019**; Accepted: **11 Januari 2020**; Published: **25 Januari 2020**

Clustering Algorithm. *International Conference on Computer Technology, Electronics and Communication (ICCTEC)*, 19-21 Dec. 2017. https://doi.org/10.1109/ICCTEC.2017.00111