

LAPORAN
MATA KULIAH ANALISIS DATA EKSPLORATIF (A)
“VISUALISASI DATA INTERAKTIF”



DISUSUN OLEH:

Reza Putri Angga (22083010006)

DOSEN PENGAMPU:

Tresna Maulana Fahrudin S.ST., M.T. (NIP. 199305012022031007)

PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR
2023

STUDI KASUS DAN PEMBAHASAN

Implementasikan dasbor visualisasi data interaktif, untuk memvisualisasikan dataset iris dengan beberapa menu, yakni :

- Variabel X dan Y : Sepal Length, Sepal Width, Petal Length, dan Petal Width (*Dropdown Button*)
- Jumlah Kelas : 3 (*Default*)

Dengan menggunakan *library* sklearn untuk melakukan *K-Means Clustering* dan di lakukan proses visualisasi interaktif dari dataset iris, di perlukan beberapa langkah-langkah, di antaranya yakni :

A. Melakukan Proses *Load* (Pembacaan) Data Dari Dataset Iris Yang Akan Di Analisis Lebih Lanjut

```
Melakukan Proses Load (Pembacaan) Data Dari Dataset Iris Yang Akan Di Analisis Lebih Lanjut

In [1]: import pandas as pd
        DataIris = pd.read_csv("iris.csv")
        DataIris

Out[1]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows x 5 columns

Pada kode script di atas, di lakukan proses *load* (pembacaan) dataset iris yang di peroleh dari github dengan url : [https://gist.github.com/curran/a08a1080b88344b0c8a7 - file-iris-csv](https://gist.github.com/curran/a08a1080b88344b0c8a7-file-iris-csv) dengan nama file “iris.csv” dengan menggunakan *library* pandas yang di permisalkan sebagai pd dan menggunakan fungsi “read_csv”. Kemudian, data-data dari dataset tersebut di simpan ke dalam dataframe dengan nama DataIris dan dapat di perlihatkan bahwa dataset iris yang akan di analisis lebih lanjut tersebut untuk pembuatan dasbor visualisasi data interaktif memiliki 150 baris dan 5 kolom.

B. Melakukan *Rename* (Pengubahan) Nama-Nama Kolom Pada Dataset

Melakukan Proses Rename (Pengubahan) Nama-Nama Kolom Pada Dataset

```

In [6]: #melakukan proses rename nama-nama kolom pada dataset
ColumnMapping = {"sepal_length" : "Sepal Length",
                 "sepal_width" : "Sepal Width",
                 "petal_length" : "Petal Length",
                 "petal_width" : "Petal Width",
                 "species" : "Species"}

DataIris.rename(columns = ColumnMapping, inplace = True)
DataIris

```

Out[6]:

	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows x 5 columns

Pada kode script di atas, di lakukan proses *rename* (pengubahan) nama-nama kolom dari dataset iris. Hal ini bertujuan agar lebih mempermudah proses analisis dengan melakukan visualisasi dengan adanya nama-nama kolom yang jelas dan lebih meningkatkan interpretasi yang nantinya akan di hasilkan. Terdapat variabel `ColumnMapping` yang di pergunakan untuk penulisan sebuah *directory* yang memetakan nama kolom yang ingin di ubah dengan nama baru. Seperti, nama kolom “sepal_length” ingin di ubah menjadi “Sepal Length”.

Kemudian di lakukan proses *rename* dengan menggunakan parameter *columns* untuk menentukan *mapping* dan argumen *inplace=True* untuk mengubah dataframe asli. Maka, akan di tampilkan nama-nama kolom dari dataset iris sesuai apa yang telah di tuliskan di variabel `ColumnMapping`.

C. Kode Script Untuk Dasbor Visualisasi Data Interaktif Dari Dataset Iris Dengan Menu Yang Di Tentukan

Kode Script Untuk Dasbor Visualisasi Data Interaktif Dari Dataset Iris Dengan Menu Yang Di Tentukan

```

In [20]: import warnings
warnings.filterwarnings("ignore")

```

Pada kode script di atas, di lakukan proses penggunaan *library* `warnings` yang di pergunakan untuk memanipulasi pemberitahuan atau peringatan yang di hasilkan dari output kode program. Dengan menggunakan *filterwarnings* argumen “ignore” sehingga akan mengabaikan atau tidak menampilkan peringatan apapun yang di hasilkan dari output kode program yang di buat. Selanjutnya terdapat kode program yang di pergunakan untuk

pembuatan dasbor visualisasi interaktif untuk dataset iris, yang akan di jelaskan lebih lanjut.

```
In [22]: #terdapat beberapa library yang dipergunakan, yakni:
#library pandas untuk manipulasi data yang telah di load sebelumnya
#library matplotlib untuk visualisasi plot
import matplotlib.pyplot as plt

#library sklearn untuk k-means clustering
from sklearn.cluster import KMeans

#library ipywidgets, widgets untuk pembuatan elemen interaktif
import ipywidgets as widgets

#library ipywidgets, interactive output untuk widgets yang merespon perubahan input
from ipywidgets import interactive_output

#library ipython, display untuk menampilkan output
from IPython.display import display

#dengan menggunakan dataset "iris.csv" yang disimpan kedalam variabel DataIris, dilakukan langkah-langkah:
#melakukan penghapusan kolom "species" karena hanya menggunakan kolom numerik
DataNumeric = DataIris.drop(columns = ["Species"])

#define function konversi label hasil clustering menjadi spesies
def ConvertLabelsToSpecies(Labels):
    LabelMapping = {0 : "setosa", 1 : "versicolor", 2 : "virginica"}
    return [LabelMapping[Label] for Label in Labels]

#define function untuk scatter plot
def ScatterPlot(X, Y, Labels):
    SpeciesLabels = ConvertLabelsToSpecies(Labels)
    ColorMapping = {"setosa" : "red", "versicolor" : "blue", "virginica" : "olive"} #peta warna untuk cluster 0,1,2
    Colors = [ColorMapping[Species] for Species in SpeciesLabels] #menggunakan peta warna untuk setiap label cluster

    plt.scatter(X, Y, c = Colors) #pembuatan scatter plot k-means clustering dengan sumbu x,y
    plt.xlabel(f"{X.name} (Cm)", fontsize = 14, fontname = "Times New Roman") #label sumbu x
    plt.ylabel(f"{Y.name} (Cm)", fontsize = 14, fontname = "Times New Roman") #label sumbu y
    plt.title("Scatter Plot Pengelompokan Dataset Iris Dengan K-Means", fontsize = 16,
             fontname = "Times New Roman") #label sumbu visualisasi

    #penambahan marker warna untuk visualisasi
    LegendMarkers = [plt.Line2D([0], [0], marker = "o", color = "w", markerfacecolor = ColorMapping[Species],
                                markersize = 10) for Species in ColorMapping.keys()]
    legend = plt.legend(LegendMarkers, ColorMapping.keys(), loc = "upper left", bbox_to_anchor = (1, 1),
                       title = "Species", title_fontsize = 10, fontsize = 10)
    for text in legend.get_texts():
        text.set_fontname("Times New Roman")

    plt.show()
```

```
#define function untuk k-means clustering
def KMeansClustering(Data, NClustering):
    KMeansModel = KMeans(n_clusters = NClustering) #objek model k-means dengan jumlah clustering yang ditentukan
    KMeansModel.fit(Data) #pelatihan dan pengelompokan data ke jumlah clustering yang ditentukan
    return KMeansModel.labels_ #jumlah keanggotaan untuk label cluster setiap data

#define function untuk perubahan visualisasi sesuai inputan
def UpdateScatterPlot(VariabelX, VariabelY, NClustering):
    X = DataNumeric[VariabelX] #inisiasi variabel x
    Y = DataNumeric[VariabelY] #inisiasi variabel y
    Labels = KMeansClustering(DataNumeric, NClustering) #perhitungan label clustering berdasarkan k-means
    ScatterPlot(X, Y, Labels) #pemanggilan fungsi untuk visualisasi scatter plot

#pembuatan menu pada dasbor untuk dropdown
OptionX = ["Sepal Length", "Sepal Width", "Petal Length", "Petal Width"]
OptionY = OptionX

#dropdown untuk variabel x,y, dan jumlah clustering
DropdownX = widgets.Dropdown(options = OptionX, value = "Sepal Length", description = "Variabel X : ",
                             style = {"description_width" : "120px"})
DropdownY = widgets.Dropdown(options = OptionY, value = "Sepal Width", description = "Variabel Y : ",
                             style = {"description_width" : "120px"})
DropdownNClustering = widgets.Dropdown(options = list(range(1, 4)), value = 3, description = "Jumlah Kelas Iris : ",
                                       style = {"description_width" : "120px"})

#pembuatan output interaktif
Output = interactive_output(UpdateScatterPlot, {"VariabelX": DropdownX, "VariabelY": DropdownY,
                                               "NClustering": DropdownNClustering})

#mengatur tata letak dashboard visualisasi
DashboardLayout = widgets.VBox([
    widgets.HTML("<h1 style='text-align:center;font-family:Times New Roman;'>Dasbor Visualisasi Interaktif Dataset",
    widgets.VBox([DropdownX, DropdownY, DropdownNClustering]), layout = widgets.Layout(align_items = "center")),
    Output], layout = widgets.Layout(align_items = "center"))

#menampilkan output dashboard visualisasi iris
display(DashboardLayout)
```

Pada kode script di atas, di lakukan proses pembuatan dasbor dari visualisasi data interaktif dari dataset iris dengan menu yang di tentukan. Di pergunakan beberapa *library* di antaranya, yakni *library* matplotlib untuk visualisasi *scatter plot* dari hasil K-Means, *library* sklearn.cluster untuk mengimpor dan menggunakan K-Means Clustering , *library* ipywidgets untuk membuat elemen interaktif dan *interactive_output* untuk menghubungkan fungsi interaktif tersebut dengan output yang di tampilkan, *library* IPython.display untuk mengimpor fungsi *display* yang menghasilkan output. Dengan menggunakan dataset iris yang telah di lakukan proses *load* dan *rename* sebelumnya, selanjutnya di lakukan penghapusan kolom “Species” hal ini di karenakan pada proses K-Means Clustering hanya mempergunakan kolom dengan tipe data numerik. Terdapat beberapa *define function* di antaranya, yakni :

➤ *Define function* ConvertLabelsToSpecies

Di pergunakan untuk mengkonversi label hasil clustering yang di tuliskan pada variabel LabelMapping menjadi nama spesies yang telah di tentukan agar lebih mudah untuk di lakukan analisis dan visualisasi lebih lanjut.

➤ *Define function* ScatterPlot

Di pergunakan untuk memberikan visualisasi dari hasil K-Means Clustering serta warna yang merepresentasikan spesies. Dengan variabel ColorMapping yang di pergunakan untuk membuat peta warna untuk setiap nama spesies, seperti setosa memiliki warna *red* atau merah sehingga visualisasi yang di hasilkan bisa sesuai dengan nama spesies yang telah di konversi dari label *clustering*. Lalu, di lakukan proses pembuatan visualisasi *scatter plot* menggunakan *library* matplotlib dengan sumbu x dan y dengan judul, label, dan marker dari setiap nama spesies sesuai dengan yang telah di buat. Di mana *font* yang akan di tampilkan menggunakan *font* times new roman.

➤ *Define function* KMeansClustering

Di pergunakan untuk melakukan proses K-Means Clustering pada dataset iris. Terdapat variabel KMeansModel yang di pergunakan untuk objek K-Means dengan jumlah *clustering* yang di tentukan oleh parameter NClustering kemudian akan di lakukan proses pengelompokkan data ke dalam jumlah yang telah di tentukan dan di tampilkan hasil *clustering* berupa label untuk setiap data.

➤ *Define function* UpdateScatterPlot

Di pergunakan untuk melakukan pembaruan hasil visualisasi interaktif *scatter plot* sesuai dengan input yang di lakukan. Dengan nilai X dan Y sesuai dengan inisiasi data dan Labels merupakan data yang di gunakan untuk *clustering* sejumlah NClustering.

Selanjutnya terdapat variabel OptionX dan OptionY yang berisi nama-nama variabel untuk sumbu x dan y yang di hasilkan pada visualisasi *scatter plot* dan di masukkan ke dalam *dropdown*. Kemudian terdapat variabel DropdownX yang di pergunakan untuk menampilkan opsi *dropdown* yang dapat di pilih sesuai dengan OptionX

dengan nama *dropdown* “Variabel X : “. Variabel *DropdownY* yang di pergunakan untuk menampilkan opsi *dropdown* yang dapat di pilih sesuai dengan *OptionY* dengan nama *dropdown* “Variabel Y : “. Dan, variabel *DropdownNClustering* yang di pergunakan untuk menampilkan opsi *dropdown* yang dapat di pilih dengan rentang kelas 1-3 (hal ini di karenakan pada dataset tersebut hanya terdapat 3 kelas spesies) dengan nama *dropdown* “Jumlah Kelas Iris : “. Variabel Output yang di pergunakan untuk menampilkan hasil atau ouput interaktif dari *define function* *UpdateScatterPlot* sesuai dengan pilihan *dropdown* yang telah di pilih. Dan terdapat variabel *DasborLayout* yang di pergunakan untuk mengatur tata letak dasbor. Kemudian, akan di tampilkan dasbor dari visualisasi data interaktif dari dataset iris.

Secara singkat pada kode script tersebut, di lakukan pembuatan dasbor visualisasi interaktif dari dataset iris menggunakan K-Means Clustering dengan menggunakan beberapa *library* yang telah di tuliskan. Dengan melakukan proses K-Means di antaranya, yakni penghapusan kolom spesies (*data preparation*), penerapan algoritma K-Means utuk membagi data ke dalam *cluster* sesuai dengan jumlah yang di pilih (*clustering*), dan visualisasi *scatter plot* yang merepresentasikan pengelompokkan data berdasarkan hasil K-Means Clustering. Di mana, dengan dasbor visualisasi data interaktif yang di hasilkan, bisa di lakukan pemilihan variabel untuk sumbu X, Y, dan jumlah *clustering* (jumlah kelas iris) dengan setiap perubahan pada inputan *dropdown* akan menghasilkan visualisasi *scatter plot* yang di perbarui secara interaktif.

D. Pilihan Dropdown Dari Dasbor Visualisasi Interaktif Data Dataset Iris

1. Dropdown Variabel X

Dasbor Visualisasi Interaktif Dataset Iris

Variabel X

✓
Sepal Length

Sepal Width

Petal Length

Petal Width

Variabel Y

Jumlah Kelas Iris

Pada tampilan *dropdown* variabel x tersebut dapat di perhatikan macam-macam pilihan *dropdown* yang dapat di pilih dan akan menghasilkan visualisasi sesuai dengan *dropdown* yang telah di pilih. Di mana macam-macam pilihan *dropdown* ini sebelumnya telah di tuliskan di dalam kode script variabel *OptionX* dan di masukkan ke dalam *DropdownX*.

2. Dropdown Variabel Y

Dasbor Visualisasi Interaktif Dataset Iris

Variabel X	Sepal Length	✓
Variabel Y	✓ Sepal Width	✓
Jumlah Kelas Iris	Petal Length	✓
	Petal Width	✓

Pada tampilan *dropdown* variabel y tersebut dapat di perlihatkan macam-macam pilihan *dropdown* yang dapat di pilih dan akan menghasilkan visualisasi sesuai dengan *dropdown* yang telah di pilih. Di mana macam-macam pilihan *dropdown* ini sebelumnya telah di tuliskan di dalam kode script variabel OptionY dan di masukkan ke dalam DropdownY.

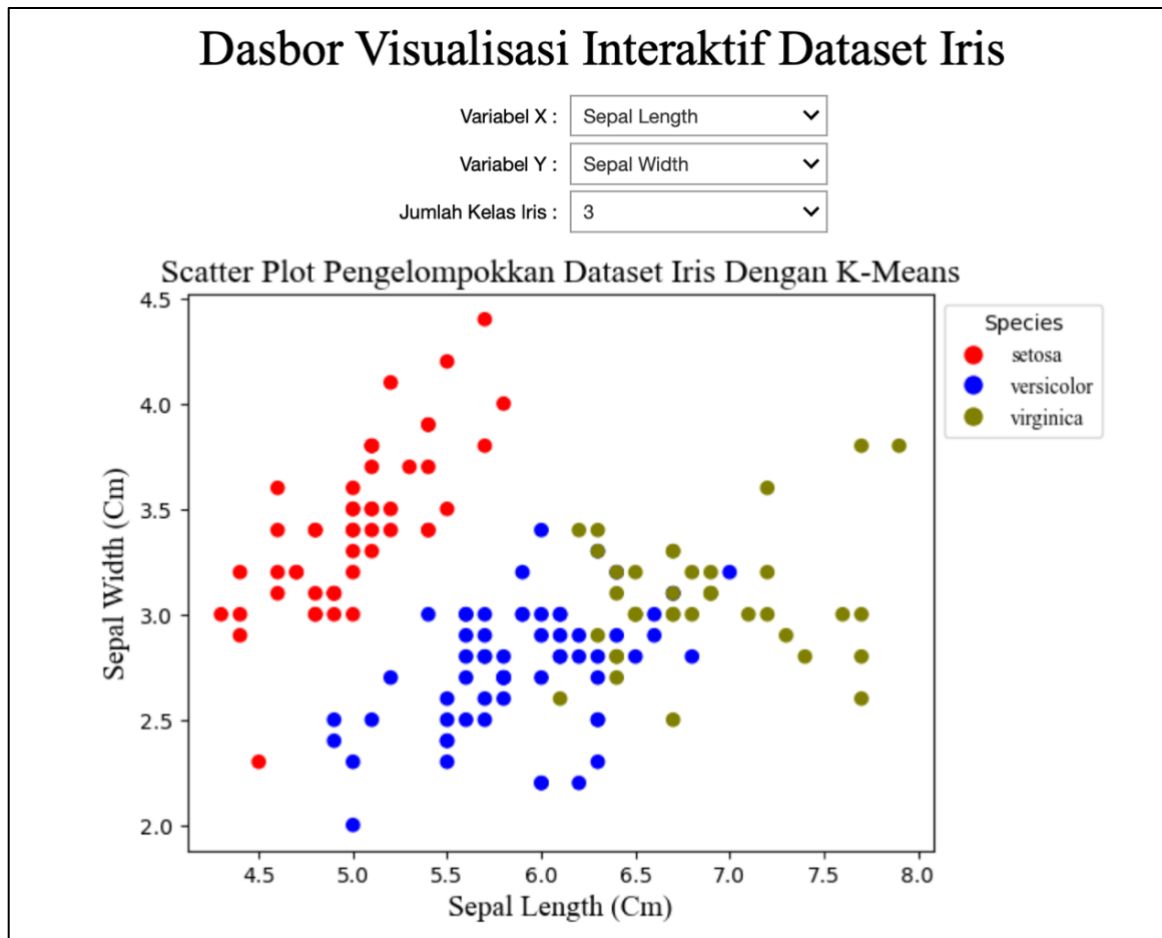
3. Dropdown Jumlah Kelas Iris

Dasbor Visualisasi Interaktif Dataset Iris

Variabel X	1	✓
Variabel Y	2	✓
Jumlah Kelas Iris	✓ 3	✓

Pada tampilan *dropdown* jumlah kelas iris tersebut dapat di perlihatkan macam-macam pilihan *dropdown* yang dapat di pilih dan akan menghasilkan visualisasi sesuai dengan *dropdown* yang telah di pilih. Di mana macam-macam pilihan *dropdown* ini sebelumnya telah di tuliskan di dalam kode script DropdownNClustering dengan range 1 sampai 3.

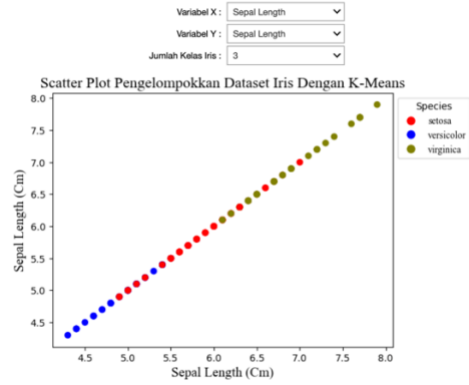
E. Output Visualisasi Interaktif Data Dari Dataset Iris



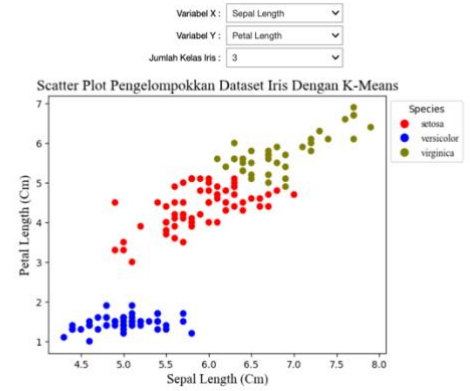
Dengan menampilkan dasbor visualisasi interaktif dataset iris, dan memilih sumbu x dengan Sepal Length (Panjang Kelopak) dan sumbu y dengan Sepal Width (Lebar Kelopak) dan dengan jumlah kelas iris 3. Di peroleh visualisasi *scatter plot* iris setosa di wakili oleh titik berwarna merah, iris versicolor di wakili oleh titik berwarna biru, dan iris virginica di wakili oleh titik berwarna olive atau hijau. Dapat di perhatikan, bahwa titik-titik berwarna merah (iris setosa) berada di area yang terpisah dari titik berwarna biru dan oliv. Hal ini menunjukkan bahwa iris setosa memiliki karakteristik yang berbeda dari iris versicolor dan virginica. Kemudian titik berwarna biru (iris versicolor) berada di area yang tumpang tindih dengan titik berwarna olive (iris virginica). Hal ini menunjukkan bahwa iris versicolor memiliki karakteristik yang mirip dengan iris virginica.

Hail visualisasi x dan y yang berbeda dapat di visualisasikan, di antaranya yakni :

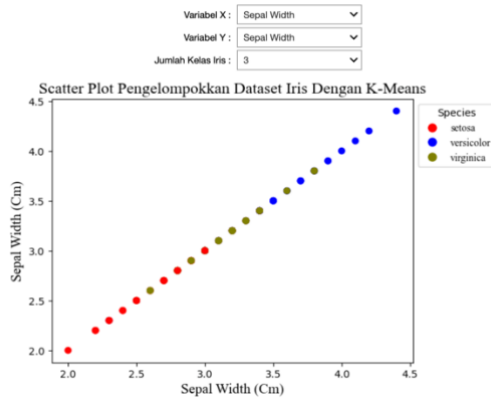
Dasbor Visualisasi Interaktif Dataset Iris



Dasbor Visualisasi Interaktif Dataset Iris



Dasbor Visualisasi Interaktif Dataset Iris



Dasbor Visualisasi Interaktif Dataset Iris

