LAPORAN

CASE BASED 2 UNSUPERVISED LEARNING

Disusun untuk memenuhi tugas

Mata Kuliah: Pembelajaran Mesin

Dosen Pengampu: BEDY PURNAMA, S.SI, MT, Ph.D (BDP)

Oleh:

MUHAMMAD RAFI ANDEO PRAJA (1301200278)

“Saya mengerjakan tugas ini dengan cara yang tidak melanggar aturan perkuliahan dan kode etik akademisi.”

KELAS IF-44-01
JURUSAN S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM

DAFTAR ISI

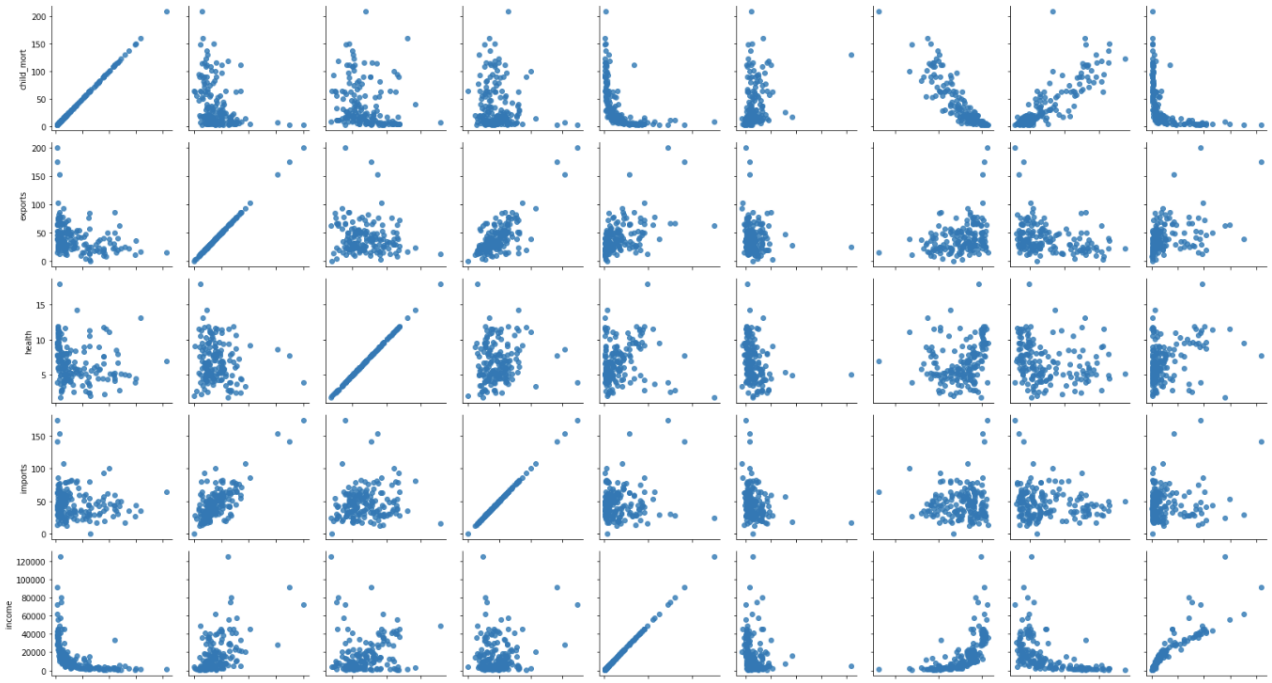
IKHTISAR

Tabel Dataset

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200

Dataset yang saya ambil berupa pengelompokkan beberapa negara yang dikategorikan berdasarkan faktor sosio-ekonomi dan faktor kesehatan yang akan menjadi penilaian perkembangan suatu negara. Dataset tersebut ditampilkan dengan jumlah data sebanyak 167 baris dengan 10 kolom feature yang masing – masing negara memiliki nilai yang berbeda. Dataset tersebut ditampilkan sudah dipasangkan dengan nama attribute atau features yang memudahkan saya dalam mengolah data tersebut yang nantinya akan diprediksi menggunakan algoritma dengan metode clustering. Dataset tersebut memiliki beragam nilai yang dimana terdapat satu nilai di setiap kolom tersebut bernominal tidak normal, atau dapat dikatakan sebagai outlier value. Berikut adalah tampilan grafik dari seluruh nominal yang terdapat pada dataset tersebut dengan pengelompokkan masing – masing features:

Grafik Value Features



Gambar grafik dapat dilihat pada link berikut:
https://drive.google.com/file/d/1HXRruuZE3HSyrOsMmvD1tKEryFPEJd66/view?usp=share_link

PRA-PEMROSESAN DATA

Sebelum masuk ke dalam tahapan pra – pemrosesan data, dataset tersebut harus diubah menjadi sebuah dataset yang layak dan dapat diproses di dalam algoritma yang dipilih. Tetapi dataset tersebut tidak hanya sudah dipasangkan dengan nama featuresnya, dataset tersebut juga tidak memiliki missing values atau value yang bernilai "?", jadi dapat saya simpulkan bahwa dataset tersebut dapat dibilang cukup layak untuk diproses.

Informasi Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   country     167 non-null    object 
 1   child_mort  167 non-null    float64
 2   exports     167 non-null    float64
 3   health      167 non-null    float64
 4   imports     167 non-null    float64
 5   income      167 non-null    int64  
 6   inflation   167 non-null    float64
 7   life_expec  167 non-null    float64
 8   total_fer   167 non-null    float64
 9   gdpp        167 non-null    int64  
dtypes: float64(7), int64(2), object(1)
memory usage: 13.2+ KB
```

```
df.isnull().any()
```

```
country       False
child_mort    False
exports       False
health        False
imports       False
income        False
inflation     False
life_expec    False
total_fer     False
gdpp          False
dtype: bool
```

Tetapi jika dilihat dari value setiap features pada grafik sebelumnya, terdapat beberapa outlier value yang bernominal tidak ekstrim. Maka dari itu saya menyimpulkan bahwa outlier value tersebut saya tetapkan sebagai normal value yang dimana layak untuk digunakan pada algoritma.

Program Scaling Data

```
pureData = data
data = pd.DataFrame(StandardScaler().fit_transform(data), columns = data.columns)
```

```
pureData.head()
```

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
4	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200

```
data.head()
```

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	1.291532	-1.138280	0.279088	-0.082455	-0.808245	0.157336	-1.619092	1.902882	-0.679180
1	-0.538949	-0.479658	-0.097016	0.070837	-0.375369	-0.312347	0.647866	-0.859973	-0.485623
2	-0.272833	-0.099122	-0.966073	-0.641762	-0.220844	0.789274	0.670423	-0.038404	-0.465376
3	2.007808	0.775381	-1.448071	-0.165315	-0.585043	1.387054	-1.179234	2.128151	-0.516268
4	-0.695634	0.160668	-0.286894	0.497568	0.101732	-0.601749	0.704258	-0.541946	-0.041817

Teknik pra – pemrosesan data yang saya gunakan adalah melakukan reduksi berdasarkan transformasi, dengan pengubahan nilai value berdasarkan range tertentu. Teknik tersebut dinamakan proses scaling yang dimana saya menggunakan proses tersebut untuk mengubah bentuk dataset supaya dapat digunakan di dalam fungsi algoritma nanti. Teknik pra – pemrosesan data reduksi berdasarkan transformasi yaitu scaling, lebih layak digunakan karena hanya mengubah value dari data tersebut yang nantinya dapat digunakan di algoritma pembelajaran mesin nanti, dengan scaling yang saya gunakan adalah scaling berdasarkan Min – Max Normalization dengan rumus berikut:

Rumus Min – Max Normalization

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$$

ALGORITMA UNSUPERVISED LEARNING

Program Clustering K-Means

```
class KMeans:

    def __init__(self, k, x, max_iterations):

        self.k = k
        self.max_iterations = max_iterations
        self.num_examples, self.num_features = x.shape
        self.plot_figure = True

    def initialize_random_centroids(self, x):

        centroids = np.zeros((self.k, self.num_features))

        for i in range(self.k):
            centroid = x[np.random.choice(range(self.num_examples))]
            centroids[i] = centroid

        return centroids

    def create_cluster(self, x, centroids):

        clusters = [[] for i in range(self.k)]

        for point_index, point in enumerate(x):
            closest_centroid = np.argmin(np.sqrt(np.sum((point-centroids)**2, axis=1)))
            clusters[closest_centroid].append(point_index)

        return clusters
```

```
def calculate_new_centroids(self, cluster, x):

    centroids = np.zeros((self.k, self.num_features))

    for index, cluster in enumerate(cluster):
        new_centroid = np.mean(x[cluster], axis=0)
        centroids[index] = new_centroid

    return centroids

def predict_cluster(self, clusters, x):

    y_pred = np.zeros(self.num_examples)

    for cluster_index, cluster in enumerate(clusters):
        for sample_index in cluster:
            y_pred[sample_index] = cluster_index

    return y_pred

def plot_fig(self, x, y):

    fig = px.scatter(x[:, 0], x[:, 1], color=y)
    fig.show()
```

```
    def fit(self, x):

        centroids = self.initialize_random_centroids(x)

        for i in range(self.max_iterations):
            clusters = self.create_cluster(x, centroids)
            previous_centroids = centroids
            centroids = self.calculate_new_centroids(clusters, x)
            diff = centroids - previous_centroids
            if not diff.any():
                break

        y_pred = self.predict_cluster(clusters, x)
        if self.plot_figure:
            self.plot_fig(x, y_pred)

        return y_pred

#source = https://www.kaggle.com/code/adinishad/kmeans-clustering-from-scratch#KMeans-From-Scratch
```

Saya menggunakan algoritma K-Means sebagai algoritma yang akan mengelompokkan atau melakukan clustering dari dataset yang diproses. Menurut saya, algoritma K-Means adalah algoritma unsupervised learning yang menggunakan rumus perhitungan jarak yang akan digunakan untuk menentukan jarak cluster atau K, saya menggunakan rumus perhitungan jarak Euclidean Distance dengan rumus berikut:

Rumus Euclidean Distance

Menghitung perhitungan jarak menggunakan formula euclidean distance

$$Euclidean\,(A,B) = \sqrt{\sum_{i=1}^{N} (fa_i - fb_i)^2}$$

Rumus tersebut akan menghitung point A dan point B dengan melakukan penjumlahan atau sum dari point A dikurang point B dikuadratkan dan di akarkan. Rumus tersebut saya gunakan dalam salah satu fungsi program K-Means yaitu membuat cluster atau K. Berikut penjelasan algoritma K-Means tersebut berdasarkan beberapa function atau fungsi yang tersedia.

1. Class K-Means memiliki function inisialisasi self yang dimana kelas tersebut memiliki 3 attribute yaitu, k sebagai cluster, max iterations sebagai banyaknya iterasi program dijalankan untuk menemukan k atau cluster yang terbaik, examples dan features atau baris dan kolom dari dataset akan dimasukkan kedalam kelas tersebut, dan plot figure sebagai gambar hasil plot dari K-Means.

2. Sebelum terbentuknya sebuah cluster atau k, kita harus melakukan inisialisasi centroids. Centroids merupakan titik tengah yang akan mengelompokkan cluster atau k. Pembuatan centroids dapat dilakukan secara acak atau random dalam range baris dataset atau examples.

3. Setelah terbentuknya centroids, maka kita dapat membuat cluster atau k yang dimana jarak antar cluster ditentukan menggunakan rumus euclidean distance tersebut.

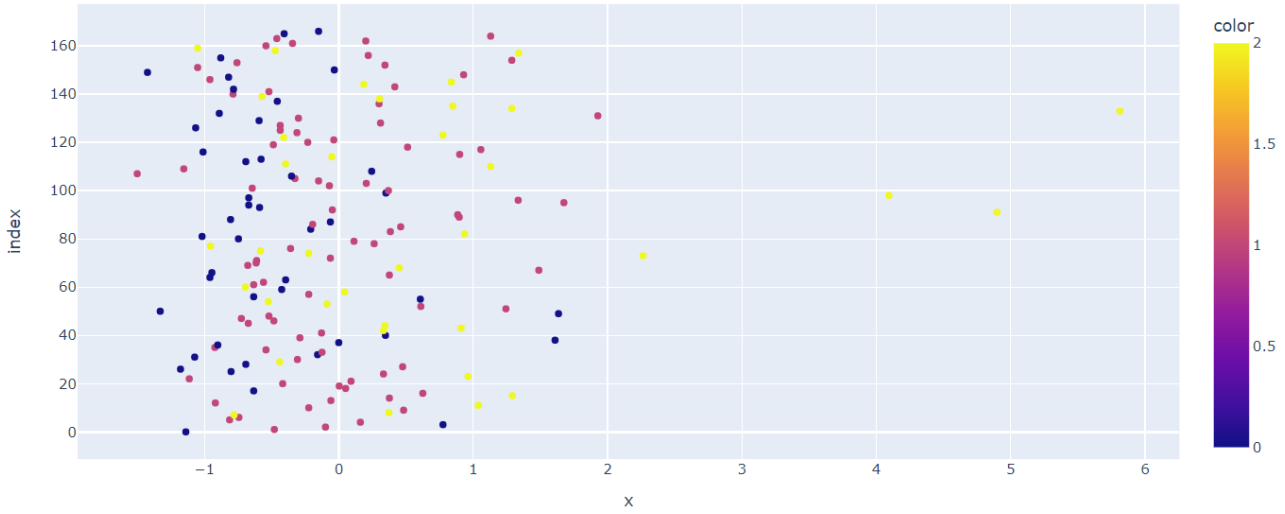
4. Sebelum dilakukannya fitting data, centroids yang sebelumnya sudah dibentuk harus dihitung berapa jumlah centroids yang kita miliki yang nantinya akan digunakan pada proses fitting data.

5. Setelah cluster atau k terbentuk, kita dapat membuat prediksi cluster atau k yang akan digunakan pada proses fitting data dengan melakukan iterasi semua cluster yang sudah dibuat.

6. Function atau fungsi plotting dapat dikatakan sebagai fungsi tambahan yang berfungsi untuk membentuk scatter plot hasil K-Means.

7. Terakhir adalah fungsi fitting data yang dimana memanggil beberapa fungsi yang nantinya akan bekerja dalam membuat centroids untuk digunakan dalam pembuatan cluster atau k, jika centroids tersebut saat di inisialisasi lagi memiliki value yang berbeda maka akan mengubah jarak cluster atau k sampai jarak tersebut dapat dikatakan jarak yang terbaik atau jarak yang tidak berubah berdasarkan perhitungan jarak sebelumnya. Jarak tersebut dapat dilihat dari value centroids yang berubah atau tidaknya berdasarkan pemilihan centroids secara acak. Jika jarak sudah dikatakan terbaik dengan value centroidsnya, maka cluster tersebut dapat diprediksi dan dibuatkan scatter plot hasil K-Means.

Scatter Plot Hasil K-Means




```
array([0., 1., 1., 0., 1., 1., 1., 2., 2., 1., 1., 2., 1., 1., 1., 2., 1.,
       0., 1., 1., 1., 1., 1., 2., 1., 0., 0., 1., 0., 2., 1., 0., 0., 1.,
       1., 1., 0., 0., 0., 1., 0., 1., 2., 2., 2., 1., 1., 1., 1., 0., 0.,
       1., 1., 2., 2., 0., 0., 1., 2., 0., 2., 1., 1., 0., 0., 1., 0., 1.,
       2., 1., 1., 1., 1., 2., 2., 2., 1., 2., 1., 1., 0., 0., 2., 1., 0.,
       1., 1., 0., 0., 1., 1., 2., 1., 0., 0., 1., 1., 0., 2., 0., 1., 1.,
       1., 1., 1., 1., 0., 1., 0., 1., 2., 2., 0., 0., 2., 1., 0., 1., 1.,
       1., 1., 1., 2., 2., 1., 1., 0., 1., 1., 0., 1., 1., 0., 2., 2., 2.,
       1., 0., 2., 2., 1., 1., 0., 1., 2., 2., 1., 0., 1., 0., 0., 1., 1.,
       1., 1., 0., 1., 2., 2., 2., 1., 1., 1., 1., 1., 0., 0.])
```

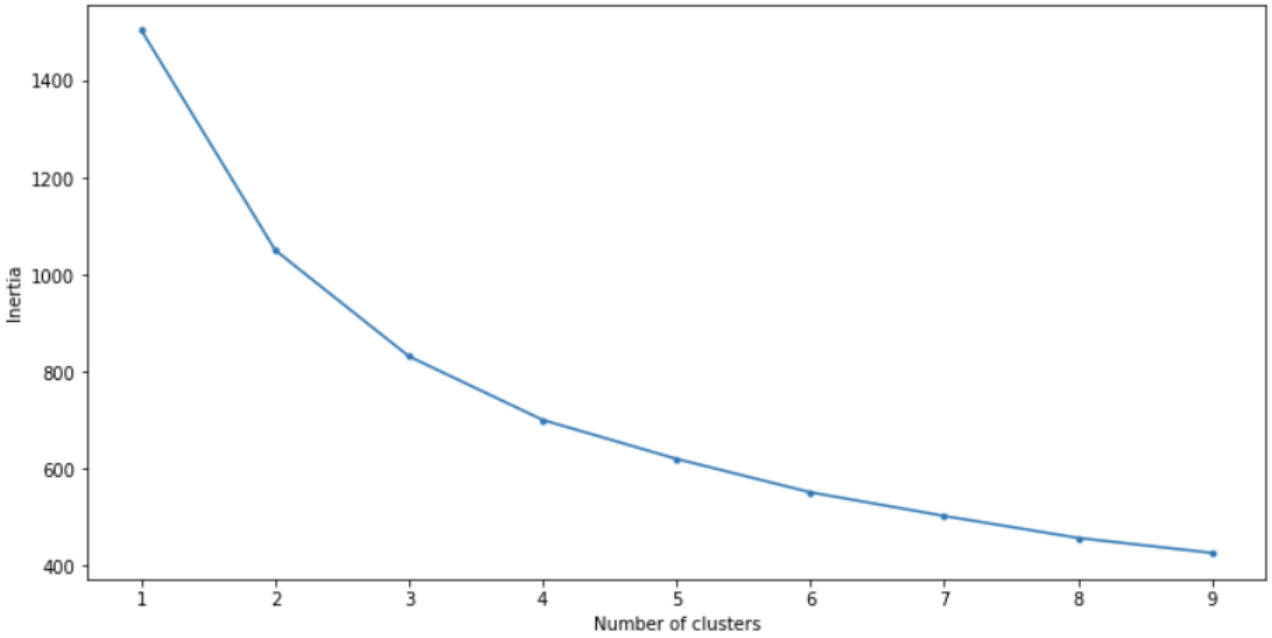
Berdasarkan hasil scatter plot tersebut, dapat disimpulkan bahwa program K-Means tersebut hanya mengelompokkan data berdasarkan cluster tetapi tidak mengumpulkan setiap letak data berdasarkan clusternya masing – masing. Terdapat 3 value dengan warna kuning bernilai cluster atau k 2 tersebar sangat jauh dari value lainnya yang dapat disimpulkan bahwa penggunaan algoritma K-Means dapat dikatakan kurang cocok dengan dataset tersebut. Dikarenakan penentuan centroids berdasarkan pemilihan acak atau random, maka nilai cluster atau k setiap value dapat berubah berdasarkan centroids yang dipilih secara acak.

Terdapat sebuah pertanyaan mengenai nilai cluster atau k yang terbentuk, apakah nilai tersebut sudah menjadi nilai yang terbaik untuk dataset tersebut? Untuk menjawab pertanyaan tersebut diperlukannya sebuah evaluasi yang dimana evaluasi tersebut akan menentukan nilai cluster atau k berapa yang menjadi terbaik. Algoritma tersebut dibangun tanpa menggunakan library pembelajaran mesin yaitu sklearn, tetapi penggunaan sklearn akan saya gunakan hanya dalam tahapan pra – pemrosesan data dan evaluasi algoritma saja.

Source: https://www.kaggle.com/code/adinishad/kmeans-clustering-from-scratch#KMeans-From-Scratch

EVALUASI

Grafik Elbow Method



Teknik evaluasi yang saya gunakan adalah elbow method yang dimana menampilkan nilai clusters atau k apa yang terbaik yang digunakan dalam algoritma. Teknik tersebut memerlukan nilai inertia dari prediksi data yang dimana saya menggunakan library sklearn hanya untuk mengambil nilai tersebut. Karena dalam grafik terdapat patahan garis dari 2 ke 3 yang cukup besar, maka dapat disimpulkan bahwa clusters dengan nilai 3 merupakan nilai clusters yang terbaik dalam K-Means dengan data tersebut. Perhitungan jarak garis tersebut dapat menggunakan variabel SSE (Sum Square Error) yang menghitung nilai cluster dalam perhitungan jarak dengan rumus berikut:

Rumus Sum Square Error

$$SSE = \sum_{j=1}^{K} \sum_{X_i \in C_j} \left\| c_j - x_i \right\|_2^2$$

Kesimpulan yang dapat saya berikan adalah penggunaan algoritma K-Means merupakan salah satu algoritma unsupervised learning yang dimana tidak menggunakan data testing yang dapat menjadi acuan dalam pengembangan prediksi dataset. Penggunaan algoritma K-Means juga memerlukan K sebagai cluster yang dimana membagi beberapa tipe dari dataset yang sudah dilakukan prediksi. Nilai K dapat dievaluasi melalui salah satu cara evaluasi algoritma unsupervised learning menggunakan elbow method yang menampilkan nilai K terbaik berdasarkan acuan miringnya sebuah garis di suatu titik clusters atau titik nilai K. Saya menyimpulkan bahwa algoritma K-Means dapat dikatakan kurang baik dalam mengolah dataset tersebut dengan hasil plotting yang masih bersebaran atau tidak dikelompokkan.

Penugasan Case Based 2 berikut menjadi sebuah salah satu kesempatan bagi saya dalam mempelajari salah satu algoritma unsupervised learning dan juga kesempatan dalam mempelajari bagaimana cara melakukan teknik pra – pemrosesan data terhadap dataset yang masih raw atau tidak layak untuk digunakan. Maka dari itu, saya atas nama Muhammad Rafi Andeo Praja mengucapkan banyak rasa syukur dan rasa terima kasih kembali kepada Bapak Bedy Purnama selaku dosen pengampu mata kuliah Pembelajaran Mesin dan seluruh Tim Dosen Pembelajaran Mesin S1 Informatika tahun ajaran 2022/2023. Saya sangat terbuka dalam menerima segala saran dan kritikan terhadap penugasan Case Based 2 ini, dan saya tau bahwa penugasan Case Based 2 ini masih sangat jauh dari kata sempurna dibandingkan dengan penugasan Case Based 1 dari segi laporan dan kode program. Saya berharap bahwa untuk penugasan berikutnya atau penugasan terakhir dalam mata kuliah pembelajaran mesin atau mata kuliah lainnya, saya dapat melatih diri saya kembali dalam pembuatan laporan yang baik dan pembuatan kode program yang baik.

Link Kode Program:

Github:
https://github.com/RafiAndeo/KMeans_Algorithm

Google Colab:
https://colab.research.google.com/drive/1G3xxvhi9L6utELZthjiviMqS_crYHuHY?usp=sharing

Link Video Presentasi:
https://drive.google.com/file/d/1sZRHHaV5ajPuAMbB2gCc4gFjpmkt4MTi/view?usp=share_link

Link Slide Presentasi:
https://docs.google.com/presentation/d/1X4RD9uJq2MyRY8zH6UqxcMJs2lnx8je2/edit?usp=share_link&ouid=116705627032248410275&rtpof=true&sd=true

DAFTAR PUSTAKA

Dios Kurniawan. 2020. Pengenalan Machine Learning dengan Python. Jakarta: PT Elex Media Komputindo

Aditta Das Nishad. 2021. KMeans Clustering From Scratch.
https://www.kaggle.com/code/adinishad/kmeans-clustering-from-scratch