

```
# Impor library
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Membaca data
data = pd.read_csv('/content/drive/MyDrive/Adv. ML/obe-pca.csv')
data.head()
```

ajaran_Lulusan	Pemetaan	Bahan_Kajian	Mata_Kuliah	Susunan	Organisasi	Peta_Pemei
9	7	8	9	7	6	
9	7	7	8	8	7	
9	8	8	9	9	8	
7	6	6	7	6	6	
8	7	7	8	7	7	

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Konsep                                5 non-null     int64
1   Kerangka                             5 non-null     int64
2   Profil_Lulusan                        5 non-null     int64
3   Capaian_Pembelajaran_Lulusan         5 non-null     int64
4   Pemetaan                             5 non-null     int64
5   Bahan_Kajian                         5 non-null     int64
6   Mata_Kuliah                          5 non-null     int64
7   Susunan                              5 non-null     int64
8   Organisasi                           5 non-null     int64
9   Peta_Pemenuhan                       5 non-null     int64
10  Metode_Penilaian                     5 non-null     int64
11  Tahapan_Penilaian                    5 non-null     int64
12  Bobot_Penilaian                      5 non-null     int64
13  Rumusan_Nilai_Akhir                  5 non-null     int64
dtypes: int64(14)
memory usage: 688.0 bytes
```

```
# Menghilangkan kolom yang tidak diperlukan
# data = data.drop('kolom_tidak_diperlukan', axis=1)
```

```
# Melakukan standardisasi data
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
```

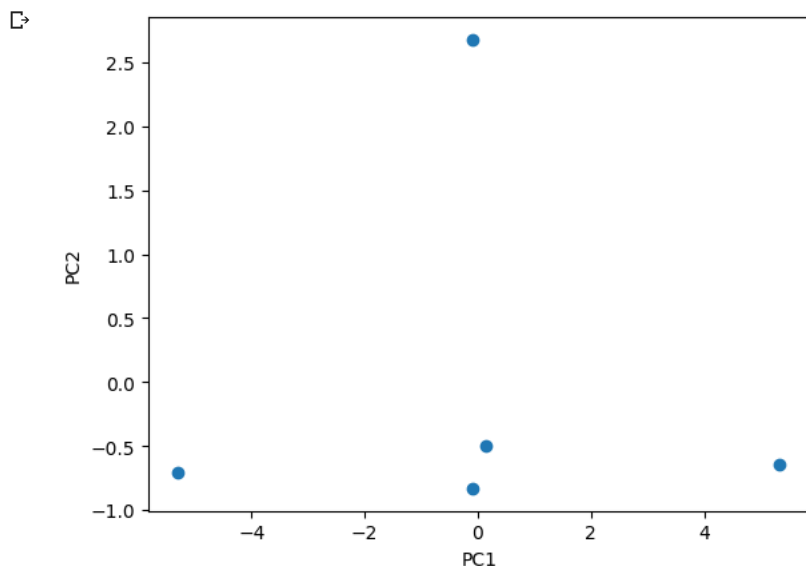
```
# Menentukan jumlah faktor yang diinginkan
n_components = 5
# Membuat objek PCA dengan jumlah faktor yang diinginkan
pca = PCA(n_components=n_components)
# Melakukan PCA pada data yang sudah di-standardisasi
pca.fit(data_scaled)
```

```
PCA
PCA(n_components=5)
```

```
pca_data = pca.transform(data_scaled)
```

```
import matplotlib.pyplot as plt
# membuat scatter plot
plt.scatter(pca_data[:,0], pca_data[:,1])
plt.xlabel('PC1')
plt.ylabel('PC2')
```

```
plt.show()
```



```
# Membuat dataframe baru dari hasil PCA
```

```
data_pca = pd.DataFrame(pca.transform(data_scaled), columns=['Faktor 1', 'Faktor 2', 'Faktor 3', 'Faktor 4', 'Faktor 5'])
```

```
# Melihat kontribusi tiap variabel terhadap setiap faktor
```

```
loadings = pd.DataFrame(pca.components_.T, index=data.columns, columns=['Faktor 1', 'Faktor 2', 'Faktor 3', 'Faktor 4', 'Faktor 5'])
```

```
# Menampilkan hasil PCA dan loadings
```

```
print('Hasil PCA:\n', data_pca.head())
```

```
print('\nLoadings:\n', loadings)
```

Hasil PCA:

	Faktor 1	Faktor 2	Faktor 3	Faktor 4	Faktor 5
0	-0.102896	2.678155	0.116622	-0.029510	9.172962e-17
1	-0.091710	-0.831456	1.607479	0.112945	5.009626e-17
2	-5.294496	-0.703208	-0.493581	-0.365174	-3.957336e-16
3	5.339925	-0.646001	-0.437370	-0.379159	-1.582219e-16
4	0.149177	-0.497490	-0.793150	0.660898	3.415298e-16

Loadings:

	Faktor 1	Faktor 2	Faktor 3	Faktor 4	Faktor 5
Konsep	-0.277807	0.208408	-0.116004	-0.513116	0.273895
Kerangka	-0.277613	-0.172995	0.283250	-0.321112	0.513808
Profil_Lulusan	-0.248700	-0.405100	-0.063076	0.079890	0.219664
Capaian_Pembelajaran_Lulusan	-0.239209	0.247903	0.569387	0.167382	-0.136632
Pemetaan	-0.297140	-0.010024	-0.024273	0.030394	-0.343538
Bahan_Kajian	-0.253559	0.388156	0.022047	-0.028515	-0.008066
Mata_Kuliah	-0.253559	0.388156	0.022047	-0.028515	0.192239
Susunan	-0.277613	-0.172995	0.283250	-0.321112	-0.490730
Organisasi	-0.248700	-0.405100	-0.063076	0.079890	0.018843
Peta_Pemenuhan	-0.275222	0.154344	-0.328411	0.377660	0.059468
Metode_Penilaian	-0.248964	0.114664	-0.607168	-0.181767	-0.154511
Tahapan_Penilaian	-0.248700	-0.405100	-0.063076	0.079890	0.018843
Bobot_Penilaian	-0.297140	-0.010024	-0.024273	0.030394	-0.343538
Rumusan_Nilai_Akhir	-0.288115	0.066599	0.106243	0.551527	0.228187

```
# membuat scatter plot
```

```
plt.scatter(pca_data[:,0], pca_data[:,1])
```

```
plt.xlabel('PC1')
```

```
plt.ylabel('PC2')
```

```
plt.show()
```

✓ 1s completed at 3:59 PM

● ×