Kode program yang akan diajarkan di sini bisa diunduh di tautan [berikut ini](https://colab.research.google.com/drive/1oRqkAWXBId10klYnOt-R27th1DsQExQp?usp=sharing). Untuk membukanya upload berkasnya ke Google Colab.

**Tujuan**

Pada latihan ini, kita akan berlatih mengimplementasi PCA dengan library SKLearn.

**Tahapan Latihan**

Tahapan pada latihan ini sebagai berikut:

1. Bagi dataset.
2. Latih model tanpa PCA.
3. Latih model dengan PCA.
4. Evaluasi hasil kedua model.

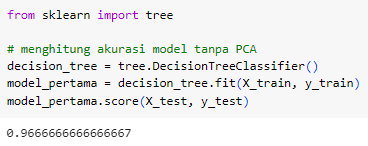
**Codelab**

Pada Colaboratory impor library yang dibutuhkan.  Kemudian kita masukkan dataset iris dan bagi data menjadi train set dan test set.

1. from sklearn import datasets
2. from sklearn.model\_selection import train\_test\_split
4. iris = datasets.load\_iris()
5. atribut = iris.data
6. label = iris.target
8. # bagi dataset menjadi train set dan test set
9. X\_train, X\_test, y\_train, y\_test = train\_test\_split(
10. atribut, label, test\_size=0.2, random\_state=1)

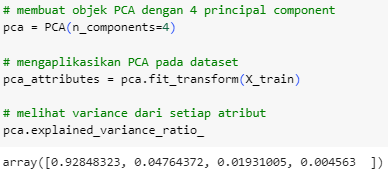
Kita akan menggunakan model Decision Tree dan menghitung berapa akurasinya tanpa menggunakan PCA. Akurasi tanpa PCA adalah 0.9666. Akurasi dari model Anda mungkin berbeda dengan keluaran di bawah.

1. from sklearn import tree
3. decision\_tree = tree.DecisionTreeClassifier()
4. model\_pertama = decision\_tree.fit(X\_train, y\_train)
5. model\_pertama.score(X\_test, y\_test)

Tampilan hasil akurasi tanpa PCA dari kode di atas sebagai berikut.  
[](https://www.dicoding.com/academies/184/tutorials/8437?from=8432)

Kemudian kita akan menggunakan PCA dan menghitung variance dari setiap atribut. Hasilnya adalah 1 atribut memiliki variance sebesar 0.922, yang berarti atribut tersebut menyimpan informasi yang tinggi dan jauh lebih signifikan dari atribut lain.

1. from sklearn.decomposition import PCA
3. # membuat objek PCA dengan 4 principal component
4. pca = PCA(n\_components=4)
6. # mengaplikasikan PCA pada dataset
7. pca\_attributes = pca.fit\_transform(X\_train)
9. # melihat variance dari setiap atribut
10. pca.explained\_variance\_ratio\_

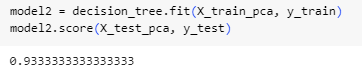
Hasil dari setiap atributnya menjadi sebagai berikut.  
[](https://www.dicoding.com/academies/184/tutorials/8437?from=8432)

Melihat dari variance sebelumnya kita bisa mengambil 2 principal component terbaik karena total variance nya adalah 0.976 yang sudah cukup tinggi.

1. # PCA dengan 2 principal component
2. pca = PCA(n\_components = 2)
3. X\_train\_pca = pca.fit\_transform(X\_train)
4. X\_test\_pca = pca.fit\_transform(X\_test)

Kita akan menguji akurasi dari classifier setelah menggunakan PCA.

1. # uji akurasi classifier
2. model2 = decision\_tree.fit(X\_train\_pca, y\_train)
3. model2.score(X\_test\_pca, y\_test)

Hasil pengujian akurasi setelah menggunakan PCA menjadi seperti di bawah ini.  
[](https://www.dicoding.com/academies/184/tutorials/8437?from=8432)

Dari percobaan di atas bisa kita lihat bahwa dengan hanya 2 principal component atau 2 atribut saja model masih memiliki akurasi yang tinggi. Dengan principal component kamu bisa mengurangi atribut yang kurang signifikan dalam prediksi dan mempercepat waktu pelatihan sebuah model machine learning.