Kode program yang akan diajarkan di sini bisa diunduh di tautan [berikut ini](https://colab.research.google.com/drive/17bvLmMe_hnOWmFZKarH2mWIpYh4tcyPX?usp=sharing). Silahkan buat salinan dari kode ini dan jalankan di Google Colab Anda.

**Tujuan**

Sekarang saatnya kita mencoba menggunakan model SVM yang tersedia di library SKLearn. Untuk latihan kali ini kita akan memakai dataset Pima Indian. Dataset ini dikumpulkan oleh *National Institute of Diabetes and Digestive and Kidney Diseases*.

**Tahapan Latihan**

Dataset berisi 8 kolom atribut dan 1 kolom label yang berisi 2 kelas yaitu 1 dan 0. Angka 1 menandakan bahwa orang tersebut positif diabetes dan 0 menandakan sebaliknya. Terdapat 768 sampel yang merupakan 768 pasien perempuan keturunan suku Indian Pima.

Model machine learning yang akan kita buat bertujuan untuk mengklasifikasikan apakah seorang pasien positif diabetes atau tidak.

Tahapan latihan kali ini adalah:

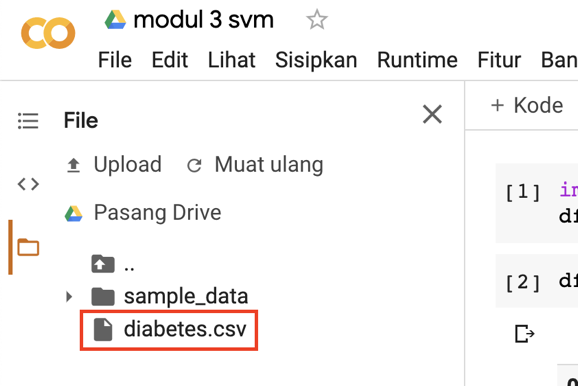
1. Ubah data ke dalam Dataframe.
2. Bagi dataset.
3. Lakukan standarisasi.
4. Buat dan latih model.
5. Evaluasi model.

**Codelab**

Tahap pertama yang perlu kita lakukan adalah mengunduh dataset **Pima Indian Diabetes** dari [tautan](https://www.kaggle.com/uciml/pima-indians-diabetes-database) berikut. Setelah mengunduh dataset, jangan lupa masukkan dataset ke dalam Google Colab.

Pada tahap selanjutnya kita akan mengimpor **library pandas** dan mengubah dataset menjadi sebuah dataframe.

Jika Anda menggunakan Colaboratory, dataset yang telah berhasil diunggah ke Colab akan tampil sebagai berikut.

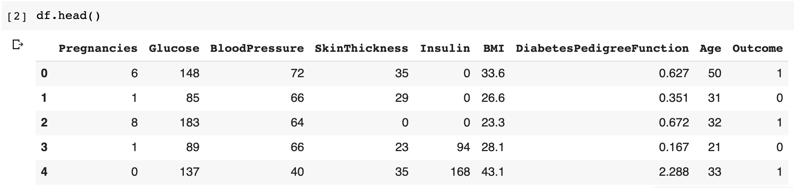


Kemudian, impor *library*pandas dan ubah file csv menjadi dataframe dengan *code*berikut.

1. import pandas as pd
3. df = pd.read\_csv('diabetes.csv')

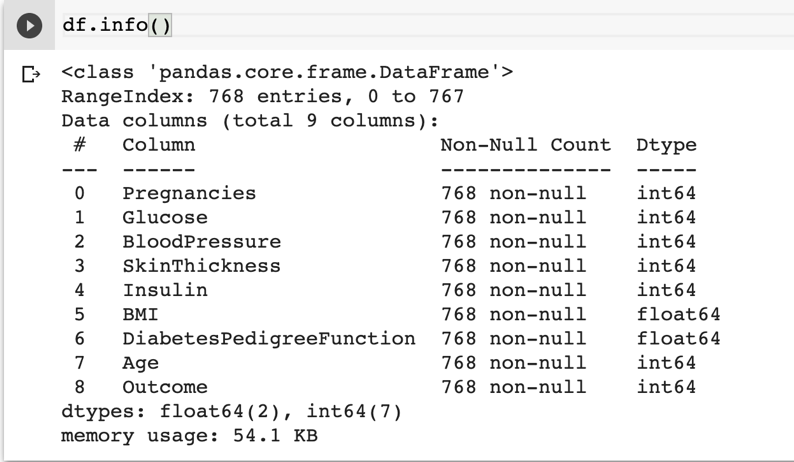
Lalu kita tampilkan 5 baris teratas dari dataframe untuk melihat isi dari dataset. Untuk melakukannya kita dapat menjalankan kode df.head() seperti di bawah.

1. df.head()



Hal paling penting selanjutnya adalah kita perlu mengecek apakah terdapat nilai-nilai yang hilang pada dataset serta apakah ada atribut yang bukan berisi bilangan numerik. Kita bisa melakukan ini dengan memanggil fungsi .info() pada dataframe.

1. df.info()



Output dari fungsi info() menunjukkan bahwa semua atribut nilainya lengkap, dan juga nilai-nilai dari tiap kolom memiliki tipe data numerik yaitu **int64** dan **float64**.

Pada tahap ini data sudah bisa dipakai untuk pelatihan model.

Kita lalu memisahkan antara atribut dan label pada dataframe. Untuk memisahkan kolom-kolom pada dataframe kamu bisa melihat dokumentasinya pada [tautan](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html) ini.

1. # memisahkan atribut pada dataset dan menyimpannya pada sebuah variabel
2. X = df[df.columns[:8]]
4. # memisahkan label pada dataset dan menyimpannya pada sebuah variabel
5. y = df['Outcome']

Jika kita lihat, nilai-nilai pada dataset memiliki skala yang berbeda. Contohnya pada kolom *Glucose* dan kolom *Diabetes Pedigree Function*. Kita perlu mengubah nilai-nilai dari setiap atribut berada pada skala yang sama. Kita dapat mencoba menggunakan standarisasi dengan fungsi *StandardScaler()* dari SKLearn.

1. from sklearn.preprocessing import StandardScaler
3. # standarisasi nilai-nilai dari dataset
4. scaler = StandardScaler()
5. scaler.fit(X)
6. X = scaler.transform(X)

Setelah atribut dan label dipisah, kita bisa memisahkan data untuk training dan testing menggunakan fungsi *.train\_test\_split()*.

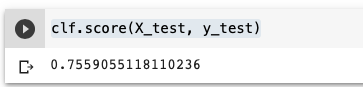
1. from sklearn.model\_selection import train\_test\_split
3. # memisahkan data untuk training dan testing
4. X\_train, X\_test, y\_train, y\_test = train\_test\_split(
5. X, y, test\_size=0.33, random\_state=42)

Kita kemudian membuat objek *Support Vector Classifier* dan menampungnya pada variabel clf. Akhirnya kita sampai pada tahapan yang kita tunggu-tunggu, kita memanggil fungsi fit untuk melatih model.

1. from sklearn.svm import SVC
3. # membuat objek SVC dan memanggil fungsi fit untuk melatih model
4. clf = SVC()
5. clf.fit(X\_train, y\_train)

Terakhir, kita bisa melihat bagaimana akurasi prediksi dari model yang kita latih terhadap data testing.

1. # Menampilkan skor akurasi prediksi
2. clf.score(X\_test, y\_test)



Selamat, Anda telah berhasil mengembangkan sebuah model Support Vector Classifier untuk mendeteksi diabetes.