```
#1. Menentukan Library yang digunakan
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
```

```
#2. Load Dataset
diabetes_dataset = pd.read_csv('diabetes.csv')
```

```
diabetes_dataset.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
diabetes_dataset.shape
```

```
(768, 9)
```

```
diabetes_dataset['Outcome'].value_counts()
```

Outcome	count
0	500
1	268

```
#Memisahkan data dan label
X = diabetes_dataset.drop(columns='Outcome', axis=1)
Y = diabetes_dataset['Outcome']
```

```
print(X)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age
0                       0.627   50
1                       0.351   31
2                       0.672   32
3                       0.167   21
4                       2.288   33
..                        ...  ...
763                     0.171   63
764                     0.340   27
765                     0.245   30
766                     0.349   47
767                     0.315   23

[768 rows x 8 columns]
```

```
print(Y)
```

```
0      1
1      0
2      1
3      0
```

```
4      1
      ..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
```

```
#Memisahkan data training dan data testing
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(768, 8) (614, 8) (154, 8)
```

```
#Membangun model menggunakan SVM
classifier = svm.SVC(kernel='linear')
```

```
classifier.fit(X_train, Y_train)
```

```
SVC
SVC(kernel='linear')
```

```
#Membuat model evaluasi untuk mengukur tingkat akurasi
X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
print('Akurasi data training adalah = ', training_data_accuracy)
```

```
Akurasi data training adalah =  0.7833876221498371
```

```
X_test_prediction = classifier.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print('Akurasi data testing adalah = ', test_data_accuracy)
```

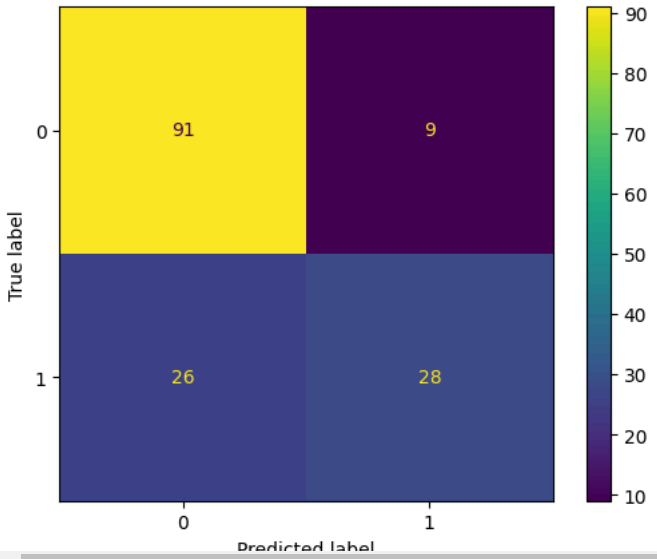
```
Akurasi data testing adalah =  0.7727272727272727
```

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

```
# Hitung confusion matrix
cm = confusion_matrix(Y_test, X_test_prediction)
```

```
# Tampilkan confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=classifier.classes_)
disp.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7e24470081f0>
```




```
from google.colab import drive
drive.mount('/content/drive')
```

```
#Membuat model prediksi
input_data = (6, 148, 72, 35, 0, 33.6, 0.627, 50)

input_data_as_numpy_array = np.array(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1, -1)


# Initialize the StandardScaler and fit it to the training data (X)
scaler = StandardScaler()
scaler.fit(X) # Fit the scaler to your training data

std_data = scaler.transform(input_data_reshaped) # Now you can use transform
print(std_data)

prediction = classifier.predict(std_data)
print(prediction)

if (prediction[0] == 0):
    print('Pasien Tidak Terkena Diabetes')
else:
    print('Pasien Terkena Diabetes')
```

```
[[ 0.63994726  0.84832379  0.14964075  0.90726993 -0.69289057  0.20401277
   0.46849198  1.4259954 ]]
[0]
Pasien Tidak Terkena Diabetes
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Star
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but SVC
  warnings.warn(
```

```
import pickle
filename = 'diabetes_model.sav'
pickle.dump(classifier, open(filename, 'wb'))
```