In [1]:

```
import pandas as pd #To load data

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, ConfusionMatrixD
```

This example shows various stages of a typical machine learning classification pipeline:

- loading the data
- selecting a classification model
- dividing into training and test set
- fiting the data into model
- predicting using fitted values

In [2]:

```
#Loading the data
#For this examle I choose the famous iris data
url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
columns = ["sepal_length", "sepal_width", "petal_length", "petal_width", "clas
iris_data = pd.read_csv(url, names = columns) #reading from url
iris_data.columns = columns #Adding column header
```

In [3]:

```
#Preprocessing the data to add headers and dividing into features and labels
X = iris_data[['sepal_length','sepal_width','petal_length','petal_width']]
Y = iris_data['class']
```

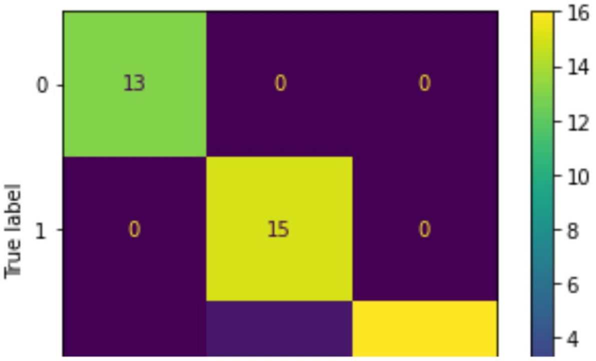
In [4]:

```
#splitting into training and testing set
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.3)
```

In [7]:

```
# Using decisoin trees with default model params
dt = DecisionTreeClassifier()

#Fitting training data into decision tree
dt.fit(x_train,y_train)

#Predicting using decision tree
prediction = dt.predict(x_test)
accuracy_dt = accuracy_score(prediction,y_test)
print('Accuracy of decision tree is', accuracy_dt)

cm = confusion_matrix(y_test, prediction)

cm_display = ConfusionMatrixDisplay(cm).plot()
```

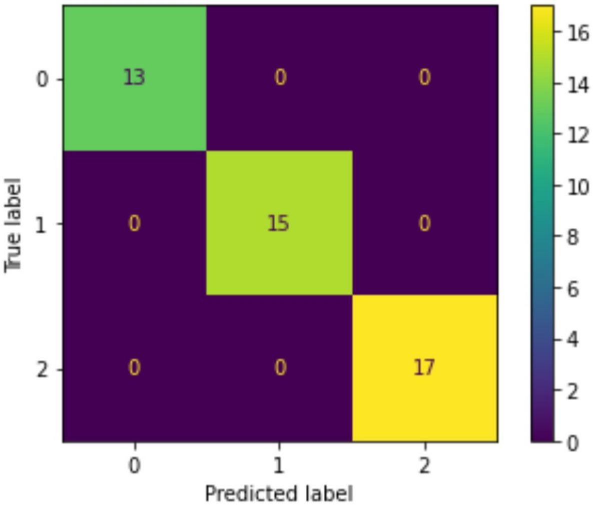
```
Accuracy of decision tree is 0.9777777777777777
```



In [6]:

```
# Using KNN with default model params
knn = KNeighborsClassifier()

#Fitting data into knn
knn.fit(x_train,y_train)

#Predicting using knn
prediction = knn.predict(x_test)
accuracy_knn = accuracy_score(prediction,y_test)
print('Accuracy of knn tree is', accuracy_knn)

cm = confusion_matrix(y_test, prediction)

cm_display = ConfusionMatrixDisplay(cm).plot()
```

```
Accuracy of knn tree is 1.0
```


True label
Predicted label