

```
# Import library yang diperlukan
from sklearn import datasets
from sklearn.model_selection import train_test_split

# Load dataset Iris
iris = datasets.load_iris()
X = iris.data
y = iris.target

# Pisahkan data menjadi data training dan testing (80:20)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Cek dimensi data training dan testing
print("Dimensi Data Training:", X_train.shape)
print("Dimensi Data Testing:", X_test.shape)
```

```
    Dimensi Data Training: (120, 4)
    Dimensi Data Testing: (30, 4)
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Latih model K-NN dengan k=3
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train, y_train)

# Lakukan prediksi terhadap data testing
y_pred_knn = knn_model.predict(X_test)

# Hitung akurasi model K-NN
akurasi_knn = accuracy_score(y_test, y_pred_knn)
print("Tingkat Akurasi Model K-NN: %d persen" % (akurasi_knn * 100))
```

```
    Tingkat Akurasi Model K-NN: 100 persen
```

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

# Latih model Naive Bayes
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)

# Lakukan prediksi terhadap data testing
y_pred_nb = nb_model.predict(X_test)

# Hitung akurasi model Naive Bayes
akurasi_nb = accuracy_score(y_test, y_pred_nb)
print("Tingkat Akurasi Model Naive Bayes: %d persen" % (akurasi_nb * 100))
```

```
    Tingkat Akurasi Model Naive Bayes: 100 persen
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Latih model C4.5
c45_model = DecisionTreeClassifier()
c45_model.fit(X_train, y_train)

# Lakukan prediksi terhadap data testing
y_pred_c45 = c45_model.predict(X_test)

# Hitung akurasi model C4.5
akurasi_c45 = accuracy_score(y_test, y_pred_c45)
print("Tingkat Akurasi Model C4.5: %d persen" % (akurasi_c45 * 100))
```

Tingkat Akurasi Model C4.5: 100 persen

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

# Latih model K-NN dengan k=3
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train, y_train)

# Lakukan prediksi terhadap data testing untuk K-NN
y_pred_knn = knn_model.predict(X_test)
akurasi_knn = accuracy_score(y_test, y_pred_knn)

# Latih model Naive Bayes
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)

# Lakukan prediksi terhadap data testing untuk Naive Bayes
y_pred_nb = nb_model.predict(X_test)
akurasi_nb = accuracy_score(y_test, y_pred_nb)

# Latih model C4.5
c45_model = DecisionTreeClassifier()
c45_model.fit(X_train, y_train)

# Lakukan prediksi terhadap data testing untuk C4.5
y_pred_c45 = c45_model.predict(X_test)
akurasi_c45 = accuracy_score(y_test, y_pred_c45)

# Bandingkan akurasi ketiga model
print("Akurasi Model K-NN: %d persen" % (akurasi_knn * 100))
print("Akurasi Model Naive Bayes: %d persen" % (akurasi_nb * 100))
print("Akurasi Model C4.5: %d persen" % (akurasi_c45 * 100))

Akurasi Model K-NN: 100 persen
Akurasi Model Naive Bayes: 100 persen
Akurasi Model C4.5: 100 persen
```