```
import tensorflow as tf
import numpy as np #untuk set data array as number
from tensorflow import keras


xs = np.array([1.0,  2.0, 3.0, 4.0, 5.0, 6.0], dtype=float) #atribut
ys = np.array([4.0, 6.0, 8.0, 10.0, 12.0, 14.0], dtype=float) #label 2X+2


#set sequential model sederhana
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])


#set model agar bisa menggunakan fungsi compile, optimizer & loss
model.compile(optimizer='sgd', loss='mean_squared_error')


#mempelajari hubungan atribut dan label denga set epoch hingga 150
model.fit(xs, ys, epochs=150)
```

```
1/1 [==============================] - 0s 9ms/step - loss: 0.1849
Epoch 111/150
1/1 [==============================] - 0s 13ms/step - loss: 0.1835
Epoch 112/150
1/1 [==============================] - 0s 12ms/step - loss: 0.1822
Epoch 113/150
1/1 [==============================] - 0s 9ms/step - loss: 0.1809
Epoch 114/150
1/1 [==============================] - 0s 9ms/step - loss: 0.1795
Epoch 115/150
1/1 [==============================] - 0s 9ms/step - loss: 0.1782
Epoch 116/150
1/1 [==============================] - 0s 13ms/step - loss: 0.1769
Epoch 117/150
1/1 [==============================] - 0s 13ms/step - loss: 0.1757
Epoch 118/150
1/1 [==============================] - 0s 17ms/step - loss: 0.1744
Epoch 119/150
1/1 [==============================] - 0s 9ms/step - loss: 0.1731
Epoch 120/150
1/1 [==============================] - 0s 9ms/step - loss: 0.1718
Epoch 121/150
1/1 [==============================] - 0s 10ms/step - loss: 0.1706
Epoch 122/150
1/1 [==============================] - 0s 14ms/step - loss: 0.1693
Epoch 123/150
1/1 [==============================] - 0s 16ms/step - loss: 0.1681
Epoch 124/150
1/1 [==============================] - 0s 12ms/step - loss: 0.1669
Epoch 125/150
1/1 [==============================] - 0s 13ms/step - loss: 0.1657
Epoch 126/150
1/1 [==============================] - 0s 10ms/step - loss: 0.1645
```

```
#memprediksi data baru
print(model.predict([10.0]))
```

```
1/1 [==============================] - 0s 148ms/step
[[23.127064]]
```

```
#set sequensial model dengan beberapa layer
model = tf.keras.Sequential([
                            keras.layers.Dense(units=1, input_shape=[1]),
                            keras.layers.Dense(units=8),
                            keras.layers.Dense(units=1)])


#set model agar bisa menggunakan fungsi compile, optimizer & loss
model.compile(optimizer='sgd', loss='mean_squared_error')


#mempelajari hubungan atribut dan label dengan epoch 300
model.fit(xs, ys, epochs=300)
```

```
1/1 [==============================] - 0s 12ms/step - loss: 1.8047
Epoch 235/300
1/1 [==============================] - 0s 9ms/step - loss: 3.1153
Epoch 236/300
1/1 [==============================] - 0s 11ms/step - loss: 1.7903
Epoch 237/300
1/1 [==============================] - 0s 13ms/step - loss: 3.0834
Epoch 238/300
1/1 [==============================] - 0s 9ms/step - loss: 1.7761
Epoch 239/300
1/1 [==============================] - 0s 9ms/step - loss: 3.0517
Epoch 240/300
1/1 [==============================] - 0s 9ms/step - loss: 1.7620
Epoch 241/300
1/1 [==============================] - 0s 9ms/step - loss: 3.0205
Epoch 242/300
1/1 [==============================] - 0s 9ms/step - loss: 1.7479
Epoch 243/300
1/1 [==============================] - 0s 9ms/step - loss: 2.9896
Epoch 244/300
1/1 [==============================] - 0s 9ms/step - loss: 1.7341
Epoch 245/300
1/1 [==============================] - 0s 9ms/step - loss: 2.9591
Epoch 246/300
1/1 [==============================] - 0s 10ms/step - loss: 1.7203
Epoch 247/300
```

```
#memprediksi data baru
print(model.predict([10.0]))
```

```
WARNING:tensorflow:5 out of the last 6 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7998b6b04a60> t
1/1 [==============================] - 0s 153ms/step
[[17.698238]]
```

```
#set sequensial model dengan 3 layer
model_3_layers = tf.keras.Sequential([
                            keras.layers.Dense(units=1, input_shape=[1]),
                            keras.layers.Dense(units=8),
                            keras.layers.Dense(units=1)])
```

```
#supaya model dapat belajar maka gunakan fungsi compile, optimizer, loss
model_3_layers.compile(optimizer='sgd', loss='mean_squared_error')
```

```
#mempelajari hubungan atribut dan label
model_3_layers.fit(xs, ys, epochs=300)
```

```
Epoch 240/300
1/1 [==============================] - 0s 9ms/step - loss: 1.7559
Epoch 241/300
1/1 [==============================] - 0s 9ms/step - loss: 3.0043
Epoch 242/300
1/1 [==============================] - 0s 9ms/step - loss: 1.7419
Epoch 243/300
1/1 [==============================] - 0s 9ms/step - loss: 2.9734
Epoch 244/300
1/1 [==============================] - 0s 9ms/step - loss: 1.7280
Epoch 245/300
1/1 [==============================] - 0s 9ms/step - loss: 2.9428
Epoch 246/300
1/1 [==============================] - 0s 9ms/step - loss: 1.7142
Epoch 247/300
```

```
import matplotlib.pyplot as plt

# Visualisasi hasil untuk model pertama
predictions_model1 = model.predict(xs) #deklarasi nilai prediksi 1 dengan data model

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.scatter(xs, ys, label='Data Asli')
plt.plot(xs, predictions_model1, label='Prediksi Model 1', color='red')
plt.xlabel('Atribut (X)')
plt.ylabel('Label (Y)')
plt.title('Visualisasi Hasil Prediksi Model 1')
plt.legend()

# Visualisasi hasil untuk model kedua
predictions_model2 = model_3_layers.predict(xs) #deklarasi nilai prediksi 2 dengan data model 3 layer

plt.subplot(1, 2, 2)
plt.scatter(xs, ys, label='Data Asli')
plt.plot(xs, predictions_model2, label='Prediksi Model 2', color='blue')
plt.xlabel('Atribut (X)')
plt.ylabel('Label (Y)')
plt.title('Visualisasi Hasil Prediksi Model 2')
plt.legend()

plt.tight_layout()
plt.show()
```

```
1/1 [==============================] - 0s 21ms/step
WARNING:tensorflow:6 out of the last 8 calls to <function Model.make_predict_function.<locals>.predict_function at 0x79
1/1 [==============================] - 0s 73ms/step
```