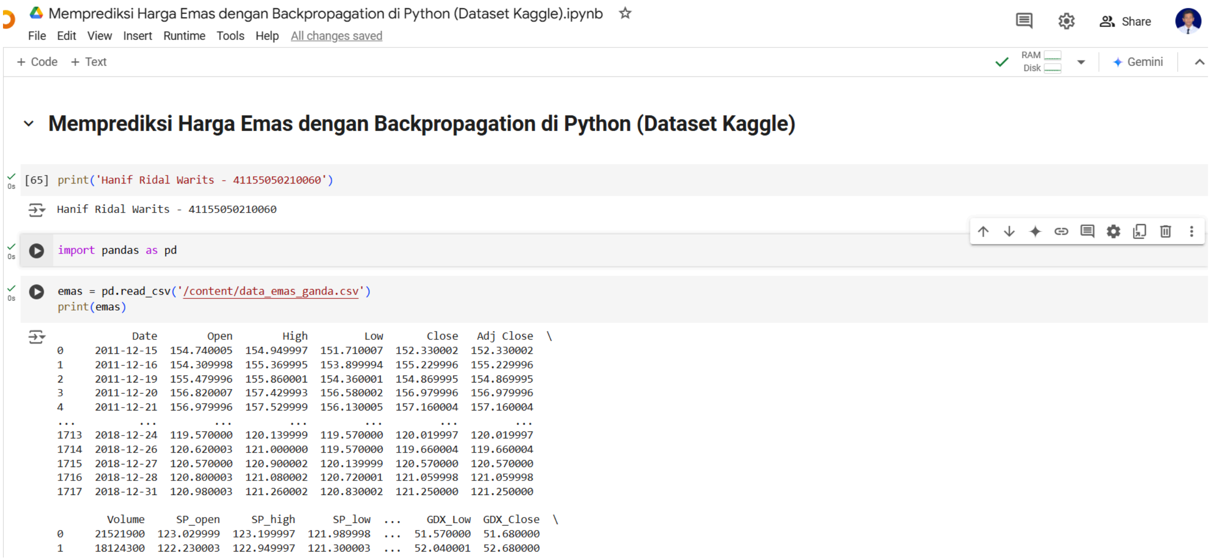
MACHINE LEARNING

Nama : Hanif Ridal Warits

NPM : 41155050210060

Kelas : Informatika A2 - 2021

Tugas Pertemuan 12



Memprediksi Harga Emas dengan Backpropagation di Python (Dataset Kaggle).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Share
+ Code + Text
RAM
Disk
Gemini

Memprediksi Harga Emas dengan Backpropagation di Python (Dataset Kaggle)

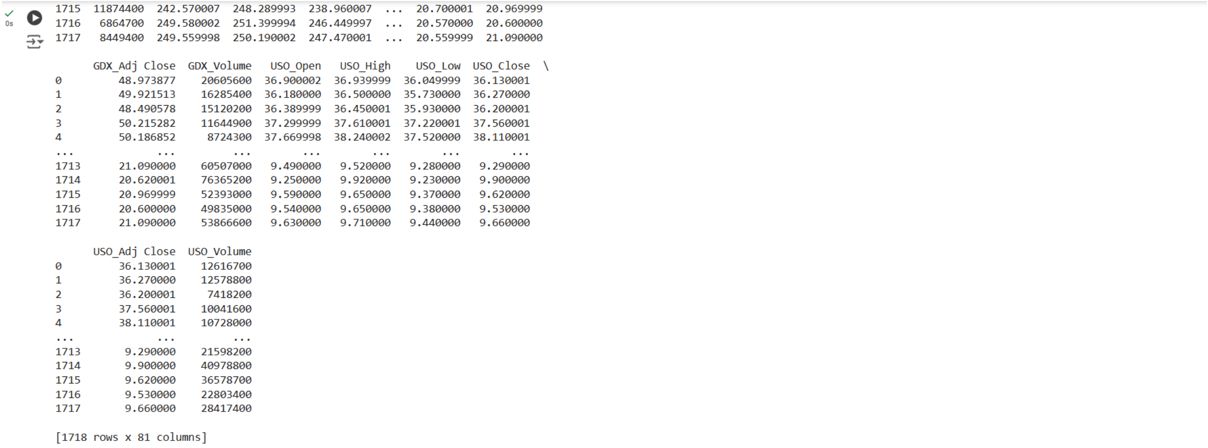
```
[65] print('Hanif Ridal Warits - 41155050210060')
```

```
Hanif Ridal Warits - 41155050210060
```

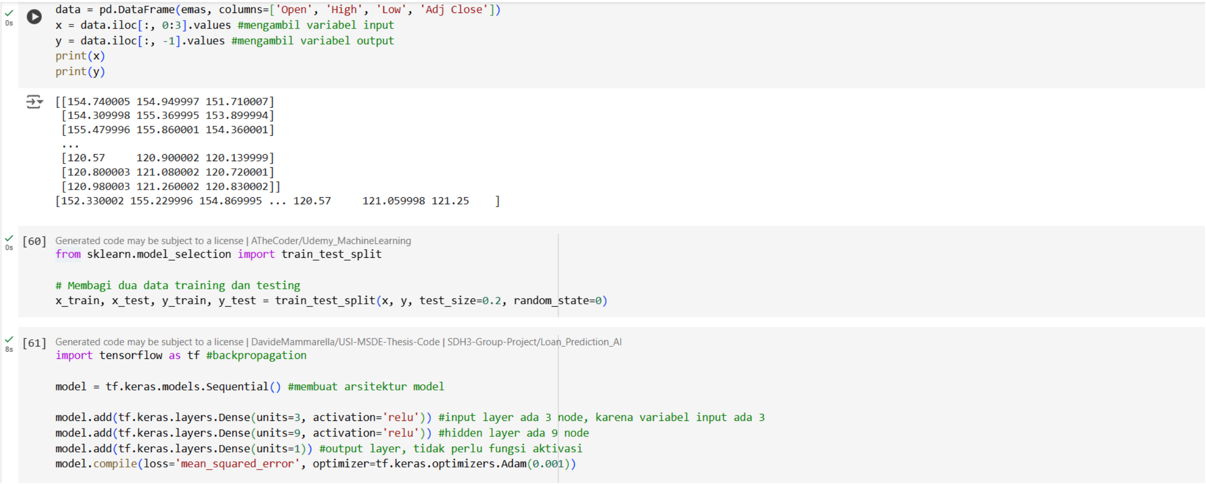
```
import pandas as pd
```

```
emas = pd.read_csv('/content/data_emas_ganda.csv')
print(emas)
```

```
            Date        Open        High         Low       Close   Adj Close  \
0     2011-12-15  154.740005  154.949997  151.710007  152.330002  152.330002
1     2011-12-16  154.309998  155.369995  153.899994  155.229996  155.229996
2     2011-12-19  155.479996  155.860001  154.360001  154.869995  154.869995
3     2011-12-20  156.820007  157.429993  156.580002  156.979996  156.979996
4     2011-12-21  156.979996  157.529999  156.130005  157.160004  157.160004
...          ...         ...         ...         ...         ...         ...
1713  2018-12-24  119.570000  120.139999  119.570000  120.019997  120.019997
1714  2018-12-26  120.620003  121.000000  119.570000  119.660004  119.660004
1715  2018-12-27  120.570000  120.900002  120.139999  120.570000  120.570000
1716  2018-12-28  120.800003  121.080002  120.720001  121.059998  121.059998
1717  2018-12-31  120.980003  121.260002  120.830002  121.250000  121.250000

        Volume     SP_open     SP_high      SP_low  ...    GDX_Low  GDX_Close  \
0     21521900  123.029999  123.199997  121.989998  ...  51.570000  51.680000
1     18124300  122.230003  122.949997  121.300003  ...  52.040001  52.680000
```



```
1715  11874400  242.570007  248.289993  238.960007  ...  20.700001  20.969999
1716   6864700  249.580002  251.399994  246.449997  ...  20.570000  20.600000
1717   8449400  249.559998  250.190002  247.470001  ...  20.559999  21.090000

      GDX_Adj Close  GDX_Volume   USO_Open   USO_High    USO_Low  USO_Close  \
0         48.973877    20605600  36.900002  36.939999  36.049999  36.130001
1         49.921513    16285400  36.180000  36.500000  35.730000  36.270000
2         48.490578    15120200  36.389999  36.450001  35.930000  36.200001
3         50.215282    11644900  37.299999  37.610001  37.220001  37.560001
4         50.186852     8724300  37.669998  38.240002  37.520000  38.110001
...             ...         ...        ...        ...        ...        ...
1713      21.090000    60507000   9.490000   9.520000   9.280000   9.290000
1714      20.620001    76365200   9.250000   9.920000   9.230000   9.900000
1715      20.969999    52393000   9.590000   9.650000   9.370000   9.620000
1716      20.600000    49835000   9.540000   9.650000   9.380000   9.530000
1717      21.090000    53866600   9.630000   9.710000   9.440000   9.660000

      USO_Adj Close  USO_Volume
0         36.130001    12616700
1         36.270000    12578800
2         36.200001     7418200
3         37.560001    10041600
4         38.110001    10728000
...             ...         ...
1713       9.290000    21598200
1714       9.900000    40978800
1715       9.620000    36578700
1716       9.530000    22803400
1717       9.660000    28417400

[1718 rows x 81 columns]
```



```
data = pd.DataFrame(emas, columns=['Open', 'High', 'Low', 'Adj Close'])
x = data.iloc[:, 0:3].values #mengambil variabel input
y = data.iloc[:, -1].values #mengambil variabel output
print(x)
print(y)
```

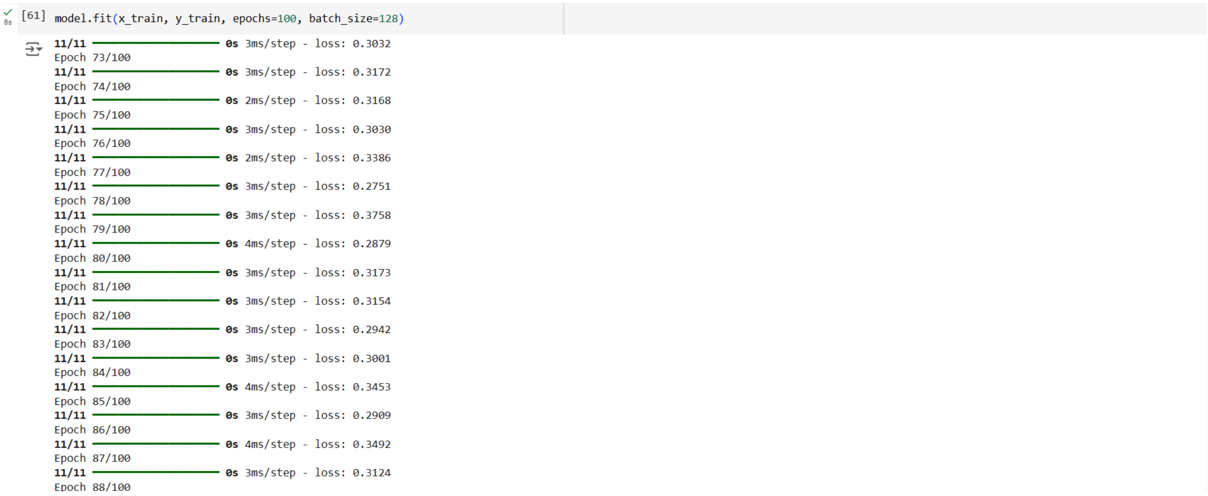
```
[[154.740005 154.949997 151.710007]
 [154.309998 155.369995 153.899994]
 [155.479996 155.860001 154.360001]
 ...
 [120.57     120.900002 120.139999]
 [120.800003 121.080002 120.720001]
 [120.980003 121.260002 120.830002]]
[152.330002 155.229996 154.869995 ... 120.57     121.059998 121.25    ]
```

[60] Generated code may be subject to a license | ATheCoder/Udemy_MachineLearning

```
from sklearn.model_selection import train_test_split

# Membagi dua data training dan testing
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```

[61] Generated code may be subject to a license | DavideMammarella/USI-MSDE-Thesis-Code | SDH3-Group-Project/Loan_Prediction_AI

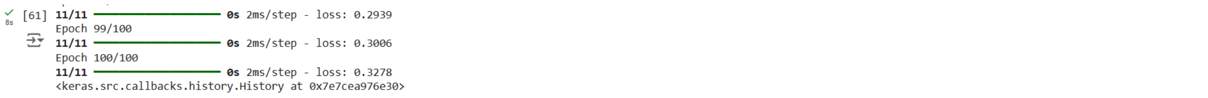
```
import tensorflow as tf #backpropagation

model = tf.keras.models.Sequential() #membuat arsitektur model

model.add(tf.keras.layers.Dense(units=3, activation='relu')) #input layer ada 3 node, karena variabel input ada 3
model.add(tf.keras.layers.Dense(units=9, activation='relu')) #hidden layer ada 9 node
model.add(tf.keras.layers.Dense(units=1)) #output layer, tidak perlu fungsi aktivasi
model.compile(loss='mean_squared_error', optimizer=tf.keras.optimizers.Adam(0.001))
```



[61]

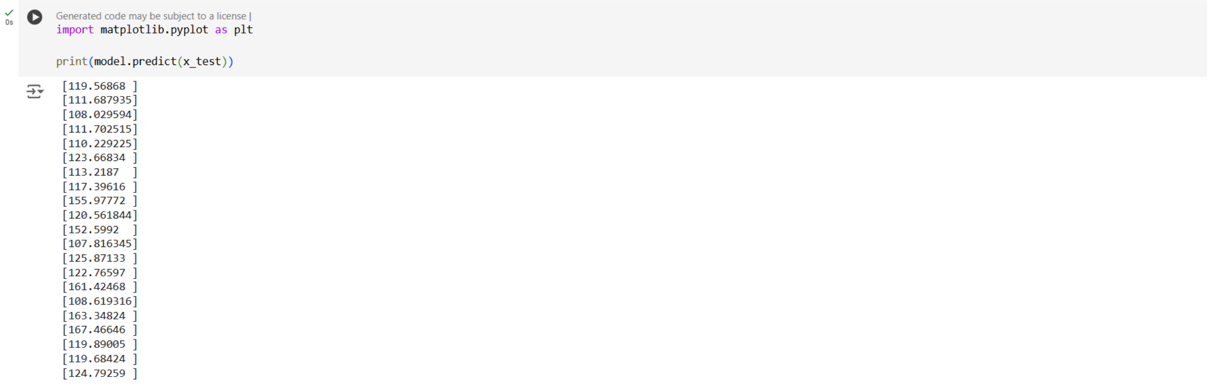
```
model.fit(x_train, y_train, epochs=100, batch_size=128)
```

```
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3032
Epoch 73/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3172
Epoch 74/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.3168
Epoch 75/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3030
Epoch 76/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.3386
Epoch 77/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2751
Epoch 78/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3758
Epoch 79/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2879
Epoch 80/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3173
Epoch 81/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3154
Epoch 82/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2942
Epoch 83/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3001
Epoch 84/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3453
Epoch 85/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2909
Epoch 86/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3492
Epoch 87/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3124
Epoch 88/100
```



[61]

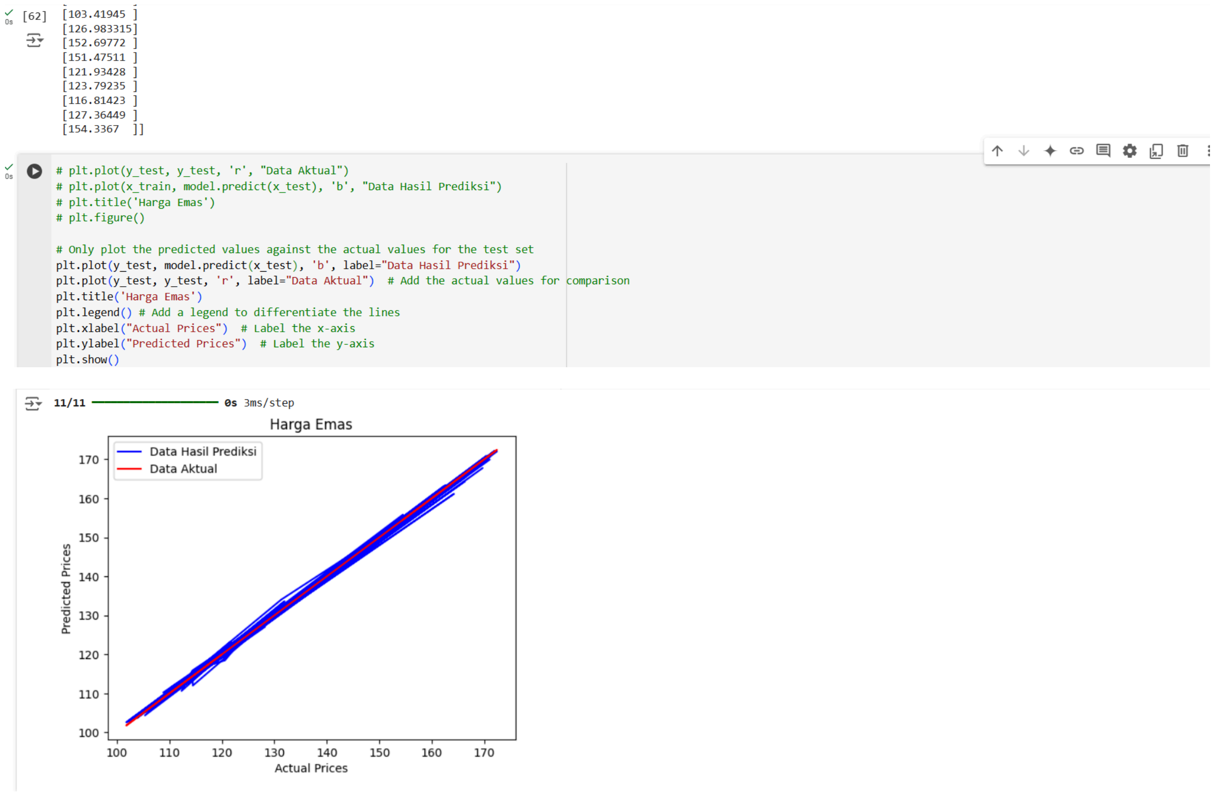
```
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.2939
Epoch 99/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.3006
Epoch 100/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.3278
<keras.src.callbacks.history.History at 0x7e7cea976e30>
```



Generated code may be subject to a license |

```
import matplotlib.pyplot as plt

print(model.predict(x_test))
```

```
 [119.56868 ]
 [111.687935]
 [108.029594]
 [111.702515]
 [110.229225]
 [123.66834 ]
 [113.2187  ]
 [117.39616 ]
 [155.97772 ]
 [120.561844]
 [152.5992  ]
 [107.816345]
 [125.87133 ]
 [122.76597 ]
 [161.42468 ]
 [108.619316]
 [163.34824 ]
 [167.46646 ]
 [119.89005 ]
 [119.68424 ]
 [124.79259 ]
```



[62]

```
 [103.41945 ]
 [126.983315]
 [152.69772 ]
 [151.47511 ]
 [121.93428 ]
 [123.79235 ]
 [116.81423 ]
 [127.36449 ]
 [154.3367  ]]
```

```
# plt.plot(y_test, y_test, 'r', "Data Aktual")
# plt.plot(x_train, model.predict(x_test), 'b', "Data Hasil Prediksi")
# plt.title('Harga Emas')
# plt.figure()

# Only plot the predicted values against the actual values for the test set
plt.plot(y_test, model.predict(x_test), 'b', label="Data Hasil Prediksi")
plt.plot(y_test, y_test, 'r', label="Data Aktual")  # Add the actual values for comparison
plt.title('Harga Emas')
plt.legend() # Add a legend to differentiate the lines
plt.xlabel("Actual Prices")  # Label the x-axis
plt.ylabel("Predicted Prices")  # Label the y-axis
plt.show()
```

```
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
```