1. **Praktikum**
2. Buka Google Colaboratory melalui [tautan ini](https://github.com/tridims/Jaringan-Saraf-Tiruan/blob/main/Bab%209/).
3. Tulis kode berikut ke dalam setiap *cell* pada *notebook* tersebut.
   1. Fungsi *Training* Backpropagation

|  |
| --- |
| import numpy as np  def bp\_fit(X, target, layer\_conf, max\_epoch, max\_error=.1, learn\_rate=.1, print\_per\_epoch=100):  np.random.seed(1)  nin = [np.empty(i) for i in layer\_conf]  n = [np.empty(j + 1) if i < len(layer\_conf) - 1 else np.empty(j) for i, j in enumerate(layer\_conf)]  w = np.array([np.random.rand(layer\_conf[i] + 1, layer\_conf[i + 1]) for i in range(len(layer\_conf) - 1)], dtype=object)  dw = [np.empty((layer\_conf[i] + 1, layer\_conf[i + 1])) for i in range(len(layer\_conf) - 1)]  d = [np.empty(s) for s in layer\_conf[1:]]  din = [np.empty(s) for s in layer\_conf[1:-1]]  epoch = 0  mse = 1  for i in range(0, len(n)-1):  n[i][-1] = 1  while (max\_epoch == -1 or epoch < max\_epoch) and mse > max\_error:  epoch += 1  mse = 0  for r in range(len(X)):  n[0][:-1] = X[r]  for L in range(1, len(layer\_conf)):  nin[L] = np.dot(n[L-1], w[L-1])  n[L][:len(nin[L])] = sig(nin[L])  e = target[r] - n[-1]  mse += sum(e \*\* 2)  d[-1] = e \* sigd(nin[-1])  dw[-1] = learn\_rate \* d[-1] \* n[-2].reshape((-1, 1))  for L in range(len(layer\_conf) - 1, 1, -1):  din[L-2] = np.dot(d[L-1], np.transpose(w[L-1][:-1]))  d[L-2] = din[L-2] \* np.array(sigd(nin[L-1]))  dw[L-2] = (learn\_rate \* d[L-2]) \* n[L-2].reshape((-1, 1))  w += dw  mse /= len(X)  if print\_per\_epoch > -1 and epoch % print\_per\_epoch == 0:  print(f'Epoch {epoch}, MSE: {mse}')  return w, epoch, mse |

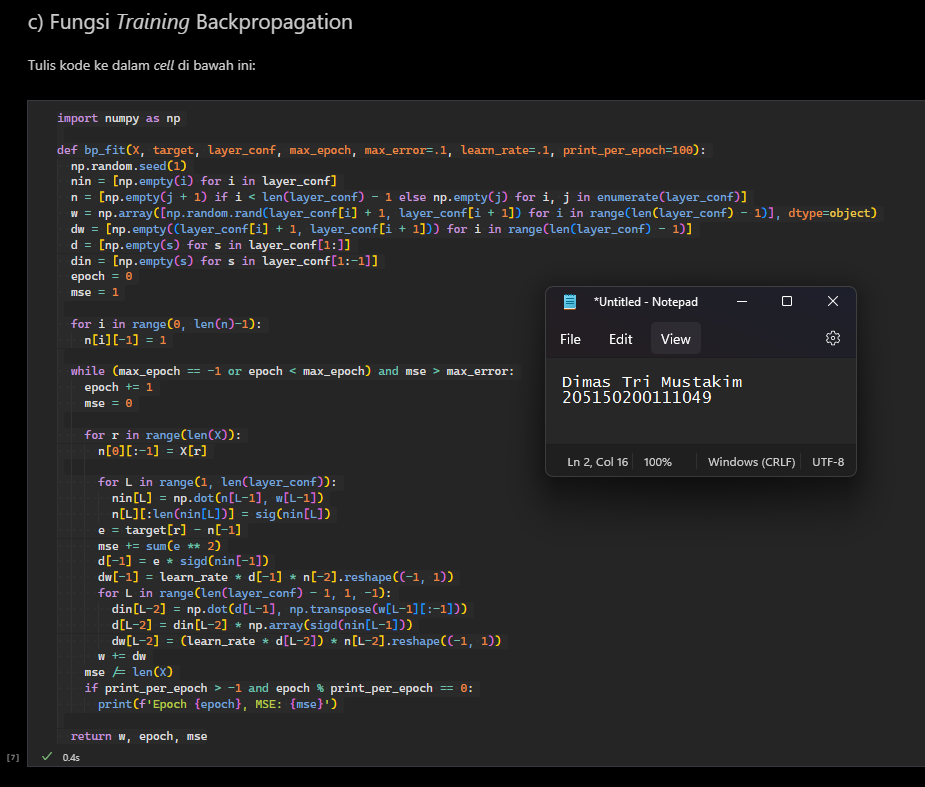
* 1. Fungsi *Testing* Backpropagation

|  |
| --- |
| def bp\_predict(X, w):  n = [np.empty(len(i)) for i in w]  nin = [np.empty(len(i[0])) for i in w]  predict = []  n.append(np.empty(len(w[-1][0])))  for x in X:  n[0][:-1] = x  for L in range(0, len(w)):  nin[L] = np.dot(n[L], w[L])  n[L + 1][:len(nin[L])] = sig(nin[L])  predict.append(n[-1].copy())  return predict |

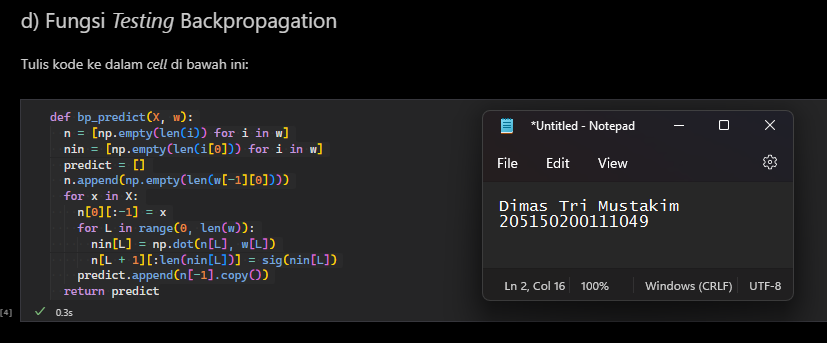
* 1. Percobaan Klasifikasi Dataset Iris

|  |
| --- |
| from sklearn import datasets  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import minmax\_scale  from sklearn.metrics import accuracy\_score  iris = datasets.load\_iris()  X = minmax\_scale(iris.data)  Y = onehot\_enc(iris.target)  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=.3,random\_state=1)  w, ep, mse = bp\_fit(X\_train, y\_train, layer\_conf=(4, 3, 3), learn\_rate=.1, max\_epoch=1000, max\_error=.1, print\_per\_epoch=25)  print(f'Epochs: {ep}, MSE: {mse}')  predict = bp\_predict(X\_test, w)  predict = onehot\_dec(predict)  y\_test = onehot\_dec(y\_test)  accuracy = accuracy\_score(predict, y\_test)  print('Output:', predict)  print('True :', y\_test)  print('Accuracy:', accuracy) |

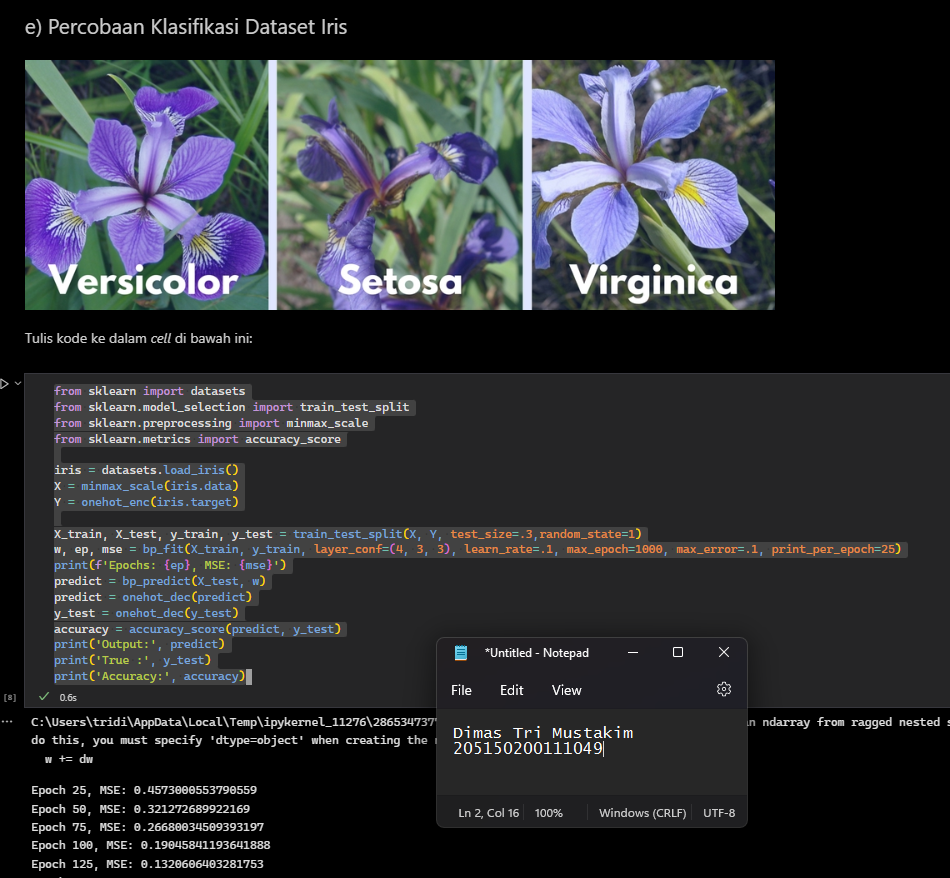
1. **Screenshot**
2. Fungsi Training Backpropagation

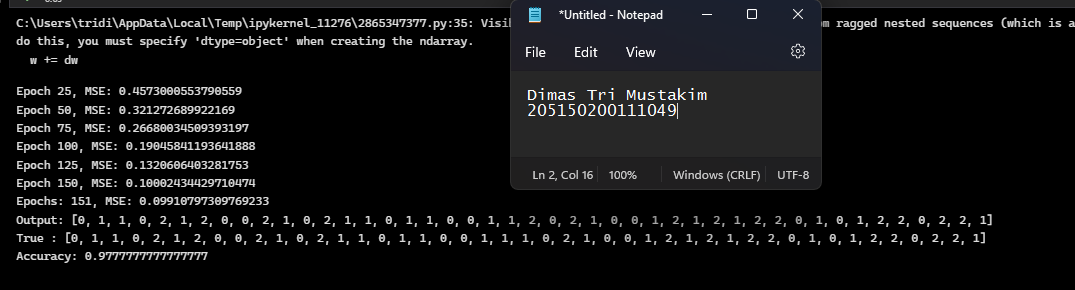


1. Fungsi Testing Backpropagation



1. Percobaan Klasifikasi Dataset Iris



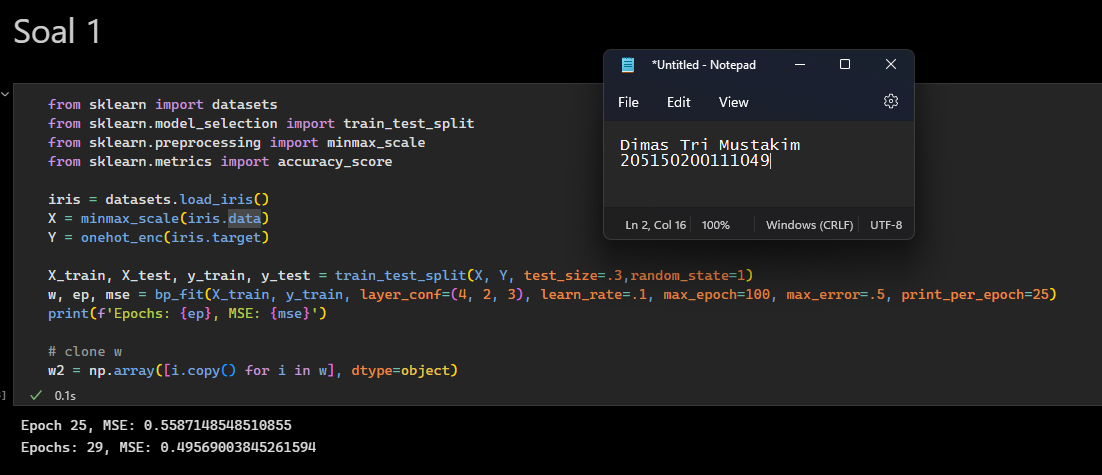


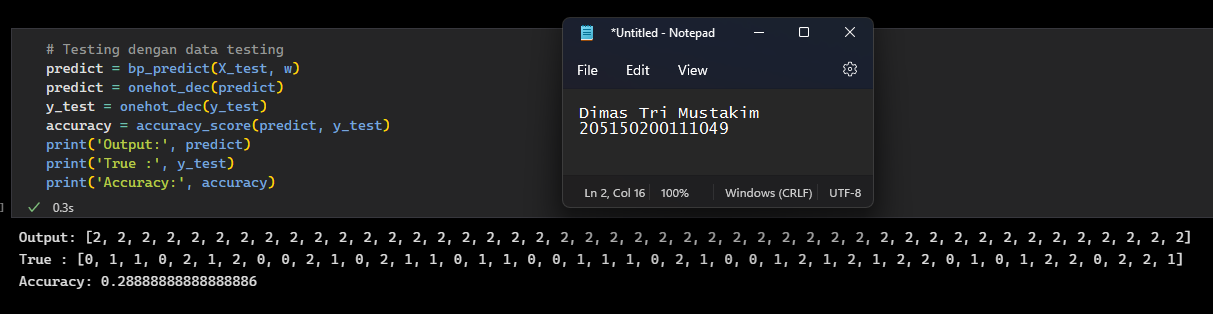
1. **Analisis**
2. Lakukan klasifikasi dengan menggunakan dataset Iris seperti di atas. Ubahlah beberapa pengaturan sebagai berikut:
   * Rasio data latih 70% dan data uji 30%
   * Hidden neuron = 2
   * Max epoch = 100
   * Learning rate = 0,1
   * Max error = 0,5

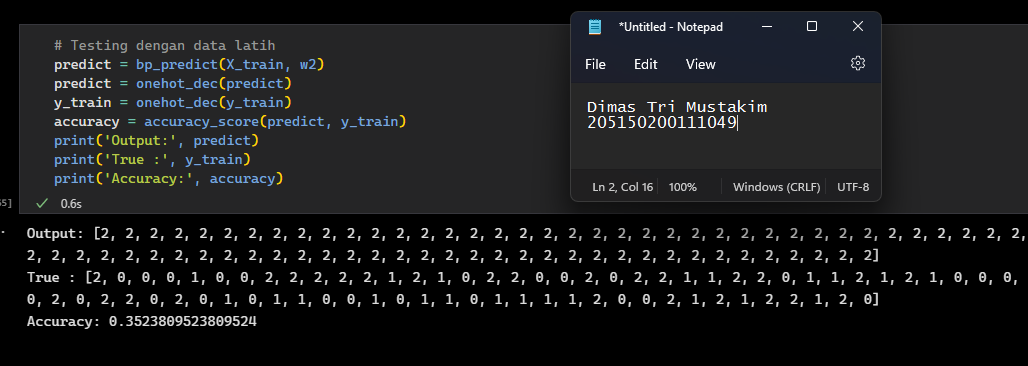
Lakukan pengujian (testing) menggunakan data latih dan data uji. Bandingkan nilai akurasi yang didapatkan. Fenomena apa yang terjadi pada pengujian ini? Mengapa hal tersebut terjadi?

**Jawab:**

Akurasi yang didapatkan dari hasil pengujian didapatkan bahwa kinerja model pada pelatihan tersebut buruk. Fenomena tersebut merupakan underfitting dan terjadi karena pemilihan parameter pelatihan yang ditunjukkan tidak optimal. Berikut screenshot proses pelatihan.





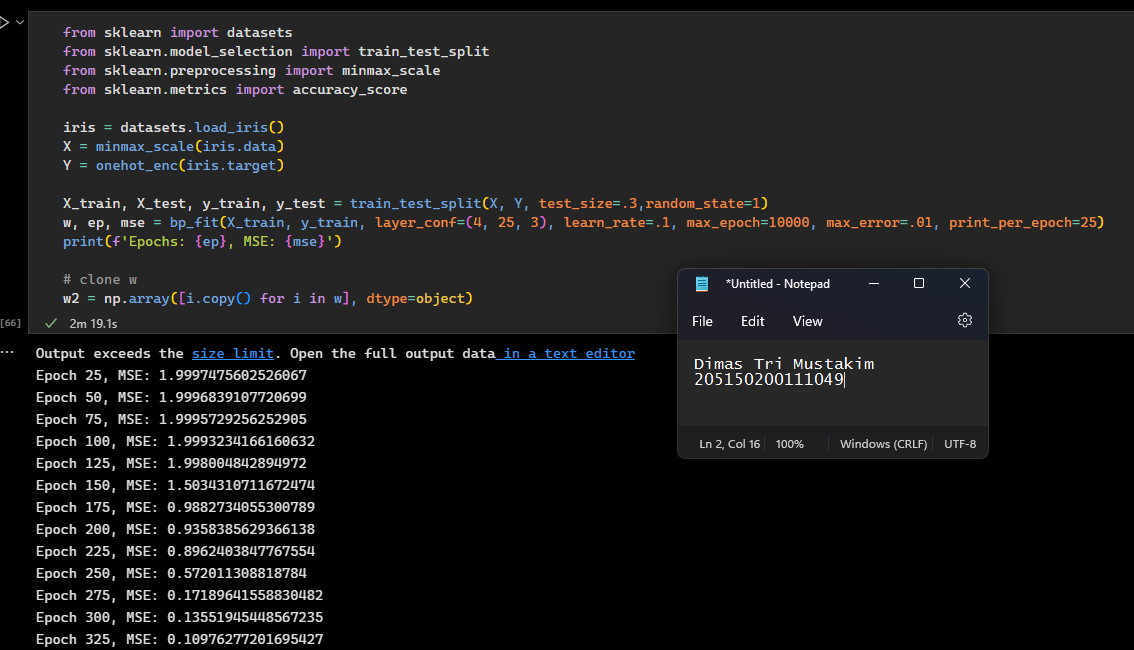


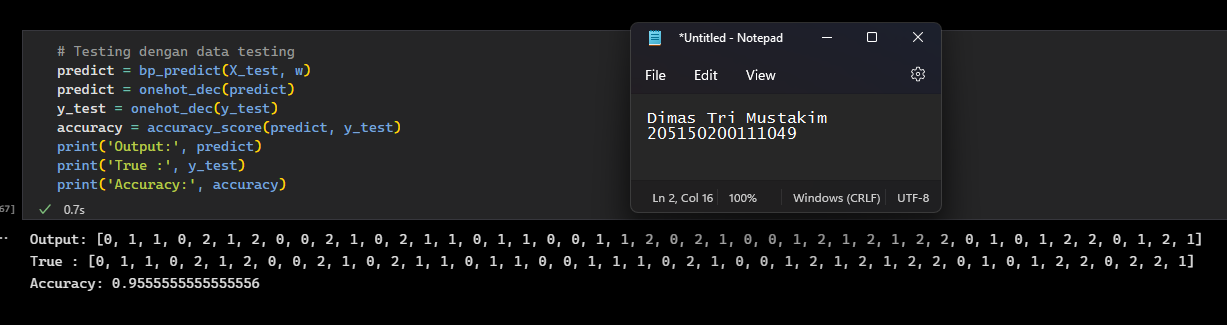
1. Lakukan klasifikasi dengan menggunakan dataset Iris seperti di atas. Ubahlah beberapa pengaturan sebagai berikut:
   * Rasio data latih 70% dan data uji 30%
   * Hidden neuron = 25
   * Max epoch = 10000
   * Learning rate = 0,1
   * Max error = 0,01

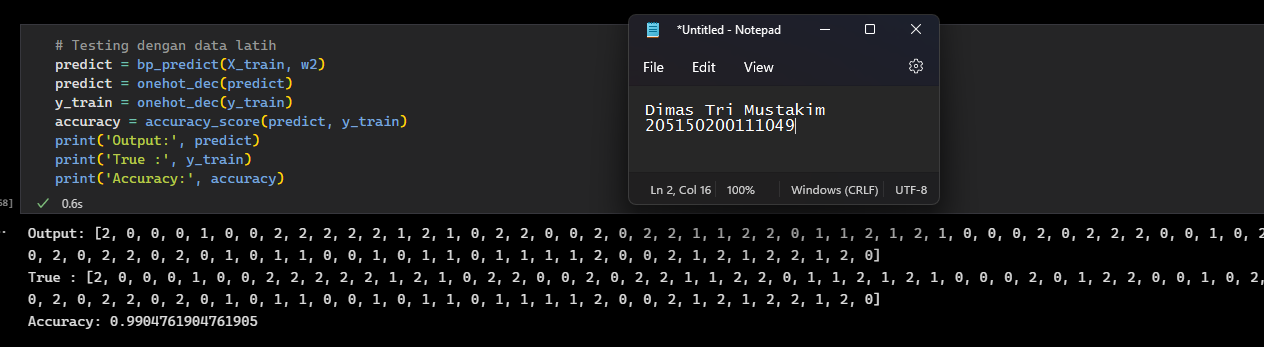
Lakukan pengujian (testing) menggunakan data latih dan data uji. Bandingkan nilai akurasi yang didapatkan. Fenomena apa yang terjadi pada pengujian ini? Mengapa hal tersebut terjadi?

**Jawab:**

Nilai akurasi yang didapatkan menunjukkan skor yang cukup tinggi, nilai akurasi ketika testing menggunakan data training menunjukkan nilai yang lebih besar daripada ketika menggunakan data testing. Fenomena tersebut menunjukkan overfitting, dimana model baik dalam mengklasifikasikan data latih, tetapi tidak dengan data testing (tidak bisa men-generalisasi). Berikut screenshot proses pelatihan dan testing.





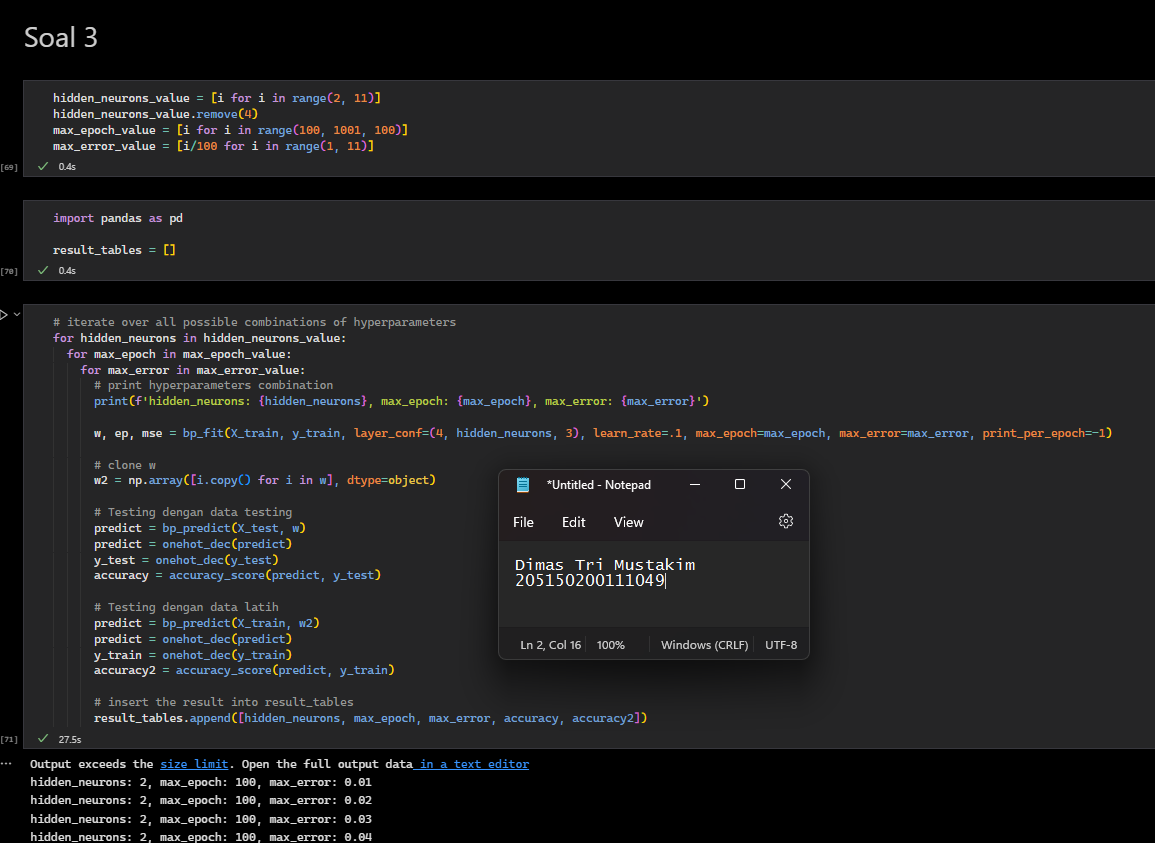


1. Ubahlah parameter berikut agar mendapatkan akurasi tertinggi saat melakukan testing menggunakan data uji :
   * Hidden neuron
   * Max epoch
   * Max error

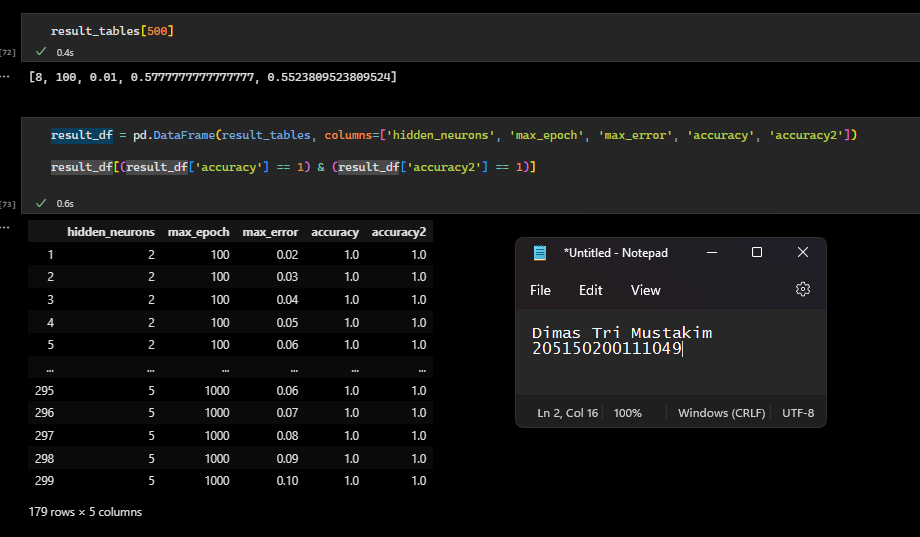
Berapakah nilai akurasi tertinggi yang dapat Anda peroleh? Berapakah nilai masing-masing parameter tersebut?

**Jawab:**

Untuk memperoleh kombinasi parameter dengan akurasi tertinggi, saya melakukan perulangan untuk beberapa parameter dari range tertentu dan kemudian dicatat akurasi terhadap data latih dan data testing. Untuk range parameter bisa dilihat di screenshot kode program dibawah.



Dari kode program diatas, didapatkan hasil seperti berikut.



Terdapat beberapa kombinasi parameter yang dapat menghasilkan nilai akurasi sempurna ketika testing menggunakan data latih dan data uji.

1. **Kesimpulan**

Overfitting merupakan fenomena ketika sebuah model machine learning memberikan hasil yang akurat ketika melakukan prediksi terhadap data latih tetapi tidak ketika mendapatkan data baru. Overfitting terjadi ketika model tidak dapat menggeneralisasi dan terlalu cocok dengan dataset pelatihan. Overfitting bisa dicegah dengan cara diversifikasi dan memperbesar data training atau menggunakan beberapa strategi data seperti early stopping, pruning, regularization, dan ensembling. Cara lain adalah dengan menghapus data yang tidak relevan (noisy data), dan mengatur hyperparameter ketika pelatihan.

Underfitting merupakan fenomena ketika model tidak bisa menentukan hubungan yang berarti antara input dan output data. Model yang underfit tidak bisa melakukan klasifikasi yang baik pada data yang diberikan. Underfit terjadi ketika tidak dilatih untuk jangka waktu yang cukup (epoch), kurangnya data latih dan fitur data, serta pemilihan hyperparameter yang tidak tepat. Cara mengatasi overfitting adalah dengan menyelesaikan penyebabnya, mulai dari mencari data dengan jumlah yang cukup banyak, fitur lengkap, dan mencari hyperparameter yang seusai.