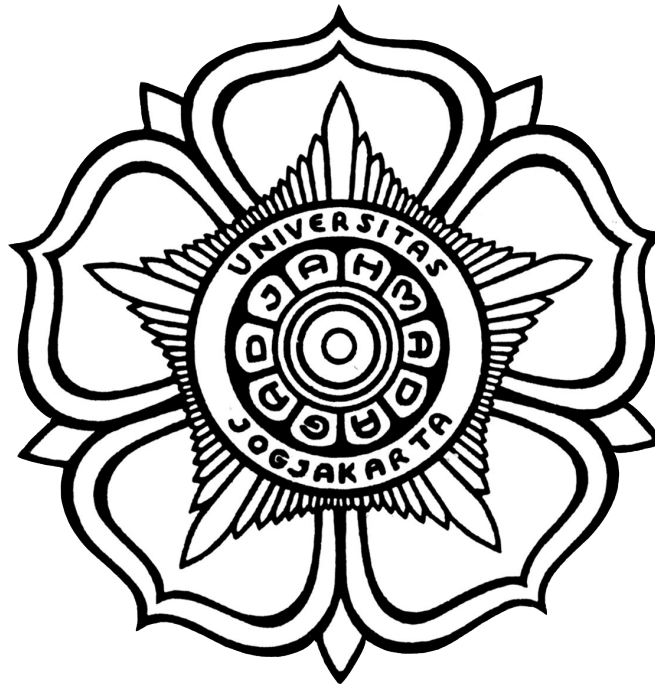


Laporan Pembelajaran Mesin
***Multiple Layer Perceptron* Untuk**
Membedakan jenis Bunga Iris



Disusun Oleh:

Nendra Haryo Wijayandanu

16/394097/PA/17188

Universitas Gadjah Mada

Link GitHub: <https://github.com/nendraharyo/MultiLayerPerceptron>

Bahasa Pemrograman: Python (PyCharm IDE)

Library yang digunakan: Pandas, Matplotlib.pyplot, Numpy

Langkah-langkah yang dilakukan adalah sebagai berikut:

1. *Split* dataset dengan perbandingan 15:135, dimana 15 data digunakan sebagai data validasi dan 135 data digunakan sebagai data *training*. 15 data yang digunakan tersebut beranggotakan 5 data dari masing-masing kelas yang ada.

```
# split the data to 15:135
df_validation = df.iloc[0:5]
df_validation = df_validation.append(df.iloc[50:55], ignore_index=True,
sort=False)
df_validation = df_validation.append(df.iloc[100:105], ignore_index=True,
sort=False)

df_training = df.iloc[5:50]
df_training = df_training.append(df.iloc[55:100], ignore_index=True,
sort=False)
df_training = df_training.append(df.iloc[105:150], ignore_index=True,
sort=False)
```

2. Tentukan *Learning-rate* dan *epoch* (pada percobaan digunakan *epoch* 500 dan 1000, dengan *Learning-rate* 0.1 dan 0.8)

```
lr = float(input("learning rate: "))
epoch = int(input("epoch: "))
```

3. *Training* data sebanyak n jumlah *epoch*, dimana setiap akhir dari suatu *epoch* dilakukan pencatatan tingkat error terakhir dan tingkat akurasi dari data *training*. Selain itu dilakukan juga validasi data setiap akhir *epoch*, pencatatan tingkat error data validasi dan tingkat akurasi dari data validasi.
4. Tingkat error dan tingkat akurasi yang telah tercatat kemudian di-*plot* dan dibandingkan dalam bentuk diagram

```
# draw the graph
fig, plotaccuracy = plt.subplots()
fig, ploterror = plt.subplots()

len(plotcorrecttrain)
plotaccuracy.set_title("Diagram Akurasi")
```

```

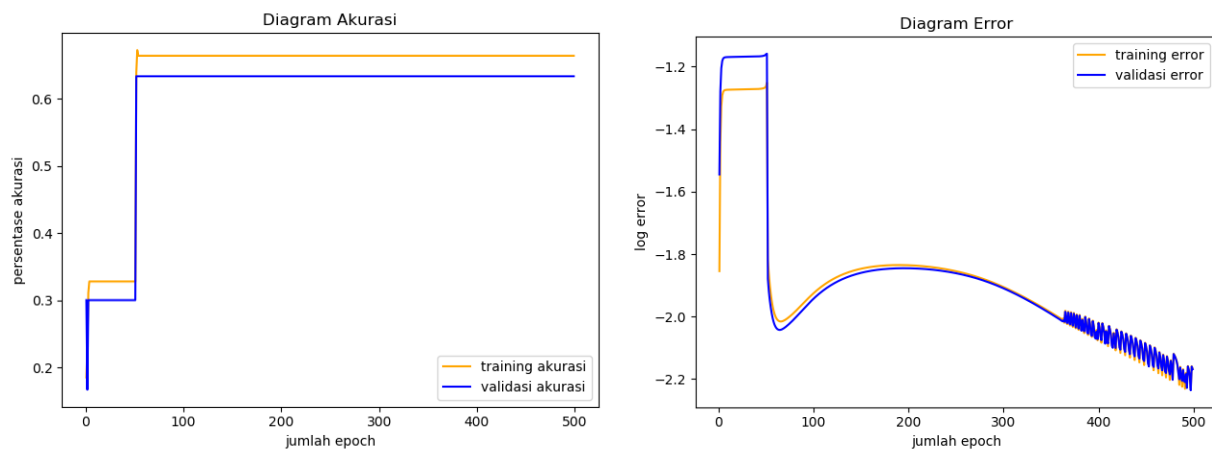
plotaccuracy.plot(plotcounter, plotcorrecttrain, color="orange", label="training
akurasi")
plotaccuracy.plot(plotcounter, plotcorrectvalidate, color="blue", label="validasi
akurasi")
plotaccuracy.set_xlabel("jumlah epoch")
plotaccuracy.set_ylabel("persentase akurasi")
plotaccuracy.legend(loc="lower right")

len(ploterrortrain)
ploterror.set_title("Diagram Error")
ploterror.plot(plotcounter, ploterrortrain, color="orange", label="training error")
ploterror.plot(plotcounter, ploterrorvalidate, color="blue", label="validasi error")
ploterror.set_xlabel("jumlah epoch")
ploterror.set_ylabel("log error")
ploterror.legend(loc="upper right")

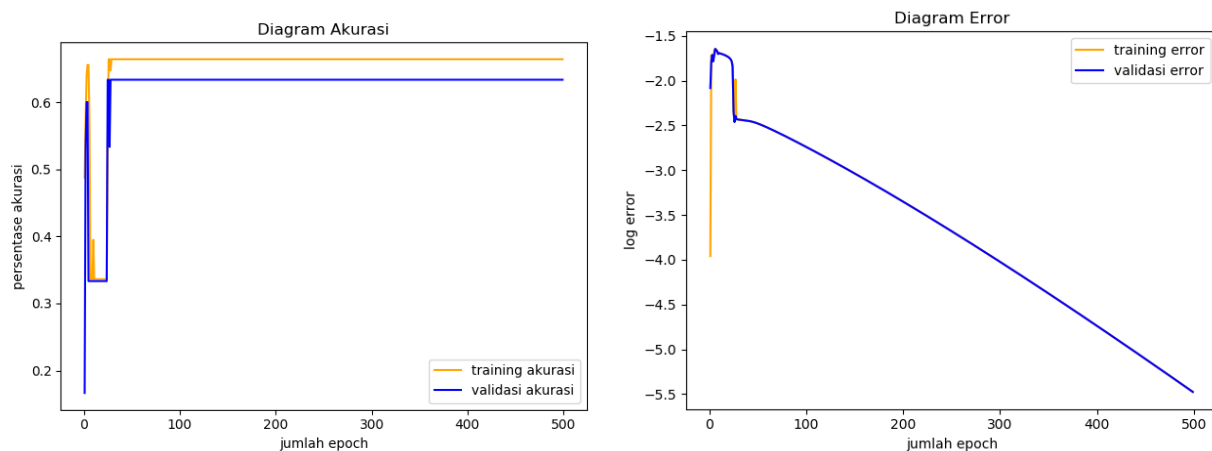
plt.show()

```

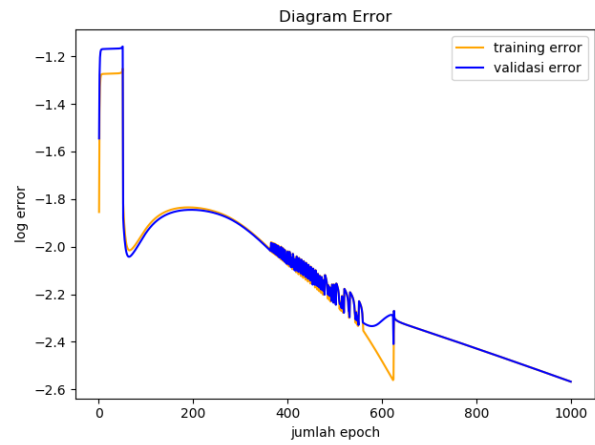
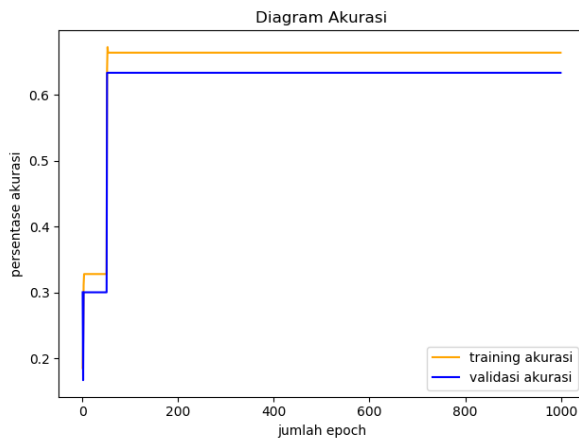
LR: 0.1, Epoch 500:



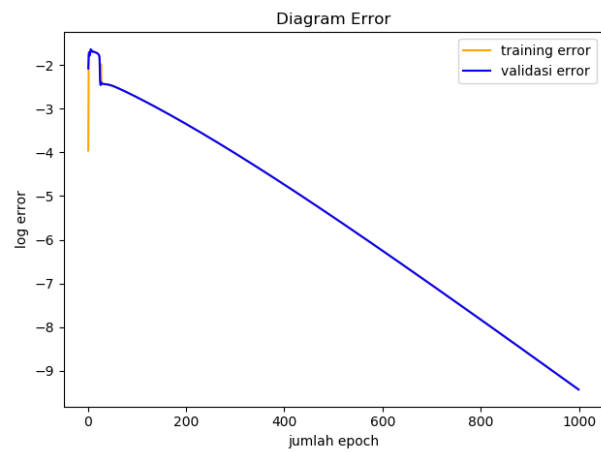
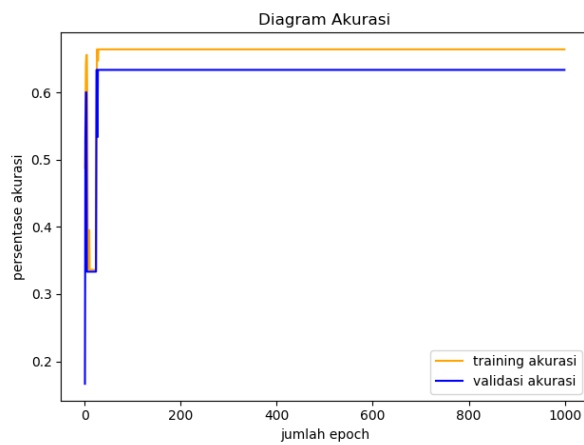
LR: 0.8, Epoch 500:



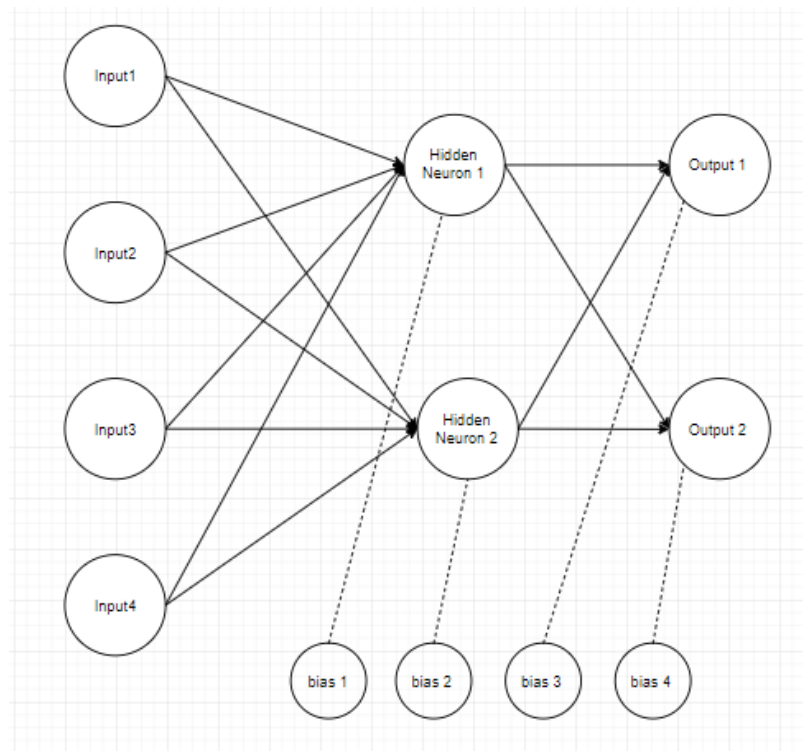
LR:0.1, Epoch 1000:



LR:0.8, Epoch 1000:



Arsitektur:



KESIMPULAN

Tingkat akurasi tertinggi tidak mencapai 70% dengan arsitektur 4 input, 1 *hidden-layer* (2 neuron), dan 2 output. Dengan meninjau kembali data hasil *training* dan validasi ditemukan bahwa kelas *Iris-Versicolor* tidak terklasifikasi sebagai kelas yang berbeda dari kelas *Iris-Virginica* (Secara tidak langsung menentukan bahwa maksimal akurasi yang mungkin tercapai tidak akan melebihi 66.6667 %).

Penggunaan bentuk arsitektur lain mungkin akan menghasilkan tingkat akurasi dan error yang lebih baik dibandingkan dengan arsitektur yang digunakan pada percobaan ini.