

# **Laporan Praktikum Pembelajaran Mesin Neural Network**



**Disusun oleh :  
Kelompok 3**

- |                                    |                |
|------------------------------------|----------------|
| 1. Gede Ranga Wira Aditya          | (082011633048) |
| 2. Muhammad Rahmadhani Ferdiansyah | (082011633068) |
| 3. Arya Danu Triatmodjo            | (082011633069) |
| 4. Mukhamad Ikhsanudin             | (082011633086) |

Kelas I3

**PROGRAM STUDI S1 SISTEM INFORMASI**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS AIRLANGGA**

**SURABAYA**

**2021/2022**

## COBA SEMUA SOURCE CODE NEURAL NETWORK

### MATLAB

#### 5. Data Training

X	Y	Target
1	1	1
1	0	1
0	1	1
0	0	0

#### 6. Membaca data dari excel

```
%Membaca data dari excel
filename = 'DataLatih.xlsx';
sheet = 1;
xlRange = 'A2:C5';
Data = xlsread (filename, sheet, xlRange);
data_latih = Data(:,1:2)';
target_latih = Data(:,3)';
[m, n] = size(data_latih);
```

#### 7. Menentukan jumlah neuron, jumlah epoch, dan nilai learning rate lalu menyimpan hasilnya

```
% Pembuatan JST
% Arsitek jaringan yang dipakai adalah 3-3-1, artinya 3 neuron input
% X,Y,Z), 3 neuron hidden (karena Inputnya ada 3 neuron maka neuron hiddennya
% Fungsi Aktivasi di hidden layer menggunakan 'logsig', di output layer menggunakan 'purelin'
% Model JST yang digunakan gradien descent maka fungsi aktivasinya adalah traingdx
net = newff(minmax(data_latih), [2 1], {'logsig', 'purelin'}, 'traingdx');

% Memberikan nilai untuk mempengaruhi proses Training
net.performFcn = 'mse';
net.trainParam.goal = 0.0001; % Errornya (0 sampai 1)
net.trainParam.show = 20; % Boleh diganti
net.trainParam.epochs = 1500; % Banyaknya epoch / iterasi training
net.trainParam.mc = 0.95;
net.trainParam.lr = 1; % Nilai learning Rate (0 sampai 1)

% Proses training
[net_keluaran, tr, Y, E] = train(net, data_latih, target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW {1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hidden = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2);

save ('C:\ikhsan\UNAIR\SEMESTER 4\PEMBELAJARAN MESIN (PRAKTIKUM)\Tugas SVM dan NN\net_keluaran.mat')
```

```

% Hasil prediksi
hasil_latih = sim(net_keluaran, data_latih);

% Performansi hasil prediksi
target_latih_asli = target_latih;

figure,
plotregression(target_latih_asli, hasil_latih, 'Regression')
figure,
plotperform(tr)

% Gambar JST
figure,
plot(hasil_latih, 'bo-')
hold on
plot(target_latih_asli, 'ro-')
hold off
grid on
title(strcat(['Grafik Keluaran JST vs Target dengan nilai MSE = ', num2str(error_MSE)]))
xlabel('Pola ke-')
ylabel('MSE')
legend('Keluaran JST', 'Target', 'Location', 'Best')

```

## 8. Data Testing

X	Y	Target
1	1	1
1	0	1
0	1	1
0	0	0

## 9. Load Jaringan Data Training

```

% load jaringan yang sudah dibuat pada proses pelatihan
load('C:\ikhshan\UNAIR\SEMESTER 4\PEMBELAJARAN MESIN (PRAKTIKUM)\Tugas SVM dan NN\net_keluaran.mat')

```

## 10. Membaca data dari excel

```

% Proses membaca data uji dari excel
filename = 'DataUji.xlsx';
sheet = 1;
xlRange = 'A2:C5';
Data = xlsread(filename, sheet, xlRange);
data_uji = Data(:,1:2)';
target_uji = Data(:, 3)';
[m, n] = size(data_uji);

```

## 11. Hasil prediksi

```

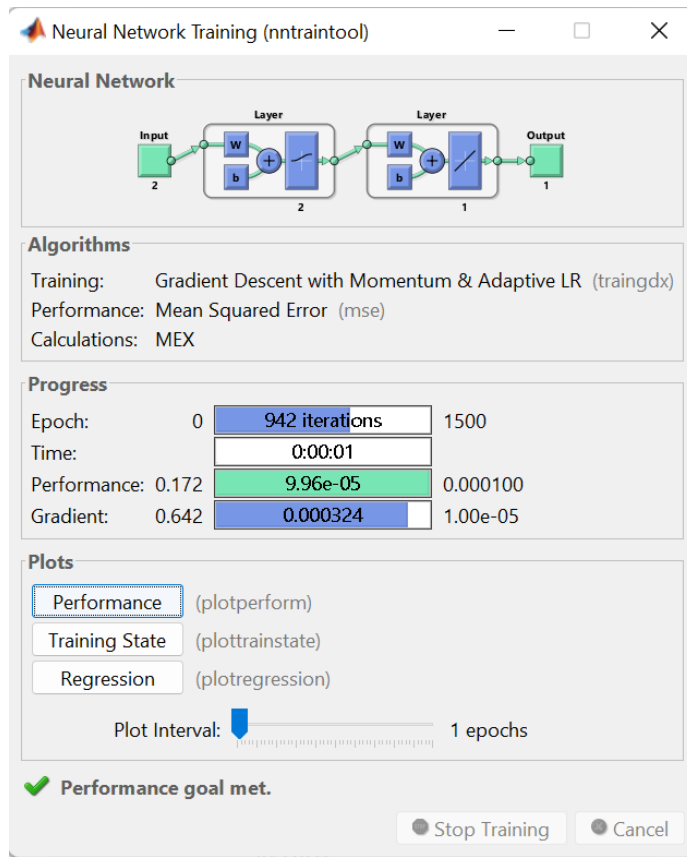
% Hasil prediksi
hasil_uji = sim(net_keluaran, data_uji);
nilai_error = abs(hasil_uji - target_uji)

% Performansi hasil prediksi
error = (1/n)*sum(nilai_error.^1);
Akurasi = (1-error)*100

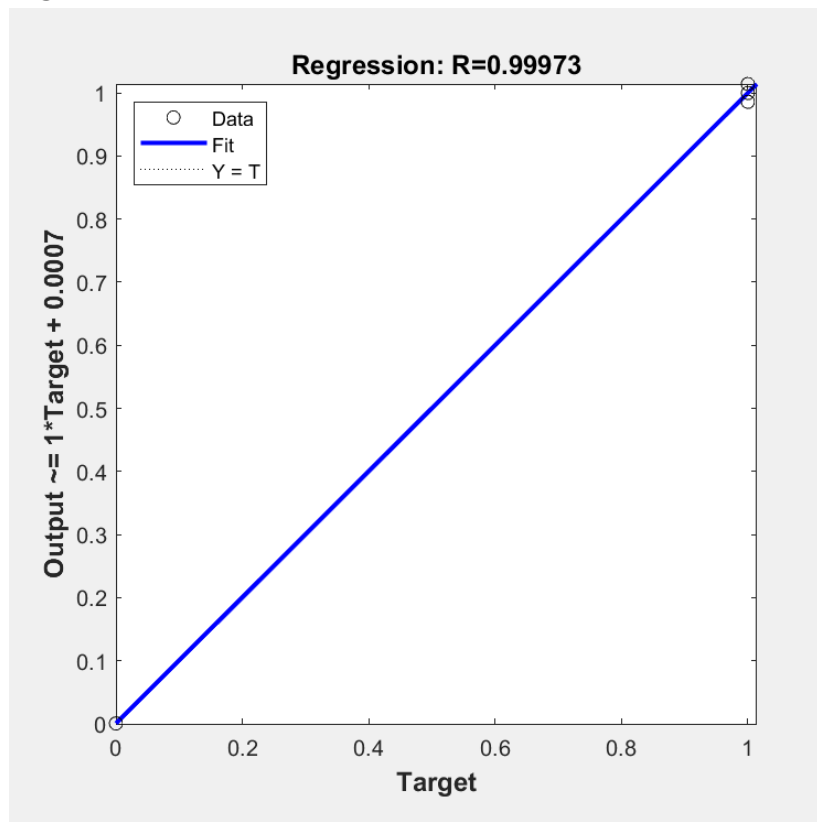
```

## Output

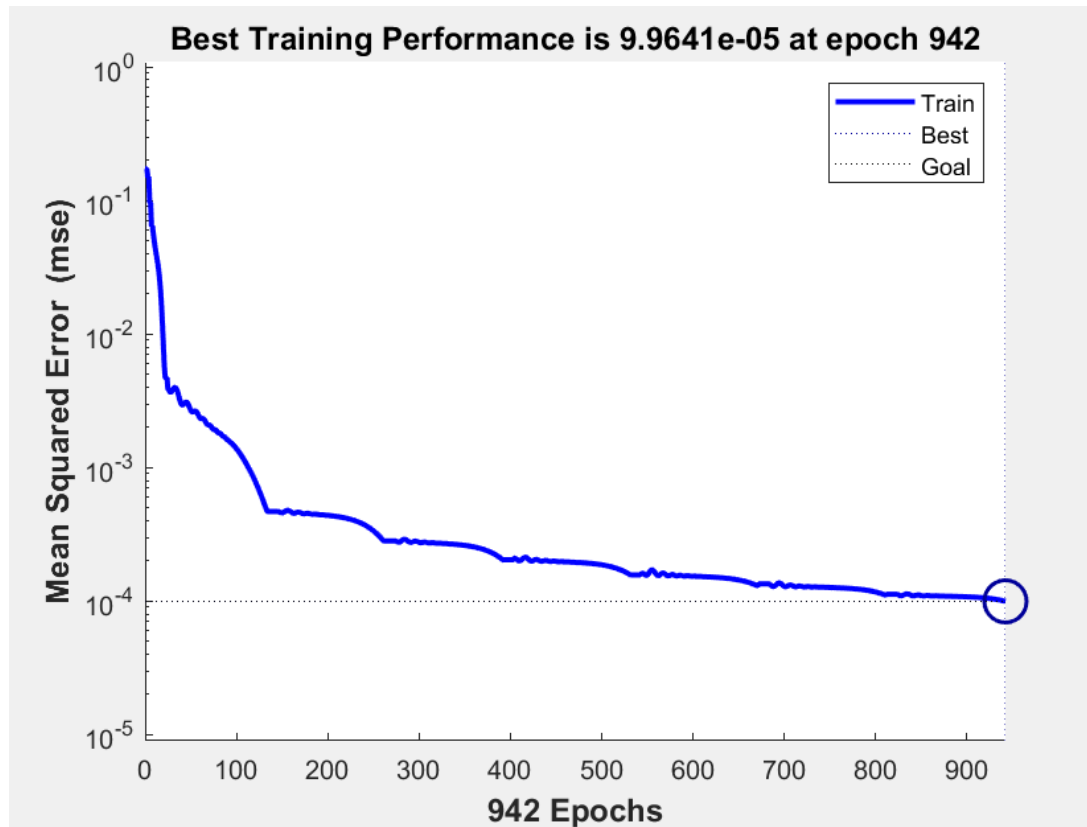
### 12. Hasil Training



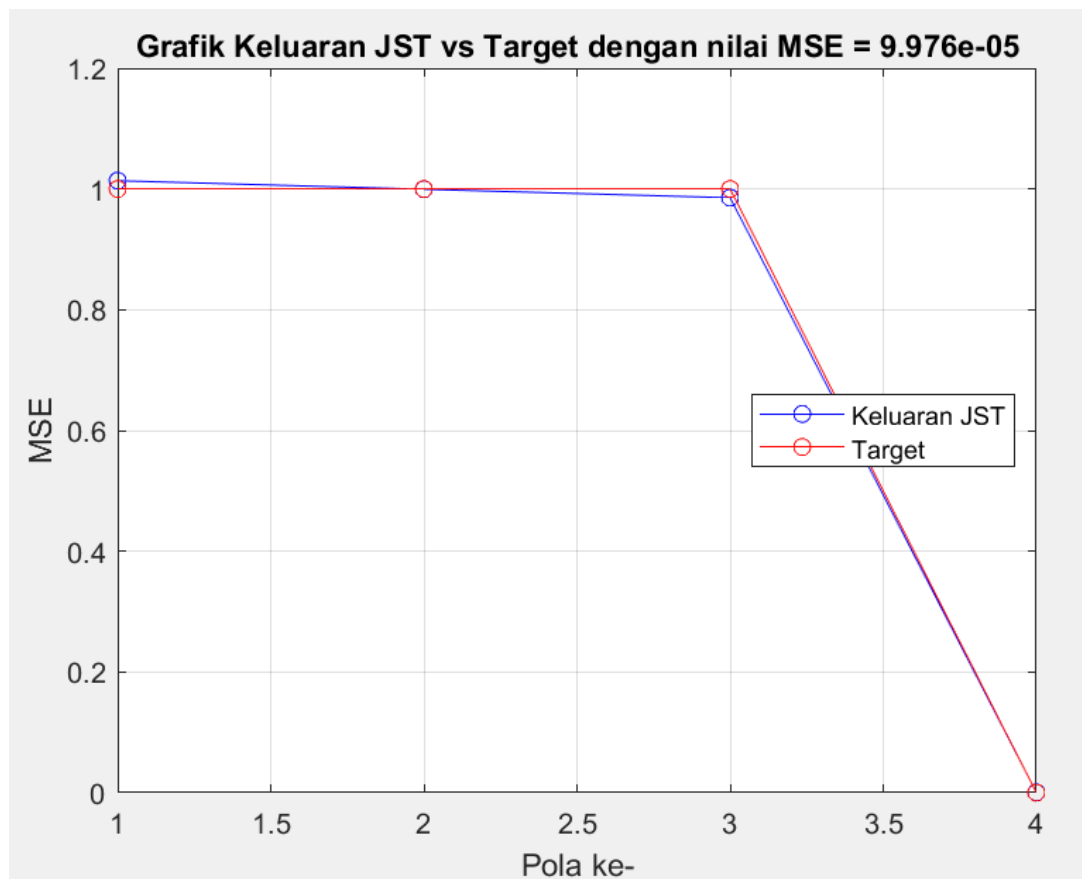
### 13. Regression



#### 14. Performance



#### 15. Grafik Keluaran



## 16. Hasil Prediksi Data Testing

Command Window

nilai\_error =

0.0046      0.0071      0.0013      0.0131

Akurasi =

99.3498

## PYTHON

### 1. Input Data dan Output Data

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 inputs = np.array([[0,1,0],[0,1,1],[0,0,0],[1,0,0],[1,1,1],[1,0,1]])
5 outputs = np.array([[0],[0],[0],[1],[1],[1]])
6
```

### 2. Membuat class Neural Network

```
6
7 class NeuralNetwork:
8     def __init__(self,inputs,outputs):
9         self.inputs = inputs
10        self.outputs = outputs
11        self.weights = np.array([[.50],[.50],[.50]])
12        self.error_history = []
13        self.epoch_list = []
14
15    def sigmoid(self,x,deriv=False):
16        if deriv == True:
17            return x * (1-x)
18        return 1 / (1+np.exp(-x))
19
20    def feed_forward(self):
21        self.hidden = self.sigmoid(np.dot(self.inputs,self.weights))
22
23    def backpropagation(self):
24        self.error = self.outputs - self.hidden
25        delta = self.error * self.sigmoid(self.hidden, deriv=True)
26        self.weights += np.dot(self.inputs.T, delta)
27
28    def train(self, epochs=25000):
29        for epoch in range(epochs):
30            self.feed_forward()
31            self.backpropagation()
32            self.error_history.append(np.average(np.abs(self.error)))
33            self.epoch_list.append(epoch)
34
35    def predict(self, new_input):
36        prediction = self.sigmoid(np.dot(new_input, self.weights))
37        return prediction
```

### 3. Create Neural Network

```

38
39     NN = NeuralNetwork(inputs,outputs)
40     NN.train()
41

```

#### 4. Testing data

```

41
42     example = np.array([[1,1,0]])
43     example_2= np.array([[0,1,1]])
44
45     print(NN.predict(example), '- Correct:', example[0][0])
46     print(NN.predict(example_2), '- Correct:', example_2[0][0])
47

```

#### 5. Plot Error

```

48     plt.figure(figsize=(15,5))
49     plt.plot(NN.epoch_list,NN.error_history)
50     plt.xlabel('Epoch')
51     plt.ylabel('Error')
52     plt.show()

```

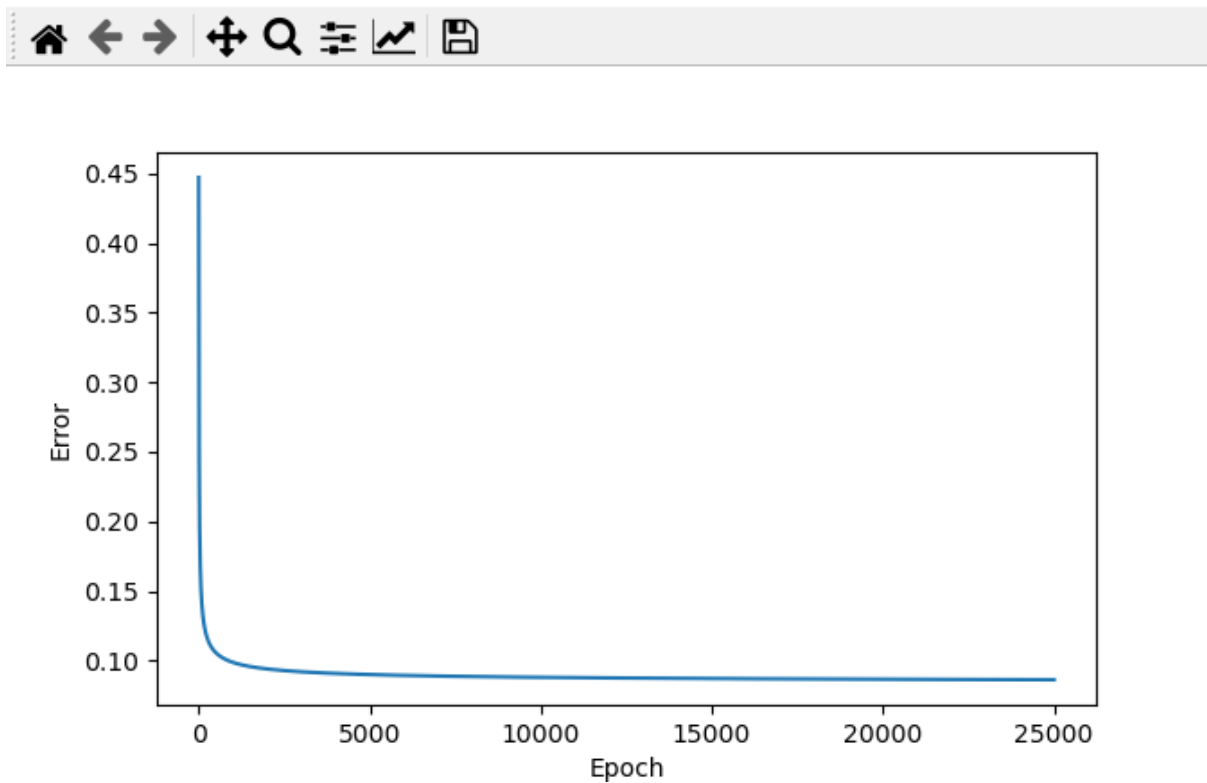
#### Output

```

[[0.99089925]] - Correct: 1
[[0.006409]] - Correct: 0

```

Figure 1



## COBA CODE DENGAN DATA SENDIRI

### MATLAB

#### 1. Data Training

long_hair	lips_thin	gender
0	0	1
1	1	0
1	0	0
1	1	0
1	1	0
1	1	0
1	0	0
0	1	1
1	1	0
1	1	0
1	0	1
1	1	0
1	0	1
1	1	0
1	1	0

#### 2. Membaca data dari excel

```
%Membaca data dari excel
filename = 'gender_training.xlsx';
sheet = 1;
xlRange = 'A2:C76';
Data = xlsread (filename, sheet, xlRange);
data_latih = Data(:,1:2)';
target_latih = Data(:,3)';
[m, n] = size(data_latih);
```

#### 3. Menentukan jumlah neuron, jumlah epoch, dan nilai learning rate lalu menyimpan hasilnya

```
% Pembuatan JST
% Arsitek jaringan yang dipakai adalah 3-3-1, artinya 3 neuron input
% X,Y,Z), 3 neuron hidden (karena Inputnya ada 3 neuron maka neuron hiddennya
% Fungsi Aktivasi di hidden layer menggunakan 'logsig', di output layer menggunakan 'purelin'
% Model JST yang digunakan gradien descent maka fungsi aktivasinya adalah traingdx
net = newff(minmax(data_latih),[2 1],{'logsig', 'purelin'}, 'traingdx');

% Memberikan nilai untuk mempengaruhi proses Training
net.performFcn= 'mse';
net.trainParam.goal = 0.0001; % Errornya (0 sampai 1)
net.trainParam.show = 20; % Boleh diganti
net.trainParam.epochs = 1500; % Banyaknya epoch / iterasi training
net.trainParam.mc = 0.95;
net.trainParam.lr = 1; % Nilai learning Rate (0 sampai 1)
```



```

% Proses training
[net_keluaran, tr, Y, E] = train(net, data_latih, target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW {1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hidden = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2);

save ('C:\ikhshan\UNAIR\SEMESTER 4\PEMBELAJARAN MESIN (PRAKTIKUM)\Tugas SVM dan NN\gender_keluaran.mat')

% Hasil prediksi
hasil_latih = sim(net_keluaran, data_latih);

% Performansi hasil prediksi
target_latih_asli = target_latih;

figure,
plotregression(target_latih_asli, hasil_latih, 'Regression')
figure,
plotperform(tr)

% Gambar JST
figure,
plot(hasil_latih, 'bo-')
hold on
plot(target_latih_asli, 'ro-')
hold off
grid on
title(strcat(['Grafik Keluaran JST vs Target dengan nilai MSE = ', num2str(error_MSE)]))
xlabel('Pola ke-')
ylabel('MSE')
legend('Keluaran JST', 'Target', 'Location', 'Best')

```

#### 4. Data Testing

long_hair	lips_thin	gender
1	1	0
0	1	1
0	1	0
0	1	0
1	0	1
1	1	0
1	1	0
0	0	1
1	1	1
1	0	1
0	1	0
1	1	0
0	1	0
1	1	0
1	0	1

## 5. Load Jaringan Data Training

```
% load jaringan yang sudah dibuat pada proses pelatihan  
load('C:\ikhshan\UNAIR\SEMESTER 4\PEMBELAJARAN MESIN (PRAKTIKUM)\Tugas SVM dan NN\gender_keluaran.mat')
```

## 6. Membaca data dari excel

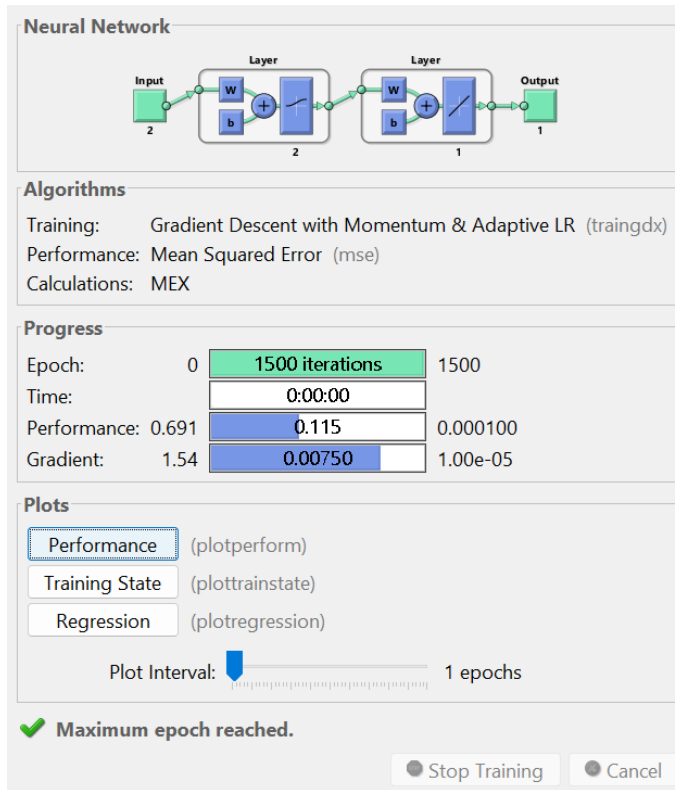
```
% Proses membaca data uji dari excel  
filename = 'gender_test.xlsx';  
sheet = 1;  
xlRange = 'A2:C76';  
Data = xlsread (filename, sheet, xlRange);  
data_uji = Data(:,1:2)';  
target_uji = Data(:,3)';  
[m, n] = size (data_uji);  
  
% Hasil prediksi  
hasil_uji = sim(net_keluaran, data_uji);  
nilai_error = abs(hasil_uji - target_uji)
```

## 7. Hasil prediksi

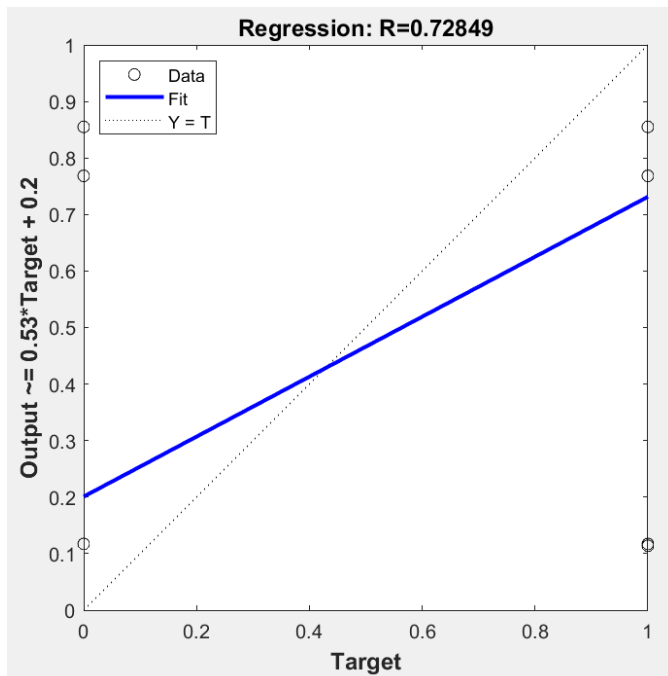
```
% Hasil prediksi  
hasil_uji = sim(net_keluaran, data_uji);  
nilai_error = abs(hasil_uji - target_uji)  
  
% Performansi hasil prediksi  
error = (1/n)*sum(nilai_error.^1);  
Akurasi = (1-error)*100
```

## Output

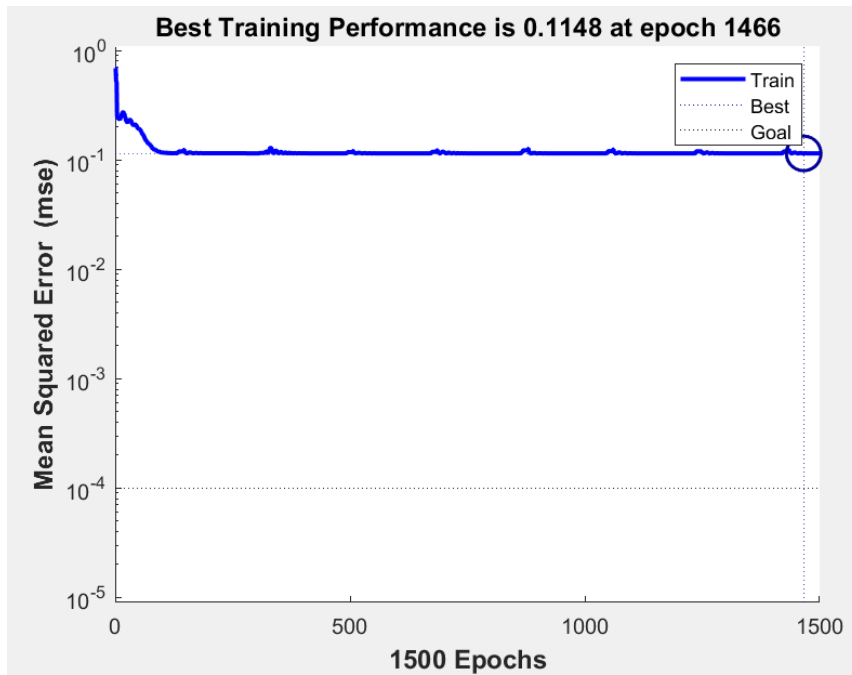
### 1. Hasil Training



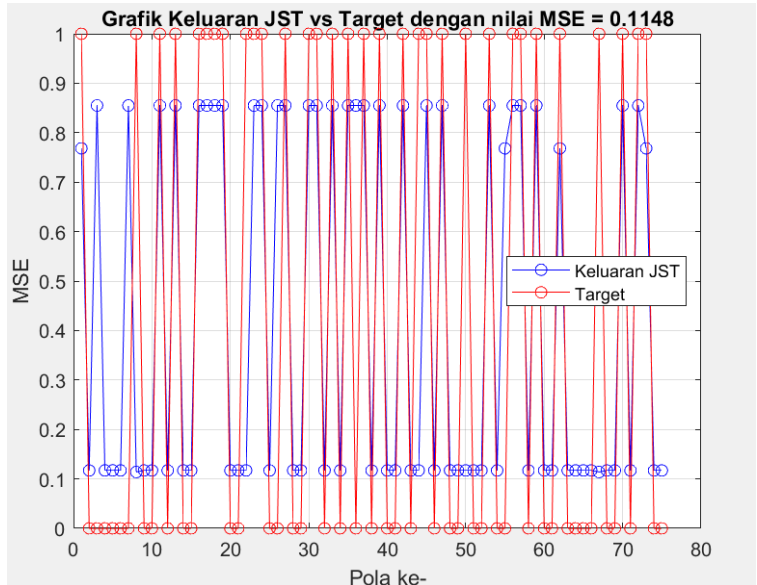
### 2. Regression



### 3. Performance



#### 4. Grafik Keluaran



#### 5. Hasil Prediksi Data Testing

```

nilai_error =

Columns 1 through 11

    0.1171    0.8861    0.1139    0.1139    0.1450    0.1171    0.1171    0.2316    0.8829    0.1450    0.1139

Akurasi =

    74.8639
  
```

# PYTHON

## 1. Input Data dan Output Data

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 inputs = np.array([[1,1,0],[0,0,1],[0,0,0],[0,1,1],
5                   [0,0,0],[0,1,1],[0,0,0],[0,1,1],
6                   [0,0,0],[1,0,1],[0,1,0],[0,0,0],
7                   [1,1,1],[1,1,0],[1,1,0],[1,0,1],
8                   [1,1,0],[0,0,0],[0,0,1],[0,1,0],
9                   [0,0,1],[1,0,1],[0,0,0],[1,1,1],
10                  [1,0,1],[1,1,1],[1,0,0],[0,0,0],
11                  [0,1,0],[0,1,1],[0,0,0],[1,1,1],
12                  [1,0,0],[0,1,1],[0,0,0],[1,1,1],
13                  [1,0,0],[0,1,1],[1,1,0],[0,0,0],
14                  [1,1,0],[1,0,0],[0,0,0],[0,0,0]])
15 outputs = np.array([[1],[0],[1],[0],[1],[0],[1],[0],
16                    [1],[1],[0],[1],[0],[1],[1],[1],
17                    [1],[1],[1],[0],[1],[0],[0],[1],
18                    [1],[1],[1],[1],[0],[0],[0],[0],
19                    [1],[0],[0],[0],[0],[0],[1],[1],
20                    [1],[0],[0],[0]])
21
```

Sebelum memasukkan data input dan data output, kita import dulu numpy dan matplotlib

## 2. Membuat class Neural Network

```

22 class NeuralNetwork:
23     def __init__(self,inputs,outputs):
24         self.inputs = inputs
25         self.outputs = outputs
26         self.weights = np.array([[.50],[.50],[.50]])
27         self.error_history = []
28         self.epoch_list = []
29
30     def sigmoid(self,x,deriv=False):
31         if deriv == True:
32             return x * (1-x)
33         return 1 / (1+np.exp(-x))
34
35     def feed_forward(self):
36         self.hidden = self.sigmoid(np.dot(self.inputs,self.weights))
37
38     def backpropagation(self):
39         self.error = self.outputs - self.hidden
40         delta = self.error * self.sigmoid(self.hidden, deriv=True)
41         self.weights += np.dot(self.inputs.T, delta)
42
43     def train(self, epochs=25000):
44         for epoch in range(epochs):
45             self.feed_forward()
46             self.backpropagation()
47             self.error_history.append(np.average(np.abs(self.error)))
48             self.epoch_list.append(epoch)
49
50     def predict(self, new_input):
51         prediction = self.sigmoid(np.dot(new_input, self.weights))
52         return prediction

```

Membuat kelas neural network dan menginisial variable

### 3. Create Neural Network

```

54 NN = NeuralNetwork(inputs,outputs)
55 NN.train()
56

```

Selanjutnya kita meng create neural network

### 4. Testing data

```

56
57 example = np.array([[1,1,0]])
58 example_2= np.array([[0,1,1]])
59
60 print(NN.predict(example), '- Correct:', example[0][0])
61 print(NN.predict(example_2), '- Correct:', example_2[0][0])

```

Selanjutnya kita membuat 2 contoh untuk prediksi lalu menampilkan hasil prediksinya

### 5. Plot Error

```

62
63 plt.figure(figsize=(15,5))
64 plt.plot(NN.epoch_list,NN.error_history)
65 plt.xlabel('Epoch')
66 plt.ylabel('Error')
67 plt.show()
68

```

Membuat plot errornya

## Output

```

wdir='C:/Users/ACER/Downloads')
[[0.731389]] - Correct: 1
[[0.10038553]] - Correct: 0

```

ini adalah hasil dari 132 data input dan 44 data output

