LAPORAN KECERDASAN BUATAN

"Project Pemrograman Pengenalan Wajah menggunakan Jaringan Syaraf Tiruan Propagasi Balik dengan bahasa Python"

Ditulis oleh:

IKHWAN EL FARIS	(1152200009)
JONATHAN NATANNAEL ZEFANYA	(1152200024)
DAFFA DANINDRA	(1152200028)
DETHALISA AURA KIRANA	(1152200037)

BAB I
PENDAHULUAN

1.1 Masalah

a. Bagaimana cara mengimplementasikan Jaringan Syaraf Tiruan Propagasi Balik untuk Pengenalan Wajah secara manual dalam Python?
b. Seberapa akurat dan efisien implementasi Jaringan Syaraf Tiruan Propagasi Balik untuk Pengenalan Wajah manual dalam mengklasifikasikan dataset B.J.Habibie dan Soeharto?
c. Bagaimana menyusun kode Python forward pass dan backward pass untuk Jaringan Syaraf Tiruan Propagasi Balik untuk Pengenalan Wajah yang mudah dipahami, dioptimalkan, dan tetap memberikan hasil yang akurat?
d. Bagaimana cara mengolah, menghitung dan menyiapkan dataset Pengenalan Wajah agar dapat digunakan dalam Jaringan Syaraf Tiruan Propagasi Balik?

1.2 Tujuan

a. Mengimplementasikan Jaringan Syaraf Tiruan Propagasi Balik untuk Pengenalan Wajah secara manual dalam Python.
b. Membandingkan akurasi dan efisiensi antara implementasi Jaringan Syaraf Tiruan Propagasi Balik untuk Pengenalan Wajah secara manual.
c. Membuat kode Python untuk Jaringan Syaraf Tiruan Propagasi Balik untuk Pengenalan Wajah secara manual agar memberikan hasil yang akurat dalam menghitung probabilitas dataset B.J.Habibie dan Soeharto.
d. Mengolah dan menyiapkan dataset Pengenalan Wajah agar dapat digunakan dalam perhitungan Jaringan Syaraf Tiruan Propagasi Balik manual dengan benar.

BAB II
DASAR TEORI

2.1 Pengenalan Wajah menggunakan Jaringan Syaraf Tiruan

Pengenalan wajah menggunakan jaringan syaraf tiruan (JST) merupakan model komputasi untuk mengenali dan membedakan wajah manusia. Pengenalan wajah ini berawal dari konsep JST, yang terinspirasi oleh cara kerja otak manusia dalam memproses informasi [ref 1]. JST terdiri dari neuron-neuron buatan yang saling terhubung dalam layer-layer, yakni input layer, hidden layer, dan output layer [ref 2]. Pada pengenalan wajah, gambar wajah yang ingin dikenali akan diolah menjadi data numerik dan dimasukkan ke dalam input layer [ref 3]. JST kemudian memproses data ini melalui hidden layer dengan menerapkan berbagai transformasi dan fungsi aktivasi [ref 4]. Hasil akhirnya adalah keluaran dari output layer yang memberikan klasifikasi atau identifikasi dari wajah tersebut [ref 3]. Teknologi ini mampu meningkatkan akurasi pengenalan wajah secara signifikan [ref 5].

BAB III
DATA DAN METODOLOGI

3.1 Data

3.1.1 Sumber Data

Sumber data berasal dari foto B.J.Habibie dan foto Soeharto.

a. Foto B.J.Habibie

Gambar 3.1 Foto B.J.Habibie

b. Foto Soeharto

Gambar 3.2 Foto Soeharto

3.1.2 Pre-processing Data

a. Data Testing Sebelum Diubah ke csv

Data Testing								
No Data	Lebar Mulut/Jarak Pupil Mata	Lebar Dagu/Jarak Pupil Mata	Lebar Pipi/Jarak Pupil Mata	Lebar Dahi/Jarak Pupil Mata	Lebar Hidung/Jarak Pupil Mata	Lebar Mata/Jarak Pupil Mata	Tinggi Hidung/Jarak Pupil Mata	Label
1	3/3,5 = 0,86	3,4/3,5 = 0,97	6/3,5 = 1,71	5/3,5=1,43	2,5/3,5=0,71	4,7/3,5=1,34	3,4/3,5=0,97	B.J.Habibie
2	3/4,1 = 0,73	3,7/4,1 = 0,90	6,9/4,1 = 1,68	6/4,1=1,46	2,5/4,1=0,61	5/4,1=1,22	3,2/4,1=0,78	B.J.Habibie
3	3,5/4,7 = 0,74	3/4,7 = 0,64	8/4,7 = 1,70	6,5/4,7=1,38	3/4,7=0,64	6/4,7=1,28	3,5/4,7=0,74	B.J.Habibie
4	3,8/5 = 0,76	4/5 = 0,80	7,6/5 = 1,52	7/5=1,40	2,9/5=0,58	6,2/5=1,24	4/5=0,80	B.J.Habibie
5	3,3/4,2 = 0,79	3,5/4,2 = 0,83	6,3/4,2 = 1,50	6/4,2=1,43	2,5/4,2=0,60	5,1/4,2=1,21	3,4/4,2=0,81	B.J.Habibie
6	3,6/4,2 = 0,86	4/4,2 = 0,95	7,5/4,2 = 1,79	6/4,2=1,43	2,8/4,2=0,67	5,9/4,2=1,40	4/4,2=0,95	Soeharto
7	5/6,3 = 0,79	6/6,3 = 0,95	10,5/6,3 = 1,67	9/6,3=1,42	4/6,3=0,63	8,5/6,3=1,35	5,5/6,3=0,87	Soeharto
8	5,3/7 = 0,76	6/7 = 0,86	11/7 = 1,57	9/7=1,29	4,7/7=0,67	9,8/7=1,40	6/7=0,86	Soeharto
9	4/4,6 = 0,87	5/4,6 = 1,09	8/4,6 = 1,74	8/4,6=1,74	3/4,6=0,65	6,6/4,6=1,43	4,6/4,6=1,00	Soeharto
10	4/4,4 = 0,91	4/4,4 = 0,91	7/4,4 = 1,59	7/4,4=1,59	4,8/4,4=1,09	6,3/4,4=1,43	4,3/4,4=0,98	Soeharto

Gambar 3.3 Data Testing Sebelum Dirubah ke csv

b. Data Training Sebelum Diubah ke csv

Data Training								
No Data	Lebar Mulut/Jarak Pupil Mata	Lebar Dagu/Jarak Pupil Mata	Lebar Pipi/Jarak Pupil Mata	Lebar Dahi/Jarak Pupil Mata	Lebar Hidung/Jarak Pupil Mata	Lebar Mata/Jarak Pupil Mata	Tinggi Hidung/Jarak Pupil Mata	Label
1	2,6/2=1,30	2,4/2=1,20	4,2/2=2,10	4/2=2,00	1,7/2=0,85	3,3/2=1,65	2/2=1,00	B.J.Habibie
2	3,5/4,5=0,78	4,2/4,5=0,93	7/4,5=1,56	6/4,5=1,33	2,6/4,5=0,58	5,5/4,5=1,22	3,5/4,5=0,78	B.J.Habibie
3	4,1/5,4=0,76	4,3/5,4=0,80	9/5,4=1,67	8/5,4=1,48	3,1/5,4=0,57	6,9/5,4=1,28	4,5/5,4=0,83	B.J.Habibie
4	3,5/4,2=0,83	4,2/4,2=1,00	6,9/4,2=1,64	6,5/4,2=1,55	2,5/4,2=0,60	5,7/4,2=1,36	3/4,2=0,71	B.J.Habibie
5	4,2/5,6=0,75	6/5,6=1,07	8/5,6=1,43	8,5/5,6=1,52	3,1/5,6=0,55	7/5,6=1,25	4,5/5,6=0,80	B.J.Habibie
6	6,6/8=0,83	8/8=1,00	13/8=1,63	10/8=1,25	5/8=0,63	10,5/8=1,31	6,7/8=0,84	B.J.Habibie
7	3/4,5=0,67	4,5/4,5=1,00	6,5/4,5=1,44	7/4,5=1,56	2,3/4,5=0,51	6/4,5=1,33	3,5/4,5=0,78	B.J.Habibie
8	2,7/3,5=0,77	3/3,5=0,86	6,2/3,5=1,77	5,5/3,5=1,57	2/3,5=0,57	4,5/3,5=1,29	3/3,5=0,86	B.J.Habibie
9	3,6/4,5=0,80	4/4,5=0,89	7,2/4,5=1,60	7/4,5=1,56	2,9/4,5=0,64	6/4,5=1,33	3,5/4,5=0,78	B.J.Habibie
10	2,5/3=0,83	2,5/3=0,83	5,5/3=1,83	6/3=2,00	2/3=0,67	3,7/3=1,23	3/3=1,00	B.J.Habibie
11	4,7/5=0,94	4,5/5=0,90	8/5=1,60	8/5=1,60	3/5=0,60	6,5/5=1,30	4,5/5=0,90	B.J.Habibie
12	3,3/4,1=0,80	4/4,1=0,98	6,7/4,1=1,63	7/4,1=1,71	2,7/4,1=0,66	5,4/4,1=1,32	3,5/4,1=0,85	B.J.Habibie
13	4/4,9=0,82	4,5/4,9=0,92	7,5/4,9=1,53	8/4,9=1,63	3/4,9=0,61	6/4,9=1,22	4,7/4,9=0,96	B.J.Habibie
14	2,6/3,7=0,70	3/3,7=0,81	6,7/3,7=1,81	7/3,7=1,89	2/3,7=0,54	5,2/3,7=1,41	3,7/3,7=1,00	B.J.Habibie
15	3/4=0,75	3,2/4=0,80	6,5/4=1,63	5,8/4=1,45	2,1/4=0,53	5/4=1,25	3,5/4=0,88	B.J.Habibie
16	2,7/3,8=0,71	3,7/3,8=0,97	6/3,8=1,58	5/3,8=1,32	2/3,8=0,53	4,5/3,8=1,18	3/3,8=0,79	B.J.Habibie
17	2,3/3,2=0,72	3,9/3,2=1,22	5/3,2=1,56	5/3,2=1,56	2,8/3,2=0,88	3,8/3,2=1,19	4,8/3,2=1,50	B.J.Habibie
18	4/6=0,67	5,5/6=0,92	10/6=1,67	8/6=1,33	3/6=0,50	7,5/6=1,25	4,5/6=0,75	B.J.Habibie
19	3,5/4,2=0,83	3,5/4,2=0,83	7/4,2=1,67	7/4,2=1,67	2,6/4,2=0,62	5,5/4,2=1,31	3,7/4,2=0,88	B.J.Habibie
20	3/4,6=0,65	4,3/4,6=0,93	7,5/4,6=1,63	6/4,6=1,30	2,6/4,6=0,57	5,7/4,6=1,24	4/4,6=0,87	B.J.Habibie
21	1,5/2,5=0,60	2/2,5=0,80	4/2,5=1,60	4/2,5=1,60	1,2/2,5=0,48	3/2,5=1,20	1,7/2,5=0,68	B.J.Habibie
22	4/5=0,80	5/5=1,00	8/5=1,60	8/5=1,60	3/5=0,60	6,5/5=1,30	4/5=0,80	B.J.Habibie
23	3/4=0,75	3,5/4=0,88	6,5/4=1,63	5,5/4=1,38	2,4/4=0,60	4,5/4=1,13	3,5/4=0,88	B.J.Habibie
24	3/4,5=0,67	4/4,5=0,89	8/4,5=1,78	7/4,5=1,56	5,5/4,5=1,22	2,5/4,5=0,56	4/4,5=0,89	B.J.Habibie
25	2,7/3,2=0,84	3,2/3,2=1,00	5/3,2=1,56	5/3,2=1,56	2/3,2=0,63	4,2/3,2=1,32	2,5/3,2=0,78	B.J.Habibie

Gambar 3.4 Data Training Sebelum Dirubah ke csv

c. Data Testing:

```
Lebar Mulut/Jarak Pupil Mata,Lebar Dagu/Jarak Pupil Mata,Lebar Pipi/Jarak Pupil Mata,Lebar Dahi/Jarak Pupil Mata,Lebar Hidung/Jarak Pupil Mata,Lebar Mata/Jarak Pupil Mata,Tinggi Hidung/Jarak Pupil Mata,Label
0.86,0.97,1.71,1.43,0.71,1.34,0.97,B.J.Habibie
0.73,0.90,1.68,1.46,0.61,1.22,0,78,B.J.Habibie
0.74,0.64,1.70,1.38,0.64,1.28,0.74,B.J.Habibie
0.76,0.80,1.52,1.40,0.58,1.24,0.80,B.J.Habibie
0.79,0.83,1.50,1.43,0.60,1.21,0.81,B.J.Habibie
0.86,0.95,1.79,1.43,0.67,1.40,0.95,Soeharto
0.80,0.95,1.67,1.43,0.63,1.35,0,87,Soeharto
0.76,0.86,1.57,1.30,0.67,1.40,0.86,Soeharto
0.87,1.09,1.74,1.74,0.65,1.43,1.00,Soeharto
0.91,0.91,1.59,1.59,1.09,1.43,0.98,Soeharto
```

Gambar 3.5 Data Testing

d. Data Training:

```
Lebar Mulut/Jarak Pupil Mata,Lebar Dagu/Jarak Pupil Mata,Lebar Pipi/Jarak Pupil Mata,Lebar Dahi/Jarak Pupil Mata,Lebar Hidung/Jarak Pupil Mata,Lebar Mata/Jarak Pupil Mata,Tinggi Hidung/Jarak Pupil Mata,Label
1.30,1.20,2.10,2.00,0.85,1.65,1.00,B.J.Habibie
0.78,0.93,1.56,1.33,0.58,1.22,0.78,B.J.Habibie
0.76,0.80,1.67,1.48,0.57,1.28,0.83,B.J.Habibie
0.83,1.00,1.64,1.55,0.60,1.36,0.71,B.J.Habibie
0.75,1.07,1.43,1.52,0.55,1.25,0.80,B.J.Habibie
0.83,1.00,1.63,1.25,0.63,1.31,0.83,B.J.Habibie
0.67,1.00,1.44,1.56,0.51,1.33,0.78,B.J.Habibie
0.77,0.86,1.77,1.57,0.57,1.29,0.86,B.J.Habibie
0.80,0.89,1.60,1.56,0.64,1.33,0.78,B.J.Habibie
0.83,0.83,1.83,2.00,0.67,1.23,1.00,B.J.Habibie
0.94,0.90,1.60,1.60,0.60,1.30,0.90,B.J.Habibie
0.80,0.98,1.63,1.70,0.66,1.32,0.85,B.J.Habibie
0.82,0.92,1.53,1.63,0.61,1.22,0.96,B.J.Habibie
0.70,0.81,1.81,1.89,0.54,1.41,1.00,B.J.Habibie
0.75,0.80,1.63,1.45,0.53,1.25,0.88,B.J.Habibie
0.71,0.97,1.58,1.32,0.53,1.18,0.79,B.J.Habibie
0.72,1.22,1.56,1.56,0.88,1.19,1.50,B.J.Habibie
0.67,0.92,1.67,1.33,0.50,1.25,0.75,B.J.Habibie
0.83,0.83,1.67,1.67,0.62,1.31,0.88,B.J.Habibie
0.65,0.93,1.63,1.30,0.57,1.24,0.87,B.J.Habibie
0.60,0.80,1.60,1.60,0.48,1.20,0.68,B.J.Habibie
0.80,1.00,1.60,1.60,0.60,1.30,0.80,B.J.Habibie
0.75,0.88,1.63,1.38,0.60,1.13,0.88,B.J.Habibie
0.67,0.89,1.78,1.56,1.22,0.56,0.89,B.J.Habibie
0.84,1.00,1.56,1.56,0.63,1.31,0.78,B.J.Habibie
```

Gambar 3.6 Data Training

3.2 Metodologi

3.2.1 Mengupload Data

pada bagian ini untuk mengupload data training dan data testing.

```
[1]  from google.colab import files

     # Upload training and testing data
     print("Upload training")
     uploaded_train = files.upload()
     print("Upload testing")
     uploaded_test = files.upload()
```

Gambar 3.7 Mengupload Data

3.2.2 Membuat table saat membaca data

pada bagian ini untuk membuat table saat sedang membaca data yang diupload.

```
[2]  # Fungsi untuk membuat tabel saat membaca data

     #Source: https://www.geeksforgeeks.org/how-to-render-pandas-dataframe-as-html-table/
     from IPython.display import display, HTML

     def tabel(df):
         display(HTML(df.to_html(index=False)))
```

Gambar 3.8 membuat table saat membaca data

3.2.3 Membaca dan Menampilkan data training dan data testing

pada bagian ini untuk membaca dan menampilkan data training dan data testing yang telah diupload.

```
# Membaca dan Menampilkan data training dan data testing
import pandas as pd

#Data Training
# Membaca
train_data = pd.read_csv(next(iter(uploaded_train.keys())))
# Menampilkan
print("Data Training:")
# Function Tabel untuk menampilkan dalam bentuk tabel
tabel(train_data)

# Data Testing
# Membaca
test_data = pd.read_csv(next(iter(uploaded_test.keys())),on_bad_lines='skip') # saat tidak memakai variabel "on_bad_lines" akan terjadi error, source penyelesaian: https://stackoverflow.com/questions/18039057/p
# Menampilkan
print("Data Testing:")
# Function Tabel untuk menampilkan dalam bentuk tabel
tabel(test_data)
```

Gambar 3.9 Membaca dan Menampilkan data training dan data testing.

3.2.4 Memisahkan ciri dan label training dan testing

pada bagian ini untuk memisahkan ciri dan label dari data training dan data testing.

```
[4]  # Pisahkan ciri dan label training
     # Referensi ".iloc": https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iloc.html
     X_train = train_data.iloc[:, :-1].values
     y_train = train_data.iloc[:, -1].astype(str).values

[5]  # Pisahkan ciri dan label testing
     # Referensi ".iloc": https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iloc.html
     X_test = test_data.iloc[:, :-1].values
     y_test = test_data.iloc[:, -1].astype(str).values
```

Gambar 3.10 Memisahkan ciri dan label training dan testing

3.2.5 Encode label

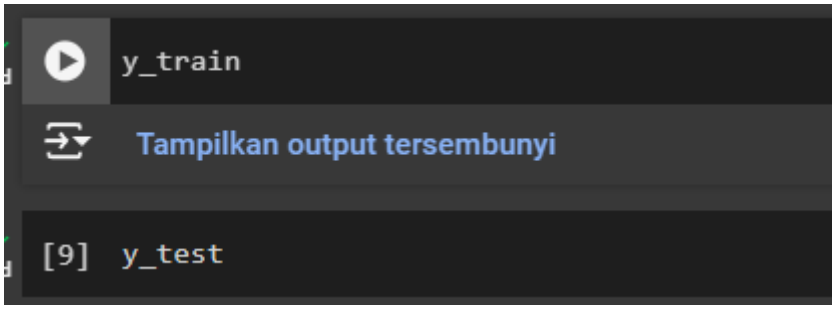
pada bagian ini untuk mengencode seluruh label dari setiap jenis data.

```
[6]  # Function encode label
     import numpy as np
     def encode_labels(labels):
         encoded_labels = []
         for label in labels:
             if label == 'B.J.Habibie': #y_input
                 encoded_labels.append(0.8677)
             else:
                 encoded_labels.append(0)
         return np.array(encoded_labels)

[7]  # encode label tiap jenis data
     y_train = encode_labels(y_train)
     y_test = encode_labels(y_test)
```

Gambar 3.11 Encode label

3.2.6 Menampilkan hasil encode label

pada bagian ini untuk menampilkan hasil dari encode label



Gambar 3.12 Menampilkan hasil encode label

3.2.7 Inisialisasi Parameter JST

Pada bagian ini untuk menentukan input, hidden, output layer.

```
[10] # Inisialisasi parameter JST
     input_neurons = X_train.shape[1]
     hidden_neurons = int(2/3 * input_neurons)
     output_neurons = 1
```

Gambar 3.13 Inisialisasi Parameter JST

3.2.8 Inisialisasi Bobot

pada bagian ini untuk menentukan nilai bobot.

```
[ ]  # Inisialisasi bobot

     # Bobot dari input ke hidden
     weights_input = np.array([[0.2, 0.3, 0.4, 0.5],
                               [0.3, 0.4, 0.5, 0.6],
                               [0.4, 0.5, 0.6, 0.7],
                               [0.5, 0.6, 0.7, 0.8],
                               [0.6, 0.7, 0.8, 0.9],
                               [0.7, 0.8, 0.9, 0.3],
                               [0.8, 0.9, 0.3, 0.5]])

     # Bobot dari hidden ke output
     weights_output = np.array([[0.3], [0.4], [0.5], [0.6]])
```

Gambar 3.14 Inisialisasi Bobot

3.2.9 Inisialisasi Bias dan Hiperparameter

Pada bagian ini untuk menentukan biasnya.

```
[12] # Inisialisasi bias
     b1 = 0.05
     b2 = 0.1
```

Gambar 3.15 Inisialisasi Bias

```
[13] # Hiperparameter
     eta = 0.5
     max_perulangan = 3
```

Gambar 3.16 Hiperparameter

3.2.10 Aktivasi h_output

Pada bagian ini untuk melakukan aktivasi terhadap h_output.

```
# Fungsi aktivasi h_output
def h_output(x):
    return 1 / (1 + np.exp(-x))
```

Gambar 3.17 Aktivasi h_output

3.2.11 Forward Pass

pada bagian ini untuk Fungsi dari Forward Pass.

```
# Fungsi untuk forward pass
def forward_pass(X):
    net_h = np.dot(X, weights_input) + b1 * 1
    out_h = h_output(net_h)
    net_o = np.dot(out_h, weights_output) + b2 * 1
    out_o = h_output(net_o)
    return net_h, out_h, net_o, out_o
```

Gambar 3.18 Forward Pass

3.2.12 Turunan H_output terhadap Net_output

Pada bagian ini untuk menghitung turunan H_output terhadap Net_output.

```
[16] # Turunan H_output terhadap Net_output
     def TurunanH_OutputTerhadapNetO(x):
         return x * (1 - x)
```

Gambar 3.19 Turunan H_output terhadap Net_output

3.2.13 Backward Pass

Pada bagian ini untuk Fungsi dari Backward Pass.

```
# Referensi: https://www.geeksforgeeks.org/backpropagation-in-neural-network/
# Fungsi untuk backward pass
def backward_pass(X, y, net_h, out_h, net_o, out_o):
    global weights_output, weights_input

    # Menghitung total error
    total_error = 0.5 * (y - out_o) ** 2

    # Turunan total error terhadap output
    a_total_error_d_out_o = -(y - out_o)

    # Turunan output terhadap net output
    a_out_o_d_net_o = TurunanH_OutputTerhadapNetO (out_o)

    # Turunan net output terhadap bobot output
    a_net_o_d_w_output = out_h.T

    # Turunan total error terhadap bobot output
    a_total_error_d_w_output = a_total_error_d_out_o * a_out_o_d_net_o * a_net_o_d_w_output

    # Turunan net output terhadap hidden output
    a_net_o_d_out_h = weights_output.T

    # Turunan total error terhadap net hidden
    a_total_error_d_net_h = (a_total_error_d_out_o * a_out_o_d_net_o).dot(a_net_o_d_out_h)

    # Turunan hidden output terhadap net hidden
    a_out_h_d_net_h = out_h * (1 - out_h)

    # Turunan total error terhadap bobot input
    a_total_error_d_w_input = X.T.dot(a_total_error_d_net_h * a_out_h_d_net_h)

    # Memperbarui bobot input dan output
    weights_output -= eta * a_total_error_d_w_output
    weights_input -= eta * a_total_error_d_w_input

    return total_error
```

Gambar 3.20 Backward Pass

3.2.14 Forward dan Backward Data Training

Pada bagian ini memiliki fungsi untuk forward dan backward pada data training.

```
# Training JST
for i in range(len(X_train)):
    print("=========================================================================================================")
    print(f"Data Latih ke-{i + 1}")
    forward_backward_repeats = 0
    while forward_backward_repeats < max_perulangan:
        # Cari nilai net_h, out_h, net_o, out_o
        net_h, out_h, net_o, out_o = forward_pass(X_train[i].reshape(1, -1))

        # Ambang Batas
        predicted_label = out_o

        # Menampilkan hasil forward pass
        print(f"Forward-backward pass {forward_backward_repeats + 1}")
        print(f"net_h: {net_h}")
        print(f"out_h: {out_h}")
        print(f"net_o: {net_o}")
        print(f"out_o: {out_o}")
        print(f"y_input: 0.8677")
        print(f"f(x): {out_o}")
        print(f"bobot input: {weights_input}")
        print(f"bobot output: {weights_output}")

        # Ambang Batas
        if predicted_label >= y_train[i]:
            print("Forward Pass ke foto selanjutnya.")
            break
        else:
            print(f"Backward Pass karena y_input = {out_o} < {y_train[i]}.")
            total_error = backward_pass(X_train[i].reshape(1, -1), y_train[i].reshape(1, -1), net_h, out_h, net_o, out_o)

            # Menampilkan hasil backward pass
            print(f"Total Error: {total_error}")
            print(f"Update Bobot input: {weights_input}")
            print(f"Updated Bobot output: {weights_output}")
            forward_backward_repeats += 1
            if forward_backward_repeats == max_perulangan:
                print("Skip, Foto Selanjutnya")
    print()
```

Gambar 3.21 Forward dan Backward Data Training

3.2.15 Forward Data Testing

Pada bagian ini memiliki fungsi untuk Forward pada data testing.

```
[20] # Pengujian Model
     correct_predictions = 0
     print("Testing Data Results:")
     for i in range(len(X_test)):
         net_h, out_h, net_o, out_o = forward_pass(X_test[i].reshape(1, -1))
         predicted_label = out_o

         # Menampilkan hasil perhitungan
         print(f"Data Testing Ke-{i + 1}")
         print(f"net_h: {net_h}")
         print(f"out_h: {out_h}")
         print(f"net_o: {net_o}")
         print(f"out_o: {out_o}")
         print(f"y_input: 0.8677")
         print(f"f(x): {out_o}")
         print(f"bobot input: {weights_input}")
         print(f"bobot output: {weights_output}")


         if predicted_label >= y_test[i]:
             correct_predictions += 1
             print(f"Hasil pengenalan foto ke-{i+1}: Benar")
         else:
             print(f"Hasil pengenalan foto ke-{i+1}: Salah")
         print()
```

Gambar 3.22 Forward Data Testing

3.2.16 Menghitung Akurasi

Pada bagian ini memiliki fungsi untuk menghitung akurasinya.

```
# Hitung akurasi
accuracy = correct_predictions / len(y_test)
print(f'Test Accuracy: {accuracy:.4f}')
```

Gambar 3.23 Menghitung Akurasi

3.2.17 Mengubah Hasil Perhitungan menjadi TXT

```
[19] # Training JST
     with open('training.txt', 'w') as train_file:
         for i in range(len(X_train)):
             train_file.write("=============================================================================================================\n")
             train_file.write(f"Data Latih ke-{i + 1}\n")
             forward_backward_repeats = 0
             while forward_backward_repeats < max_perulangan:
                 # Cari nilai net_h, out_h, net_o, out_o
                 net_h, out_h, net_o, out_o = forward_pass(X_train[i].reshape(1, -1))

                 # Ambang Batas
                 predicted_label = out_o

                 # Menampilkan hasil forward pass
                 train_file.write(f"Forward-backward pass {forward_backward_repeats + 1}\n")
                 train_file.write(f"net_h: {net_h}\n")
                 train_file.write(f"out_h: {out_h}\n")
                 train_file.write(f"net_o: {net_o}\n")
                 train_file.write(f"out_o: {out_o}\n")
                 train_file.write("y_input: 0.8677\n")
                 train_file.write(f"f(x): {out_o}\n")
                 train_file.write(f"bobot input: {weights_input}\n")
                 train_file.write(f"bobot output: {weights_output}\n")

                 # Ambang Batas
                 if predicted_label >= y_train[i]:
                     train_file.write("Forward Pass ke foto selanjutnya.\n")
                     break
                 else:
                     train_file.write(f"Backward Pass karena y_input = {out_o} < {y_train[i]}.\n")
                     total_error = backward_pass(X_train[i].reshape(1, -1), y_train[i].reshape(1, -1), net_h, out_h, net_o, out_o)

                     # Menampilkan hasil backward pass
                     train_file.write(f"Total Error: {total_error}\n")
                     train_file.write(f"Update Bobot input: {weights_input}\n")
                     train_file.write(f"Updated Bobot output: {weights_output}\n")
                     forward_backward_repeats += 1
             train_file.write("\n")
```

Gambar 3.24 Mengubah Hasil Perhitungan Data Training menjadi TXT

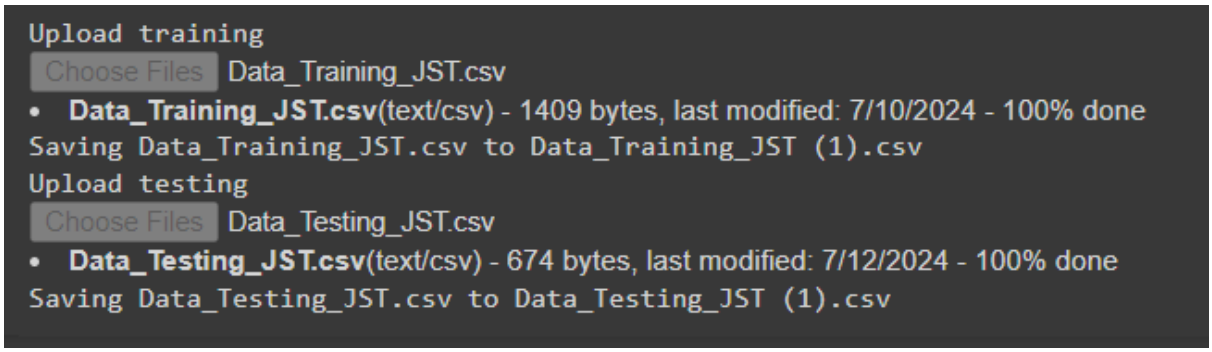
```
[20] # Pengujian Model
     correct_predictions = 0
     print("Testing Data Results:")
     for i in range(len(X_test)):
         net_h, out_h, net_o, out_o = forward_pass(X_test[i].reshape(1, -1))
         predicted_label = out_o

         # Menampilkan hasil perhitungan
         print(f"Data Testing Ke-{i + 1}")
         print(f"net_h: {net_h}")
         print(f"out_h: {out_h}")
         print(f"net_o: {net_o}")
         print(f"out_o: {out_o}")
         print(f"y_input: 0.8677")
         print(f"f(x): {out_o}")
         print(f"bobot input: {weights_input}")
         print(f"bobot output: {weights_output}")


         if predicted_label >= y_test[i]:
             correct_predictions += 1
             print(f"Hasil pengenalan foto ke-{i+1}: Benar")
         else:
             print(f"Hasil pengenalan foto ke-{i+1}: Salah")
         print()
```

Gambar 3.25 Mengubah Hasil Perhitungan Data Testing menjadi TXT

BAB IV
HASIL DAN ANALISA

4.1 Hasil

4.1.1 Hasil Mengupload Data



Gambar 4.1 Hasil Mengupload Data

4.1.2 Hasil Menampilkan Data Testing

Data Testing:

Lebar Mulut/Jarak Pupil Mata	Lebar Dagu/Jarak Pupil Mata	Lebar Pipi/Jarak Pupil Mata	Lebar Dahi/Jarak Pupil Mata	Lebar Hidung/Jarak Pupil Mata	Lebar Mata/Jarak Pupil Mata	Tinggi Hidung/Jarak Pupil Mata	Label
0.86	0.97	1.71	1.43	0.71	1.34	0.97	B.J.Habibie
0.73	0.90	1.68	1.46	0.61	1.22	0.78	B.J.Habibie
0.74	0.64	1.70	1.38	0.64	1.28	0.74	B.J.Habibie
0.76	0.80	1.52	1.40	0.58	1.24	0.80	B.J.Habibie
0.79	0.83	1.50	1.43	0.60	1.21	0.81	B.J.Habibie
0.86	0.95	1.79	1.43	0.67	1.40	0.95	Soeharto
0.80	0.95	1.67	1.43	0.63	1.35	0.87	Soeharto
0.76	0.86	1.57	1.30	0.67	1.40	0.86	Soeharto
0.87	1.09	1.74	1.74	0.65	1.43	1.00	Soeharto
0.91	0.91	1.59	1.59	1.09	1.43	0.98	Soeharto

Gambar 4.2 Hasil Menampilkan Data Testing

4.1.3 Hasil Menampilkan Data Training

Data Training:

Lebar Mulut/Jarak Pupil Mata	Lebar Dagu/Jarak Pupil Mata	Lebar Pipi/Jarak Pupil Mata	Lebar Dahi/Jarak Pupil Mata	Lebar Hidung/Jarak Pupil Mata	Lebar Mata/Jarak Pupil Mata	Tinggi Hidung/Jarak Pupil Mata	Label
1.30	1.20	2.10	2.00	0.85	1.65	1.00	B.J.Habibie
0.78	0.93	1.56	1.33	0.58	1.22	0.78	B.J.Habibie
0.76	0.80	1.67	1.48	0.57	1.28	0.83	B.J.Habibie
0.83	1.00	1.64	1.55	0.60	1.36	0.71	B.J.Habibie
0.75	1.07	1.43	1.52	0.55	1.25	0.80	B.J.Habibie
0.83	1.00	1.63	1.25	0.63	1.31	0.83	B.J.Habibie
0.67	1.00	1.44	1.56	0.51	1.33	0.78	B.J.Habibie
0.77	0.86	1.77	1.57	0.57	1.29	0.86	B.J.Habibie
0.80	0.89	1.60	1.56	0.64	1.33	0.78	B.J.Habibie
0.83	0.83	1.83	2.00	0.67	1.23	1.00	B.J.Habibie
0.94	0.90	1.60	1.60	0.60	1.30	0.90	B.J.Habibie
0.80	0.98	1.63	1.70	0.66	1.32	0.85	B.J.Habibie
0.82	0.92	1.53	1.63	0.61	1.22	0.96	B.J.Habibie
0.70	0.81	1.81	1.89	0.54	1.41	1.00	B.J.Habibie
0.75	0.80	1.63	1.45	0.53	1.25	0.88	B.J.Habibie
0.71	0.97	1.58	1.32	0.53	1.18	0.79	B.J.Habibie
0.72	1.22	1.56	1.56	0.88	1.19	1.50	B.J.Habibie
0.67	0.92	1.67	1.33	0.50	1.25	0.75	B.J.Habibie
0.83	0.83	1.67	1.67	0.62	1.31	0.88	B.J.Habibie
0.65	0.93	1.63	1.30	0.57	1.24	0.87	B.J.Habibie
0.60	0.80	1.60	1.60	0.48	1.20	0.68	B.J.Habibie
0.80	1.00	1.60	1.60	0.60	1.30	0.80	B.J.Habibie
0.75	0.88	1.63	1.38	0.60	1.13	0.88	B.J.Habibie
0.67	0.89	1.78	1.56	1.22	0.56	0.89	B.J.Habibie
0.84	1.00	1.56	1.56	0.63	1.31	0.78	B.J.Habibie

Gambar 4.3 Hasil Menampilkan Data Training

4.1.4 Hasil Encode label Data Training

```
array([0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.8677,
       0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.8677,
       0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.8677,
       0.8677])
```

Gambar 4.4 Hasil Encode label Data Training

4.1.5 Hasil Encode label Data Testing

```
array([0.8677, 0.8677, 0.8677, 0.8677, 0.8677, 0.    , 0.    , 0.    ,
       0.    , 0.    ])
```

Gambar 4.5 Hasil Encode label Data Testing

4.1.6 Hasil Forward dan Backward Data Training

```
==========================================================================================================
Data Latih ke-1
Forward-backward pass 1
net_h: [[4.975 5.985 6.295 6.25 ]]
out_h: [[0.99313888 0.9974901  0.99815789 0.99807327]]
net_o: [[1.89486061]]
out_o: [[0.86930874]]
y_input: 0.8677
f(x): [[0.86930874]]
bobot input: [[0.2 0.3 0.4 0.5]
 [0.3 0.4 0.5 0.6]
 [0.4 0.5 0.6 0.7]
 [0.5 0.6 0.7 0.8]
 [0.6 0.7 0.8 0.9]
 [0.7 0.8 0.9 0.3]
 [0.8 0.9 0.3 0.5]]
bobot output: [[0.3]
 [0.4]
 [0.5]
 [0.6]]
Forward Pass ke foto selanjutnya.

==========================================================================================================
Data Latih ke-2
Forward-backward pass 1
net_h: [[3.6   4.318 4.49  4.432]]
out_h: [[0.97340301 0.98684875 0.98890386 0.98824904]]
net_o: [[1.87416176]]
out_o: [[0.86693909]]
y_input: 0.8677
f(x): [[0.86693909]]
bobot input: [[0.2 0.3 0.4 0.5]
 [0.3 0.4 0.5 0.6]
 [0.4 0.5 0.6 0.7]
 [0.5 0.6 0.7 0.8]
 [0.6 0.7 0.8 0.9]
 [0.7 0.8 0.9 0.3]
 [0.8 0.9 0.3 0.5]]
bobot output: [[0.3]
 [0.4]
 [0.5]
 [0.6]]
Backward Pass karena y_input = [[0.86693909]] < 0.8677.
Total Error: [[2.8948927e-07]]
Update Bobot input: [[0.20000027 0.30000018 0.40000019 0.50000024]
 [0.30000032 0.40000021 0.50000022 0.60000028]
 [0.40000053 0.50000036 0.60000038 0.70000048]
```

Gambar 4.6 Hasil Forward dan Backward Data Training

```
 [0.50000045 0.6000003  0.70000032 0.80000041]
 [0.6000002  0.70000013 0.80000014 0.90000018]
 [0.70000042 0.80000028 0.90000029 0.30000037]
 [0.80000027 0.90000018 0.30000019 0.50000024]]
Updated Bobot output: [[0.30004272]
 [0.40004331]
 [0.5000434 ]
 [0.60004337]]
Forward-backward pass 2
net_h: [[3.60000276 4.31800185 4.49000195 4.43200248]]
out_h: [[0.97340308 0.98684877 0.98890388 0.98824907]]
net_o: [[1.87433192]]
out_o: [[0.86695872]]
y_input: 0.8677
f(x): [[0.86695872]]
bobot input: [[0.20000027 0.30000018 0.40000019 0.50000024]
 [0.30000032 0.40000021 0.50000022 0.60000028]
 [0.40000053 0.50000036 0.60000038 0.70000048]
 [0.50000045 0.6000003  0.70000032 0.80000041]
 [0.6000002  0.70000013 0.80000014 0.90000018]
 [0.70000042 0.80000028 0.90000029 0.30000037]
 [0.80000027 0.90000018 0.30000019 0.50000024]]
bobot output: [[0.30004272]
 [0.40004331]
 [0.5000434 ]
 [0.60004337]]
Backward Pass karena y_input = [[0.86695872]] < 0.8677.
Total Error: [[2.74746654e-07]]
Update Bobot input: [[0.20000052 0.30000035 0.40000037 0.50000047]
 [0.30000063 0.40000042 0.50000044 0.60000056]
 [0.40000105 0.5000007  0.60000074 0.70000094]
 [0.5000009  0.6000006  0.70000063 0.8000008 ]
 [0.60000039 0.70000026 0.80000028 0.90000035]
 [0.70000082 0.80000055 0.90000058 0.30000074]
 [0.80000052 0.90000035 0.30000037 0.50000047]]
Updated Bobot output: [[0.30008433]
 [0.4000855 ]
 [0.50008568]
 [0.60008562]]
Forward-backward pass 3
net_h: [[3.60000546 4.31800365 4.49000385 4.4320049 ]]
out_h: [[0.97340315 0.9868488  0.9889039  0.9882491 ]]
net_o: [[1.87449768]]
out_o: [[0.86697784]]
y_input: 0.8677
f(x): [[0.86697784]]
bobot input: [[0.20000052 0.30000035 0.40000037 0.50000047]
```

Gambar 4.7 Hasil Forward dan Backward Data Training

```
 [0.30000063 0.40000042 0.50000044 0.60000056]
 [0.40000105 0.5000007  0.60000074 0.70000094]
 [0.5000009  0.6000006  0.70000063 0.8000008 ]
 [0.60000039 0.70000026 0.80000028 0.90000035]
 [0.70000082 0.80000055 0.90000058 0.30000074]
 [0.80000052 0.90000035 0.30000037 0.50000047]]
bobot output: [[0.30008433]
 [0.4000855 ]
 [0.50008568]
 [0.60008562]]
Backward Pass karena y_input = [[0.86697784]] < 0.8677.
Total Error: [[2.60758249e-07]]
Update Bobot input: [[0.20000078 0.30000052 0.40000055 0.5000007 ]
 [0.30000093 0.40000062 0.50000065 0.60000083]
 [0.40000155 0.50000104 0.6000011  0.70000139]
 [0.50000133 0.60000089 0.70000094 0.80000119]
 [0.60000058 0.70000039 0.80000041 0.90000052]
 [0.70000122 0.80000081 0.90000086 0.30000109]
 [0.80000078 0.90000052 0.30000055 0.5000007 ]]
Updated Bobot output: [[0.30012487]
 [0.40012659]
 [0.50012686]
 [0.60012677]]
Skip, Foto Selanjutnya

==========================================================================================================
Data Latih ke-3
Forward-backward pass 1
net_h: [[3.75200842 4.49100563 4.64900595 4.57500755]]
out_h: [[0.97706767 0.98891489 0.99051963 0.98979891]]
net_o: [[1.87832375]]
out_o: [[0.86741847]]
y_input: 0.8677
f(x): [[0.86741847]]
bobot input: [[0.20000078 0.30000052 0.40000055 0.5000007 ]
 [0.30000093 0.40000062 0.50000065 0.60000083]
 [0.40000155 0.50000104 0.6000011  0.70000139]
 [0.50000133 0.60000089 0.70000094 0.80000119]
 [0.60000058 0.70000039 0.80000041 0.90000052]
 [0.70000122 0.80000081 0.90000086 0.30000109]
 [0.80000078 0.90000052 0.30000055 0.5000007 ]]
bobot output: [[0.30012487]
 [0.40012659]
 [0.50012686]
 [0.60012677]]
Backward Pass karena y_input = [[0.86741847]] < 0.8677.
Total Error: [[3.9629588e-08]]
```

Gambar 4.8 Hasil Forward dan Backward Data Training

```
Update Bobot input: [[0.20000086 0.30000057 0.40000061 0.50000077]
 [0.30000101 0.40000068 0.50000072 0.60000091]
 [0.40000174 0.50000116 0.60000122 0.70000156]
 [0.50000149 0.60000099 0.70000105 0.80000133]
 [0.60000064 0.70000043 0.80000045 0.90000057]
 [0.70000136 0.8000009  0.90000096 0.30000122]
 [0.80000087 0.90000058 0.30000061 0.50000078]]
Updated Bobot output: [[0.30014069]
 [0.4001426 ]
 [0.50014289]
 [0.6001428 ]]
Forward-backward pass 2
net_h: [[3.75200938 4.49100626 4.64900662 4.57500842]]
out_h: [[0.9770677  0.9889149  0.99051963 0.98979892]]
net_o: [[1.87838679]]
out_o: [[0.86742572]]
y_input: 0.8677
f(x): [[0.86742572]]
bobot input: [[0.20000086 0.30000057 0.40000061 0.50000077]
 [0.30000101 0.40000068 0.50000072 0.60000091]
 [0.40000174 0.50000116 0.60000122 0.70000156]
 [0.50000149 0.60000099 0.70000105 0.80000133]
 [0.60000064 0.70000043 0.80000045 0.90000057]
 [0.70000136 0.8000009  0.90000096 0.30000122]
 [0.80000087 0.90000058 0.30000061 0.50000078]]
bobot output: [[0.30014069]
 [0.4001426 ]
 [0.50014289]
 [0.6001428 ]]
Backward Pass karena y_input = [[0.86742572]] < 0.8677.
Total Error: [[3.76146583e-08]]
Update Bobot input: [[0.20000094 0.30000063 0.40000066 0.50000084]
 [0.3000011  0.40000073 0.50000077 0.60000099]
 [0.40000191 0.50000127 0.60000135 0.70000172]
 [0.50000164 0.60000109 0.70000116 0.80000148]
 [0.6000007  0.70000047 0.80000049 0.90000063]
 [0.70000149 0.80000099 0.90000105 0.30000134]
 [0.80000096 0.90000064 0.30000067 0.50000086]]
Updated Bobot output: [[0.30015609]
 [0.4001582 ]
 [0.50015851]
 [0.60015841]]
Forward-backward pass 3
net_h: [[3.75201032 4.49100687 4.64900728 4.57500927]]
out_h: [[0.97706772 0.9889149  0.99051964 0.98979893]]
net_o: [[1.87844821]]
```

Gambar 4.9 Hasil Forward dan Backward Data Training

```
out_o: [[0.86743278]]
y_input: 0.8677
f(x): [[0.86743278]]
bobot input: [[0.20000094 0.30000063 0.40000066 0.50000084]
 [0.3000011  0.40000073 0.50000077 0.60000099]
 [0.40000191 0.50000127 0.60000135 0.70000172]
 [0.50000164 0.60000109 0.70000116 0.80000148]
 [0.6000007  0.70000047 0.80000049 0.90000063]
 [0.70000149 0.80000099 0.90000105 0.30000134]
 [0.80000096 0.90000064 0.30000067 0.50000086]]
bobot output: [[0.30015609]
 [0.4001582 ]
 [0.50015851]
 [0.60015841]]
Backward Pass karena y_input = [[0.86743278]] < 0.8677.
Total Error: [[3.57023496e-08]]
Update Bobot input: [[0.20000102 0.30000068 0.40000072 0.50000092]
 [0.30000118 0.40000079 0.50000083 0.60000106]
 [0.40000209 0.50000139 0.60000147 0.70000187]
 [0.5000018  0.60000119 0.70000127 0.80000161]
 [0.60000076 0.7000005  0.80000053 0.90000068]
 [0.70000162 0.80000108 0.90000114 0.30000146]
 [0.80000104 0.90000069 0.30000073 0.50000094]]
Updated Bobot output: [[0.30017111]
 [0.40017339]
 [0.50017373]
 [0.60017361]]
Skip, Foto Selanjutnya

=============================================================================================================
Data Latih ke-4
Forward-backward pass 1
net_h: [[3.82701163 4.59600773 4.8680082  4.75601045]]
out_h: [[0.97868944 0.99000879 0.99237    0.99147348]]
net_o: [[1.88136309]]
out_o: [[0.86776762]]
y_input: 0.8677
f(x): [[0.86776762]]
bobot input: [[0.20000102 0.30000068 0.40000072 0.50000092]
 [0.30000118 0.40000079 0.50000083 0.60000106]
 [0.40000209 0.50000139 0.60000147 0.70000187]
 [0.5000018  0.60000119 0.70000127 0.80000161]
 [0.60000076 0.7000005  0.80000053 0.90000068]
 [0.70000162 0.80000108 0.90000114 0.30000146]
 [0.80000104 0.90000069 0.30000073 0.50000094]]
```

Gambar 4.10 Hasil Forward dan Backward Data Training

```
bobot output: [[0.30017111]
 [0.40017339]
 [0.50017373]
 [0.60017361]]
Forward Pass ke foto selanjutnya.

==========================================================================================================
Data Latih ke-5
Forward-backward pass 1
net_h: [[3.69801102 4.43500732 4.61200776 4.5540099 ]]
out_h: [[0.9758261  0.98828391 0.99016581 0.9895847 ]]
net_o: [[1.87757728]]
out_o: [[0.8673326]]
y_input: 0.8677
f(x): [[0.8673326]]
bobot input: [[0.20000102 0.30000068 0.40000072 0.50000092]
 [0.30000118 0.40000079 0.50000083 0.60000106]
 [0.40000209 0.50000139 0.60000147 0.70000187]
 [0.5000018  0.60000119 0.70000127 0.80000161]
 [0.60000076 0.7000005  0.80000053 0.90000068]
 [0.70000162 0.80000108 0.90000114 0.30000146]
 [0.80000104 0.90000069 0.30000073 0.50000094]]
bobot output: [[0.30017111]
 [0.40017339]
 [0.50017373]
 [0.60017361]]
Backward Pass karena y_input = [[0.8673326]] < 0.8677.
Total Error: [[6.74912226e-08]]
Update Bobot input: [[0.20000113 0.30000075 0.4000008  0.50000101]
 [0.30000134 0.40000089 0.50000094 0.6000012 ]
 [0.4000023  0.50000153 0.60000162 0.70000206]
 [0.50000202 0.60000134 0.70000142 0.80000181]
 [0.60000084 0.70000056 0.80000059 0.90000075]
 [0.70000181 0.8000012  0.90000127 0.30000162]
 [0.80000116 0.90000077 0.30000082 0.50000104]]
Updated Bobot output: [[0.30019173]
 [0.40019428]
 [0.50019466]
 [0.60019453]]
Forward-backward pass 2
net_h: [[3.6980123  4.43500816 4.61200865 4.55401102]]
out_h: [[0.97582613 0.98828392 0.99016582 0.98958472]]
net_o: [[1.87765951]]
out_o: [[0.86734206]]
y_input: 0.8677
f(x): [[0.86734206]]
```

Gambar 4.11 Hasil Forward dan Backward Data Training

```
bobot input: [[0.20000113 0.30000075 0.4000008  0.50000101]
 [0.30000134 0.40000089 0.50000094 0.6000012 ]
 [0.4000023  0.50000153 0.60000162 0.70000206]
 [0.50000202 0.60000134 0.70000142 0.80000181]
 [0.60000084 0.70000056 0.80000059 0.90000075]
 [0.70000181 0.8000012  0.90000127 0.30000162]
 [0.80000116 0.90000077 0.30000082 0.50000104]]
bobot output: [[0.30019173]
 [0.40019428]
 [0.50019466]
 [0.60019453]]
Backward Pass karena y_input = [[0.86734206]] < 0.8677.
Total Error: [[6.40601298e-08]]
Update Bobot input: [[0.20000124 0.30000082 0.40000087 0.50000111]
 [0.3000015  0.40000099 0.50000105 0.60000134]
 [0.40000251 0.50000166 0.60000176 0.70000224]
 [0.50000225 0.60000149 0.70000157 0.80000201]
 [0.60000092 0.70000061 0.80000065 0.90000082]
 [0.70000199 0.80000132 0.9000014  0.30000178]
 [0.80000128 0.90000085 0.3000009  0.50000114]]
Updated Bobot output: [[0.30021183]
 [0.40021463]
 [0.50021505]
 [0.60021491]]
Forward-backward pass 3
net_h: [[3.69801355 4.43500898 4.6120095  4.55401211]]
out_h: [[0.97582616 0.98828393 0.99016583 0.98958473]]
net_o: [[1.87773961]]
out_o: [[0.86735128]]
y_input: 0.8677
f(x): [[0.86735128]]
bobot input: [[0.20000124 0.30000082 0.40000087 0.50000111]
 [0.3000015  0.40000099 0.50000105 0.60000134]
 [0.40000251 0.50000166 0.60000176 0.70000224]
 [0.50000225 0.60000149 0.70000157 0.80000201]
 [0.60000092 0.70000061 0.80000065 0.90000082]
 [0.70000199 0.80000132 0.9000014  0.30000178]
 [0.80000128 0.90000085 0.3000009  0.50000114]]
bobot output: [[0.30021183]
 [0.40021463]
 [0.50021505]
 [0.60021491]]
Backward Pass karena y_input = [[0.86735128]] < 0.8677.
Total Error: [[6.08038511e-08]]
Update Bobot input: [[0.20000135 0.30000089 0.40000094 0.5000012 ]
 [0.30000165 0.40000109 0.50000115 0.60000147]
 [0.40000271 0.5000018  0.6000019  0.70000242]
```

Gambar 4.12 Hasil Forward dan Backward Data Training

```
 [0.50000246 0.60000163 0.70000172 0.80000219]
 [0.600001   0.70000066 0.8000007  0.90000089]
 [0.70000217 0.80000144 0.90000152 0.30000194]
 [0.80000139 0.90000092 0.30000097 0.50000124]]
Updated Bobot output: [[0.3002314 ]
 [0.40023446]
 [0.50023491]
 [0.60023476]]
Skip, Foto Selanjutnya

==========================================================================================================
Data Latih ke-6
Forward-backward pass 1
net_h: [[3.75201489 4.50000986 4.66701043 4.58101328]]
out_h: [[0.97706782 0.98901316 0.99068721 0.98985938]]
net_o: [[1.87890793]]
out_o: [[0.86748564]]
y_input: 0.8677
f(x): [[0.86748564]]
bobot input: [[0.20000135 0.30000089 0.40000094 0.5000012 ]
 [0.30000165 0.40000109 0.50000115 0.60000147]
 [0.40000271 0.5000018  0.6000019  0.70000242]
 [0.50000246 0.60000163 0.70000172 0.80000219]
 [0.600001   0.70000066 0.8000007  0.90000089]
 [0.70000217 0.80000144 0.90000152 0.30000194]
 [0.80000139 0.90000092 0.30000097 0.50000124]]
bobot output: [[0.3002314 ]
 [0.40023446]
 [0.50023491]
 [0.60023476]]
Backward Pass karena y_input = [[0.86748564]] < 0.8677.
Total Error: [[2.29754221e-08]]
Update Bobot input: [[0.20000142 0.30000094 0.40000099 0.50000126]
 [0.30000173 0.40000115 0.50000121 0.60000154]
 [0.40000285 0.50000188 0.60000199 0.70000254]
 [0.50000257 0.6000017  0.70000179 0.80000229]
 [0.60000105 0.7000007  0.80000074 0.90000094]
 [0.70000228 0.80000151 0.90000159 0.30000203]
 [0.80000146 0.90000097 0.30000102 0.5000013 ]]
Updated Bobot output: [[0.30024344]
 [0.40024664]
 [0.50024712]
 [0.60024696]]
Forward-backward pass 2
net_h: [[3.75201561 4.50001032 4.66701092 4.58101393]]
out_h: [[0.97706784 0.98901317 0.99068722 0.98985938]]
```

Gambar 4.13 Hasil Forward dan Backward Data Training

```
net_o: [[1.87895592]]
out_o: [[0.86749116]]
y_input: 0.8677
f(x): [[0.86749116]]
bobot input: [[0.20000142 0.30000094 0.40000099 0.50000126]
 [0.30000173 0.40000115 0.50000121 0.60000154]
 [0.40000285 0.50000188 0.60000199 0.70000254]
 [0.50000257 0.6000017  0.70000179 0.80000229]
 [0.60000105 0.7000007  0.80000074 0.90000094]
 [0.70000228 0.80000151 0.90000159 0.30000203]
 [0.80000146 0.90000097 0.30000102 0.5000013 ]]
bobot output: [[0.30024344]
 [0.40024664]
 [0.50024712]
 [0.60024696]]
Backward Pass karena y_input = [[0.86749116]] < 0.8677.
Total Error: [[2.18080589e-08]]
Update Bobot input: [[0.20000148 0.30000098 0.40000104 0.50000132]
 [0.30000181 0.4000012  0.50000127 0.60000162]
 [0.40000298 0.50000197 0.60000208 0.70000266]
 [0.50000267 0.60000176 0.70000186 0.80000238]
 [0.6000011  0.70000073 0.80000077 0.90000098]
 [0.70000238 0.80000157 0.90000167 0.30000213]
 [0.80000153 0.90000101 0.30000107 0.50000136]]
Updated Bobot output: [[0.30025517]
 [0.40025852]
 [0.50025901]
 [0.60025884]]
Forward-backward pass 3
net_h: [[3.75201632 4.50001078 4.66701141 4.58101456]]
out_h: [[0.97706785 0.98901317 0.99068722 0.98985939]]
net_o: [[1.87900268]]
out_o: [[0.86749653]]
y_input: 0.8677
f(x): [[0.86749653]]
bobot input: [[0.20000148 0.30000098 0.40000104 0.50000132]
 [0.30000181 0.4000012  0.50000127 0.60000162]
 [0.40000298 0.50000197 0.60000208 0.70000266]
 [0.50000267 0.60000176 0.70000186 0.80000238]
 [0.6000011  0.70000073 0.80000077 0.90000098]
 [0.70000238 0.80000157 0.90000167 0.30000213]
 [0.80000153 0.90000101 0.30000107 0.50000136]]
bobot output: [[0.30025517]
 [0.40025852]
 [0.50025901]
 [0.60025884]]
```

Gambar 4.14 Hasil Forward dan Backward Data Training

```
Backward Pass karena y_input = [[0.86749653]] < 0.8677.
Total Error: [[2.07000851e-08]]
Update Bobot input: [[0.20000155 0.30000102 0.40000108 0.50000138]
 [0.30000189 0.40000125 0.50000132 0.60000169]
 [0.40000311 0.50000205 0.60000217 0.70000277]
 [0.50000276 0.60000182 0.70000193 0.80000247]
 [0.60000115 0.70000076 0.80000081 0.90000103]
 [0.70000249 0.80000164 0.90000174 0.30000222]
 [0.80000159 0.90000105 0.30000111 0.50000142]]
Updated Bobot output: [[0.3002666 ]
 [0.40027008]
 [0.5002706 ]
 [0.60027041]]
Skip, Foto Selanjutnya

=========================================================================================================
Data Latih ke-7
Forward-backward pass 1
net_h: [[3.70101685 4.43001112 4.61301177 4.48901504]]
out_h: [[0.97589691 0.98822592 0.99017559 0.98889305]]
net_o: [[1.87754548]]
out_o: [[0.86732894]]
y_input: 0.8677
f(x): [[0.86732894]]
bobot input: [[0.20000155 0.30000102 0.40000108 0.50000138]
 [0.30000189 0.40000125 0.50000132 0.60000169]
 [0.40000311 0.50000205 0.60000217 0.70000277]
 [0.50000276 0.60000182 0.70000193 0.80000247]
 [0.60000115 0.70000076 0.80000081 0.90000103]
 [0.70000249 0.80000164 0.90000174 0.30000222]
 [0.80000159 0.90000105 0.30000111 0.50000142]]
bobot output: [[0.3002666 ]
 [0.40027008]
 [0.5002706 ]
 [0.60027041]]
Backward Pass karena y_input = [[0.86732894]] < 0.8677.
Total Error: [[6.88423207e-08]]
Update Bobot input: [[0.20000165 0.30000109 0.40000115 0.50000148]
 [0.30000204 0.40000135 0.50000142 0.60000183]
 [0.40000332 0.50000219 0.60000232 0.70000298]
 [0.500003   0.60000198 0.70000209 0.80000269]
 [0.60000123 0.70000081 0.80000086 0.9000011 ]
 [0.70000269 0.80000177 0.90000188 0.30000241]
 [0.80000171 0.90000113 0.30000119 0.50000153]]
Updated Bobot output: [[0.30028743]
 [0.40029118]
```

Gambar 4.15 Hasil Forward dan Backward Data Training

```
 [0.50029174]
 [0.60029153]]
Forward-backward pass 2
net_h: [[3.70101815 4.43001198 4.61301266 4.48901625]]
out_h: [[0.97589694 0.98822593 0.99017559 0.98889306]]
net_o: [[1.8776285]]
out_o: [[0.86733849]]
y_input: 0.8677
f(x): [[0.86733849]]
bobot input: [[0.20000165 0.30000109 0.40000115 0.50000148]
 [0.30000204 0.40000135 0.50000142 0.60000183]
 [0.40000332 0.50000219 0.60000232 0.70000298]
 [0.500003   0.60000198 0.70000209 0.80000269]
 [0.60000123 0.70000081 0.80000086 0.9000011 ]
 [0.70000269 0.80000177 0.90000188 0.30000241]
 [0.80000171 0.90000113 0.30000119 0.50000153]]
bobot output: [[0.30028743]
 [0.40029118]
 [0.50029174]
 [0.60029153]]
Backward Pass karena y_input = [[0.86733849]] < 0.8677.
Total Error: [[6.53435157e-08]]
Update Bobot input: [[0.20000175 0.30000115 0.40000122 0.50000157]
 [0.30000219 0.40000144 0.50000153 0.60000196]
 [0.40000354 0.50000233 0.60000247 0.70000317]
 [0.50000323 0.60000213 0.70000225 0.8000029 ]
 [0.6000013  0.70000086 0.80000091 0.90000117]
 [0.70000288 0.8000019  0.90000201 0.30000259]
 [0.80000182 0.9000012  0.30000127 0.50000164]]
Updated Bobot output: [[0.30030773]
 [0.40031173]
 [0.50031233]
 [0.60031209]]
Forward-backward pass 3
net_h: [[3.70101941 4.43001281 4.61301353 4.48901743]]
out_h: [[0.97589697 0.98822594 0.9901756  0.98889308]]
net_o: [[1.87770937]]
out_o: [[0.8673478]]
y_input: 0.8677
f(x): [[0.8673478]]
bobot input: [[0.20000175 0.30000115 0.40000122 0.50000157]
 [0.30000219 0.40000144 0.50000153 0.60000196]
 [0.40000354 0.50000233 0.60000247 0.70000317]
 [0.50000323 0.60000213 0.70000225 0.8000029 ]
 [0.6000013  0.70000086 0.80000091 0.90000117]
 [0.70000288 0.8000019  0.90000201 0.30000259]
```

Gambar 4.16 Hasil Forward dan Backward Data Training

```
 [0.80000182 0.9000012  0.30000127 0.50000164]]
bobot output: [[0.30030773]
 [0.40031173]
 [0.50031233]
 [0.60031209]]
Backward Pass karena y_input = [[0.8673478]] < 0.8677.
Total Error: [[6.20229291e-08]]
Update Bobot input: [[0.20000184 0.30000122 0.40000129 0.50000166]
 [0.30000233 0.40000154 0.50000162 0.6000021 ]
 [0.40000374 0.50000247 0.60000261 0.70000337]
 [0.50000345 0.60000228 0.7000024  0.80000311]
 [0.60000138 0.70000091 0.80000096 0.90000124]
 [0.70000307 0.80000203 0.90000214 0.30000277]
 [0.80000194 0.90000128 0.30000135 0.50000174]]
Updated Bobot output: [[0.3003275 ]
 [0.40033175]
 [0.50033239]
 [0.60033213]]
Skip, Foto Selanjutnya

==========================================================================================================
Data Latih ke-8
Forward-backward pass 1
net_h: [[3.88802188 4.65701444 4.82401525 4.77601969]]
out_h: [[0.97992542 0.99059454 0.99202958 0.99164098]]
net_o: [[1.88252347]]
out_o: [[0.86790071]]
y_input: 0.8677
f(x): [[0.86790071]]
bobot input: [[0.20000184 0.30000122 0.40000129 0.50000166]
 [0.30000233 0.40000154 0.50000162 0.6000021 ]
 [0.40000374 0.50000247 0.60000261 0.70000337]
 [0.50000345 0.60000228 0.7000024  0.80000311]
 [0.60000138 0.70000091 0.80000096 0.90000124]
 [0.70000307 0.80000203 0.90000214 0.30000277]
 [0.80000194 0.90000128 0.30000135 0.50000174]]
bobot output: [[0.3003275 ]
 [0.40033175]
 [0.50033239]
 [0.60033213]]
Forward Pass ke foto selanjutnya.
```

Gambar 4.17 Hasil Forward dan Backward Data Training

```
==========================================================================================================
Data Latih ke-9
Forward-backward pass 1
net_h: [[3.8360214  4.59601412 4.81001491 4.71701926]]
out_h: [[0.97887654 0.99000885 0.99191811 0.99113746]]
net_o: [[1.88161594]]
out_o: [[0.86779663]]
y_input: 0.8677
f(x): [[0.86779663]]
bobot input: [[0.20000184 0.30000122 0.40000129 0.50000166]
 [0.30000233 0.40000154 0.50000162 0.6000021 ]
 [0.40000374 0.50000247 0.60000261 0.70000337]
 [0.50000345 0.60000228 0.7000024  0.80000311]
 [0.60000138 0.70000091 0.80000096 0.90000124]
 [0.70000307 0.80000203 0.90000214 0.30000277]
 [0.80000194 0.90000128 0.30000135 0.50000174]]
bobot output: [[0.3003275 ]
 [0.40033175]
 [0.50033239]
 [0.60033213]]
Forward Pass ke foto selanjutnya.

==========================================================================================================
Data Latih ke-10
Forward-backward pass 1
net_h: [[4.26002386 5.09901574 5.23801662 5.31602147]]
out_h: [[0.98607469 0.99393427 0.99471728 0.99511175]]
net_o: [[1.88913563]]
out_o: [[0.86865694]]
y_input: 0.8677
f(x): [[0.86865694]]
bobot input: [[0.20000184 0.30000122 0.40000129 0.50000166]
 [0.30000233 0.40000154 0.50000162 0.6000021 ]
 [0.40000374 0.50000247 0.60000261 0.70000337]
 [0.50000345 0.60000228 0.7000024  0.80000311]
 [0.60000138 0.70000091 0.80000096 0.90000124]
 [0.70000307 0.80000203 0.90000214 0.30000277]
 [0.80000194 0.90000128 0.30000135 0.50000174]]
bobot output: [[0.3003275 ]
 [0.40033175]
 [0.50033239]
 [0.60033213]]
Forward Pass ke foto selanjutnya.
```

Gambar 4.18 Hasil Forward dan Backward Data Training

```
==========================================================================================================
Data Latih ke-11
Forward-backward pass 1
net_h: [[3.93802191 4.72201445 4.87601526 4.84001971]]
out_h: [[0.98088575 0.99118123 0.99243039 0.99215513]]
net_o: [[1.88355596]]
out_o: [[0.86801904]]
y_input: 0.8677
f(x): [[0.86801904]]
bobot input: [[0.20000184 0.30000122 0.40000129 0.50000166]
 [0.30000233 0.40000154 0.50000162 0.6000021 ]
 [0.40000374 0.50000247 0.60000261 0.70000337]
 [0.50000345 0.60000228 0.7000024  0.80000311]
 [0.60000138 0.70000091 0.80000096 0.90000124]
 [0.70000307 0.80000203 0.90000214 0.30000277]
 [0.80000194 0.90000128 0.30000135 0.50000174]]
bobot output: [[0.3003275 ]
 [0.40033175]
 [0.50033239]
 [0.60033213]]
Forward Pass ke foto selanjutnya.

==========================================================================================================
Data Latih ke-12
Forward-backward pass 1
net_h: [[4.00602234 4.80001474 4.99901556 4.9540201 ]]
out_h: [[0.98211985 0.99183755 0.9933006  0.99299443]]
net_o: [[1.88512859]]
out_o: [[0.8681991]]
y_input: 0.8677
f(x): [[0.8681991]]
bobot input: [[0.20000184 0.30000122 0.40000129 0.50000166]
 [0.30000233 0.40000154 0.50000162 0.6000021 ]
 [0.40000374 0.50000247 0.60000261 0.70000337]
 [0.50000345 0.60000228 0.7000024  0.80000311]
 [0.60000138 0.70000091 0.80000096 0.90000124]
 [0.70000307 0.80000203 0.90000214 0.30000277]
 [0.80000194 0.90000128 0.30000135 0.50000174]]
bobot output: [[0.3003275 ]
 [0.40033175]
 [0.50033239]
 [0.60033213]]
Forward Pass ke foto selanjutnya.
```

Gambar 4.19 Hasil Forward dan Backward Data Training

```
==========================================================================================================
Data Latih ke-13
Forward-backward pass 1
net_h: [[3.90502146 4.67401416 4.77101495 4.78201931]]
out_h: [[0.98025711 0.99075161 0.99159939 0.99169056]]
net_o: [[1.8825005]]
out_o: [[0.86789807]]
y_input: 0.8677
f(x): [[0.86789807]]
bobot input: [[0.20000184 0.30000122 0.40000129 0.50000166]
 [0.30000233 0.40000154 0.50000162 0.6000021 ]
 [0.40000374 0.50000247 0.60000261 0.70000337]
 [0.50000345 0.60000228 0.7000024  0.80000311]
 [0.60000138 0.70000091 0.80000096 0.90000124]
 [0.70000307 0.80000203 0.90000214 0.30000277]
 [0.80000194 0.90000128 0.30000135 0.50000174]]
bobot output: [[0.3003275 ]
 [0.40033175]
 [0.50033239]
 [0.60033213]]
Forward Pass ke foto selanjutnya.

==========================================================================================================
Data Latih ke-14
Forward-backward pass 1
net_h: [[4.21302349 5.0290155  5.14501637 5.07402114]]
out_h: [[0.98541434 0.99349731 0.99420539 0.9937817 ]]
net_o: [[1.88770779]]
out_o: [[0.86849395]]
y_input: 0.8677
f(x): [[0.86849395]]
bobot input: [[0.20000184 0.30000122 0.40000129 0.50000166]
 [0.30000233 0.40000154 0.50000162 0.6000021 ]
 [0.40000374 0.50000247 0.60000261 0.70000337]
 [0.50000345 0.60000228 0.7000024  0.80000311]
 [0.60000138 0.70000091 0.80000096 0.90000124]
 [0.70000307 0.80000203 0.90000214 0.30000277]
 [0.80000194 0.90000128 0.30000135 0.50000174]]
bobot output: [[0.3003275 ]
 [0.40033175]
 [0.50033239]
 [0.60033213]]
Forward Pass ke foto selanjutnya.
```

Gambar 4.20 Hasil Forward dan Backward Data Training

```
==========================================================================================================
Data Latih ke-15
Forward-backward pass 1
net_h: [[3.71402063 4.44301361 4.55601437 4.49801856]]
out_h: [[0.9762009  0.98837626 0.98960534 0.98899151]]
net_o: [[1.87771336]]
out_o: [[0.86734826]]
y_input: 0.8677
f(x): [[0.86734826]]
bobot input: [[0.20000184 0.30000122 0.40000129 0.50000166]
 [0.30000233 0.40000154 0.50000162 0.6000021 ]
 [0.40000374 0.50000247 0.60000261 0.70000337]
 [0.50000345 0.60000228 0.7000024  0.80000311]
 [0.60000138 0.70000091 0.80000096 0.90000124]
 [0.70000307 0.80000203 0.90000214 0.30000277]
 [0.80000194 0.90000128 0.30000135 0.50000174]]
bobot output: [[0.3003275 ]
 [0.40033175]
 [0.50033239]
 [0.60033213]]
Backward Pass karena y_input = [[0.86734826]] < 0.8677.
Total Error: [[6.18614079e-08]]
Update Bobot input: [[0.20000195 0.30000129 0.40000136 0.50000176]
 [0.30000245 0.40000161 0.50000171 0.6000022 ]
 [0.40000397 0.50000262 0.60000278 0.70000358]
 [0.50000366 0.60000241 0.70000256 0.8000033 ]
 [0.60000145 0.70000096 0.80000102 0.90000131]
 [0.70000325 0.80000214 0.90000227 0.30000293]
 [0.80000206 0.90000136 0.30000144 0.50000186]]
Updated Bobot output: [[0.30034725]
 [0.40035175]
 [0.50035242]
 [0.60035214]]
Forward-backward pass 2
net_h: [[3.71402184 4.44301441 4.55601527 4.4980197 ]]
out_h: [[0.97620093 0.98837627 0.98960535 0.98899152]]
net_o: [[1.87779204]]
out_o: [[0.86735731]]
y_input: 0.8677
f(x): [[0.86735731]]
bobot input: [[0.20000195 0.30000129 0.40000136 0.50000176]
 [0.30000245 0.40000161 0.50000171 0.6000022 ]
 [0.40000397 0.50000262 0.60000278 0.70000358]
 [0.50000366 0.60000241 0.70000256 0.8000033 ]
 [0.60000145 0.70000096 0.80000102 0.90000131]
 [0.70000325 0.80000214 0.90000227 0.30000293]
```

Gambar 4.21 Hasil Forward dan Backward Data Training

```
 [0.80000206 0.90000136 0.30000144 0.50000186]]
bobot output: [[0.30034725]
 [0.40035175]
 [0.50035242]
 [0.60035214]]
Backward Pass karena y_input = [[0.86735731]] < 0.8677.
Total Error: [[5.87181917e-08]]
Update Bobot input: [[0.20000205 0.30000135 0.40000144 0.50000185]
 [0.30000256 0.40000169 0.50000179 0.60000231]
 [0.4000042  0.50000277 0.60000294 0.70000379]
 [0.50000386 0.60000254 0.7000027  0.80000349]
 [0.60000153 0.70000101 0.80000107 0.90000138]
 [0.70000342 0.80000226 0.9000024  0.30000309]
 [0.80000218 0.90000144 0.30000153 0.50000197]]
Updated Bobot output: [[0.3003665 ]
 [0.40037124]
 [0.50037192]
 [0.60037164]]
Forward-backward pass 3
net_h: [[3.71402302 4.44301519 4.55601614 4.4980208 ]]
out_h: [[0.97620096 0.98837627 0.98960536 0.98899153]]
net_o: [[1.8778687]]
out_o: [[0.86736613]]
y_input: 0.8677
f(x): [[0.86736613]]
bobot input: [[0.20000205 0.30000135 0.40000144 0.50000185]
 [0.30000256 0.40000169 0.50000179 0.60000231]
 [0.4000042  0.50000277 0.60000294 0.70000379]
 [0.50000386 0.60000254 0.7000027  0.80000349]
 [0.60000153 0.70000101 0.80000107 0.90000138]
 [0.70000342 0.80000226 0.9000024  0.30000309]
 [0.80000218 0.90000144 0.30000153 0.50000197]]
bobot output: [[0.3003665 ]
 [0.40037124]
 [0.50037192]
 [0.60037164]]
Backward Pass karena y_input = [[0.86736613]] < 0.8677.
Total Error: [[5.57350223e-08]]
Update Bobot input: [[0.20000215 0.30000142 0.40000151 0.50000195]
 [0.30000266 0.40000176 0.50000187 0.60000241]
 [0.40000441 0.50000291 0.6000031  0.700004  ]
 [0.50000405 0.60000267 0.70000285 0.80000367]
 [0.6000016  0.70000105 0.80000112 0.90000144]
 [0.70000359 0.80000237 0.90000252 0.30000325]
 [0.8000023  0.90000152 0.30000162 0.50000208]]
```

Gambar 4.22 Hasil Forward dan Backward Data Training

```
Updated Bobot output: [[0.30038524]
 [0.40039022]
 [0.50039093]
 [0.60039063]]
Skip, Foto Selanjutnya

==========================================================================================================
Data Latih ke-16
Forward-backward pass 1
net_h: [[3.55102333 4.25901539 4.41401639 4.37502112]]
out_h: [[0.97210519 0.98606083 0.98803836 0.98756861]]
net_o: [[1.87414754]]
out_o: [[0.86693745]]
y_input: 0.8677
f(x): [[0.86693745]]
bobot input: [[0.20000215 0.30000142 0.40000151 0.50000195]
 [0.30000266 0.40000176 0.50000187 0.60000241]
 [0.40000441 0.50000291 0.6000031  0.700004  ]
 [0.50000405 0.60000267 0.70000285 0.80000367]
 [0.6000016  0.70000105 0.80000112 0.90000144]
 [0.70000359 0.80000237 0.90000252 0.30000325]
 [0.8000023  0.90000152 0.30000162 0.50000208]]
bobot output: [[0.30038524]
 [0.40039022]
 [0.50039093]
 [0.60039063]]
Backward Pass karena y_input = [[0.86693745]] < 0.8677.
Total Error: [[2.90738798e-07]]
Update Bobot input: [[0.20000241 0.30000159 0.4000017  0.50000218]
 [0.30000301 0.40000199 0.5000212 0.60000272]
 [0.40000498 0.50000329 0.60000352 0.70000451]
 [0.50000452 0.60000299 0.70000319 0.8000041 ]
 [0.60000179 0.70000118 0.80000126 0.90000162]
 [0.70000401 0.80000265 0.90000283 0.30000363]
 [0.80000258 0.90000171 0.30000182 0.50000234]]
Updated Bobot output: [[0.300428  ]
 [0.40043359]
 [0.50043439]
 [0.60043407]]
Forward-backward pass 2
net_h: [[3.55102619 4.25901732 4.41401847 4.37502371]]
out_h: [[0.97210527 0.98606086 0.98803838 0.98756864]]
net_o: [[1.87431776]]
out_o: [[0.86695709]]
y_input: 0.8677
f(x): [[0.86695709]]
bobot input: [[0.20000241 0.30000159 0.4000017  0.50000218]
```

Gambar 4.23 Hasil Forward dan Backward Data Training

```
 [0.30000301 0.40000199 0.50000212 0.60000272]
 [0.40000498 0.50000329 0.60000352 0.70000451]
 [0.50000452 0.60000299 0.70000319 0.8000041 ]
 [0.60000179 0.70000118 0.80000126 0.90000162]
 [0.70000401 0.80000265 0.90000283 0.30000363]
 [0.80000258 0.90000171 0.30000182 0.50000234]]
bobot output: [[0.300428  ]
 [0.40043359]
 [0.50043439]
 [0.60043407]]
Backward Pass karena y_input = [[0.86695709]] < 0.8677.
Total Error: [[2.75958612e-07]]
Update Bobot input: [[0.20000266 0.30000176 0.40000188 0.5000024 ]
 [0.30000335 0.40000222 0.50000237 0.60000303]
 [0.40000553 0.50000367 0.60000392 0.70000501]
 [0.50000498 0.6000033  0.70000352 0.80000451]
 [0.60000197 0.70000131 0.80000139 0.90000178]
 [0.70000442 0.80000293 0.90000313 0.30000401]
 [0.80000286 0.90000189 0.30000202 0.50000259]]
Updated Bobot output: [[0.30046965]
 [0.40047584]
 [0.50047672]
 [0.60047638]]
Forward-backward pass 3
net_h: [[3.55102898 4.25901921 4.41402049 4.37502623]]
out_h: [[0.97210534 0.98606089 0.98803841 0.98756867]]
net_o: [[1.87448359]]
out_o: [[0.86697621]]
y_input: 0.8677
f(x): [[0.86697621]]
bobot input: [[0.20000266 0.30000176 0.40000188 0.5000024 ]
 [0.30000335 0.40000222 0.50000237 0.60000303]
 [0.40000553 0.50000367 0.60000392 0.70000501]
 [0.50000498 0.6000033  0.70000352 0.80000451]
 [0.60000197 0.70000131 0.80000139 0.90000178]
 [0.70000442 0.80000293 0.90000313 0.30000401]
 [0.80000286 0.90000189 0.30000202 0.50000259]]
bobot output: [[0.30046965]
 [0.40047584]
 [0.50047672]
 [0.60047638]]
Backward Pass karena y_input = [[0.86697621]] < 0.8677.
Total Error: [[2.61933236e-07]]
Update Bobot input: [[0.2000029  0.30000192 0.40000205 0.50000262]
 [0.30000368 0.40000244 0.50000261 0.60000333]
 [0.40000607 0.50000403 0.60000431 0.70000549]
```

Gambar 4.24 Hasil Forward dan Backward Data Training

```
 [0.50000543 0.60000361 0.70000385 0.80000492]
 [0.60000215 0.70000143 0.80000152 0.90000195]
 [0.70000483 0.8000032  0.90000342 0.30000437]
 [0.80000313 0.90000208 0.30000222 0.50000283]]
Updated Bobot output: [[0.30051022]
 [0.40051699]
 [0.50051796]
 [0.6005176 ]]
Skip, Foto Selanjutnya

==========================================================================================================
Data Latih ke-17
Forward-backward pass 1
net_h: [[4.52503684 5.38802446 5.20102612 5.38103335]]
out_h: [[0.98928181 0.9954498  0.9945193  0.99541803]]
net_o: [[1.89152466]]
out_o: [[0.86892927]]
y_input: 0.8677
f(x): [[0.86892927]]
bobot input: [[0.2000029  0.30000192 0.40000205 0.50000262]
 [0.30000368 0.40000244 0.50000261 0.60000333]
 [0.40000607 0.50000403 0.60000431 0.70000549]
 [0.50000543 0.60000361 0.70000385 0.80000492]
 [0.60000215 0.70000143 0.80000152 0.90000195]
 [0.70000483 0.8000032  0.90000342 0.30000437]
 [0.80000313 0.90000208 0.30000222 0.50000283]]
bobot output: [[0.30051022]
 [0.40051699]
 [0.50051796]
 [0.6005176 ]]
Forward Pass ke foto selanjutnya.

==========================================================================================================
Data Latih ke-18
Forward-backward pass 1
net_h: [[3.56803214 4.27702134 4.46102278 4.37002909]]
out_h: [[0.97256273 0.98630617 0.98858135 0.98750717]]
net_o: [[1.87511557]]
out_o: [[0.86704908]]
y_input: 0.8677
f(x): [[0.86704908]]
bobot input: [[0.2000029  0.30000192 0.40000205 0.50000262]
 [0.30000368 0.40000244 0.50000261 0.60000333]
 [0.40000607 0.50000403 0.60000431 0.70000549]
 [0.50000543 0.60000361 0.70000385 0.80000492]
 [0.60000215 0.70000143 0.80000152 0.90000195]
```

Gambar 4.25 Hasil Forward dan Backward Data Training

```
 [0.70000483 0.8000032  0.90000342 0.30000437]
 [0.80000313 0.90000208 0.30000222 0.50000283]]
bobot output: [[0.30051022]
 [0.40051699]
 [0.50051796]
 [0.6005176 ]]
Backward Pass karena y_input = [[0.86704908]] < 0.8677.
Total Error: [[2.11846892e-07]]
Update Bobot input: [[0.2000031  0.30000206 0.4000022  0.50000281]
 [0.30000396 0.40000263 0.5000028  0.60000358]
 [0.40000657 0.50000437 0.60000466 0.70000596]
 [0.50000583 0.60000388 0.70000413 0.80000529]
 [0.6000023  0.70000153 0.80000163 0.90000208]
 [0.7000052  0.80000346 0.90000368 0.30000472]
 [0.80000335 0.90000223 0.30000238 0.50000304]]
Updated Bobot output: [[0.30054671]
 [0.40055399]
 [0.50055504]
 [0.60055464]]
Forward-backward pass 2
net_h: [[3.56803461 4.27702301 4.46102453 4.37003138]]
out_h: [[0.97256279 0.98630619 0.98858137 0.9875072 ]]
net_o: [[1.87526086]]
out_o: [[0.86706583]]
y_input: 0.8677
f(x): [[0.86706583]]
bobot input: [[0.2000031  0.30000206 0.4000022  0.50000281]
 [0.30000396 0.40000263 0.5000028  0.60000358]
 [0.40000657 0.50000437 0.60000466 0.70000596]
 [0.50000583 0.60000388 0.70000413 0.80000529]
 [0.6000023  0.70000153 0.80000163 0.90000208]
 [0.7000052  0.80000346 0.90000368 0.30000472]
 [0.80000335 0.90000223 0.30000238 0.50000304]]
bobot output: [[0.30054671]
 [0.40055399]
 [0.50055504]
 [0.60055464]]
Backward Pass karena y_input = [[0.86706583]] < 0.8677.
Total Error: [[2.01085925e-07]]
Update Bobot input: [[0.20000329 0.30000219 0.40000233 0.50000299]
 [0.30000423 0.40000281 0.50000299 0.60000383]
 [0.40000706 0.5000047  0.600005   0.70000641]
 [0.50000622 0.60000414 0.70000441 0.80000565]
 [0.60000245 0.70000163 0.80000173 0.90000222]
 [0.70000557 0.8000037  0.90000394 0.30000505]
 [0.80000357 0.90000238 0.30000253 0.50000324]]
```

Gambar 4.26 Hasil Forward dan Backward Data Training

```
Updated Bobot output: [[0.30058225]
 [0.40059004]
 [0.50059118]
 [0.60059074]]
Forward-backward pass 3
net_h: [[3.56803703 4.27702463 4.46102623 4.37003361]]
out_h: [[0.97256286 0.98630621 0.98858139 0.98750723]]
net_o: [[1.87540239]]
out_o: [[0.86708214]]
y_input: 0.8677
f(x): [[0.86708214]]
bobot input: [[0.20000329 0.30000219 0.40000233 0.50000299]
 [0.30000423 0.40000281 0.50000299 0.60000383]
 [0.40000706 0.5000047  0.600005   0.70000641]
 [0.50000622 0.60000414 0.70000441 0.80000565]
 [0.60000245 0.70000163 0.80000173 0.90000222]
 [0.70000557 0.8000037  0.90000394 0.30000505]
 [0.80000357 0.90000238 0.30000253 0.50000324]]
bobot output: [[0.30058225]
 [0.40059004]
 [0.50059118]
 [0.60059074]]
Backward Pass karena y_input = [[0.86708214]] < 0.8677.
Total Error: [[1.90873707e-07]]
Update Bobot input: [[0.20000349 0.30000232 0.40000247 0.50000316]
 [0.30000449 0.40000299 0.50000318 0.60000407]
 [0.40000754 0.50000502 0.60000534 0.70000685]
 [0.5000066  0.6000044  0.70000467 0.800006  ]
 [0.60000259 0.70000172 0.80000183 0.90000235]
 [0.70000592 0.80000394 0.90000419 0.30000538]
 [0.80000379 0.90000252 0.30000268 0.50000344]]
Updated Bobot output: [[0.30061688]
 [0.40062516]
 [0.50062637]
 [0.60062589]]
Skip, Foto Selanjutnya

==========================================================================================================
Data Latih ke-19
Forward-backward pass 1
net_h: [[3.96104293 4.74202859 4.9070304  4.85903901]]
out_h: [[0.98131263 0.99135446 0.99265986 0.99230179]]
net_o: [[1.88511453]]
out_o: [[0.86819749]]
y_input: 0.8677
f(x): [[0.86819749]]
```

Gambar 4.27 Hasil Forward dan Backward Data Training

```
bobot input: [[0.20000349 0.30000232 0.40000247 0.50000316]
 [0.30000449 0.40000299 0.50000318 0.60000407]
 [0.40000754 0.50000502 0.60000534 0.70000685]
 [0.5000066  0.6000044  0.70000467 0.800006  ]
 [0.60000259 0.70000172 0.80000183 0.90000235]
 [0.70000592 0.80000394 0.90000419 0.30000538]
 [0.80000379 0.90000252 0.30000268 0.50000344]]
bobot output: [[0.30061688]
 [0.40062516]
 [0.50062637]
 [0.60062589]]
Forward Pass ke foto selanjutnya.

==========================================================================================================
Data Latih ke-20
Forward-backward pass 1
net_h: [[3.66703943 4.38602625 4.49602792 4.43403582]]
out_h: [[0.97508463 0.98770299 0.98896981 0.98827266]]
net_o: [[1.87751209]]
out_o: [[0.8673251]]
y_input: 0.8677
f(x): [[0.8673251]]
bobot input: [[0.20000349 0.30000232 0.40000247 0.50000316]
 [0.30000449 0.40000299 0.50000318 0.60000407]
 [0.40000754 0.50000502 0.60000534 0.70000685]
 [0.5000066  0.6000044  0.70000467 0.800006  ]
 [0.60000259 0.70000172 0.80000183 0.90000235]
 [0.70000592 0.80000394 0.90000419 0.30000538]
 [0.80000379 0.90000252 0.30000268 0.50000344]]
bobot output: [[0.30061688]
 [0.40062516]
 [0.50062637]
 [0.60062589]]
Backward Pass karena y_input = [[0.8673251]] < 0.8677.
Total Error: [[7.02754934e-08]]
Update Bobot input: [[0.20000359 0.30000239 0.40000255 0.50000326]
 [0.30000463 0.40000309 0.50000329 0.60000421]
 [0.40000779 0.50000519 0.60000553 0.7000071 ]
 [0.50000681 0.60000453 0.70000483 0.80000619]
 [0.60000268 0.70000178 0.8000019  0.90000244]
 [0.70000612 0.80000407 0.90000434 0.30000557]
 [0.80000392 0.90000261 0.30000278 0.50000357]]
Updated Bobot output: [[0.30063792]
 [0.40064646]
 [0.50064771]
 [0.60064721]]
```

Gambar 4.28 Hasil Forward dan Backward Data Training

```
Forward-backward pass 2
net_h: [[3.66704073 4.38602712 4.49602889 4.43403706]]
out_h: [[0.97508466 0.987703   0.98896982 0.98827268]]
net_o: [[1.87759584]]
out_o: [[0.86733474]]
y_input: 0.8677
f(x): [[0.86733474]]
bobot input: [[0.20000359 0.30000239 0.40000255 0.50000326]
 [0.30000463 0.40000309 0.50000329 0.60000421]
 [0.40000779 0.50000519 0.60000553 0.7000071 ]
 [0.50000681 0.60000453 0.70000483 0.80000619]
 [0.60000268 0.70000178 0.8000019  0.90000244]
 [0.70000612 0.80000407 0.90000434 0.30000557]
 [0.80000392 0.90000261 0.30000278 0.50000357]]
bobot output: [[0.30063792]
 [0.40064646]
 [0.50064771]
 [0.60064721]]
Backward Pass karena y_input = [[0.86733474]] < 0.8677.
Total Error: [[6.67092454e-08]]
Update Bobot input: [[0.20000369 0.30000246 0.40000262 0.50000336]
 [0.30000478 0.40000318 0.50000339 0.60000435]
 [0.40000804 0.50000536 0.60000572 0.70000733]
 [0.50000701 0.60000466 0.70000498 0.80000639]
 [0.60000277 0.70000184 0.80000197 0.90000252]
 [0.70000631 0.8000042  0.90000448 0.30000575]
 [0.80000406 0.9000027  0.30000288 0.5000037 ]]
Updated Bobot output: [[0.30065841]
 [0.40066722]
 [0.50066849]
 [0.60066798]]
Forward-backward pass 3
net_h: [[3.667042   4.38602796 4.49602984 4.43403827]]
out_h: [[0.97508469 0.98770302 0.98896983 0.98827269]]
net_o: [[1.87767742]]
out_o: [[0.86734412]]
y_input: 0.8677
f(x): [[0.86734412]]
bobot input: [[0.20000369 0.30000246 0.40000262 0.50000336]
 [0.30000478 0.40000318 0.50000339 0.60000435]
 [0.40000804 0.50000536 0.60000572 0.70000733]
 [0.50000701 0.60000466 0.70000498 0.80000639]
 [0.60000277 0.70000184 0.80000197 0.90000252]
 [0.70000631 0.8000042  0.90000448 0.30000575]
 [0.80000406 0.9000027  0.30000288 0.5000037 ]]
```

Gambar 4.29 Hasil Forward dan Backward Data Training

```
bobot output: [[0.30065841]
 [0.40066722]
 [0.50066849]
 [0.60066798]]
Backward Pass karena y_input = [[0.86734412]] < 0.8677.
Total Error: [[6.33243808e-08]]
Update Bobot input: [[0.20000379 0.30000252 0.40000269 0.50000345]
 [0.30000492 0.40000327 0.5000035  0.60000448]
 [0.40000829 0.50000552 0.6000059  0.70000757]
 [0.5000072  0.60000479 0.70000512 0.80000657]
 [0.60000285 0.7000019  0.80000203 0.9000026 ]
 [0.7000065  0.80000433 0.90000462 0.30000593]
 [0.80000419 0.90000279 0.30000298 0.50000382]]
Updated Bobot output: [[0.30067837]
 [0.40068744]
 [0.50068874]
 [0.60068821]]
Skip, Foto Selanjutnya

==========================================================================================================
Data Latih ke-21
Forward-backward pass 1
net_h: [[3.522043   4.21802863 4.4380306  4.36203924]]
out_h: [[0.97130849 0.98548611 0.98831887 0.98740822]]
net_o: [[1.87488796]]
out_o: [[0.86702284]]
y_input: 0.8677
f(x): [[0.86702284]]
bobot input: [[0.20000379 0.30000252 0.40000269 0.50000345]
 [0.30000492 0.40000327 0.5000035  0.60000448]
 [0.40000829 0.50000552 0.6000059  0.70000757]
 [0.5000072  0.60000479 0.70000512 0.80000657]
 [0.60000285 0.7000019  0.80000203 0.9000026 ]
 [0.7000065  0.80000433 0.90000462 0.30000593]
 [0.80000419 0.90000279 0.30000298 0.50000382]]
bobot output: [[0.30067837]
 [0.40068744]
 [0.50068874]
 [0.60068821]]
Backward Pass karena y_input = [[0.86702284]] < 0.8677.
Total Error: [[2.29270589e-07]]
Update Bobot input: [[0.20000398 0.30000265 0.40000283 0.50000363]
 [0.30000518 0.40000345 0.50000368 0.60000472]
 [0.40000881 0.50000588 0.60000626 0.70000803]
 [0.50000772 0.60000515 0.70000548 0.80000704]
 [0.60000301 0.70000201 0.80000214 0.90000274]
```

Gambar 4.30 Hasil Forward dan Backward Data Training

```
 [0.70000689 0.80000459 0.90000489 0.30000628]
 [0.80000441 0.90000294 0.30000314 0.50000402]]
Updated Bobot output: [[0.30071629]
 [0.40072591]
 [0.50072732]
 [0.60072676]]
Forward-backward pass 2
net_h: [[3.5220457  4.21803048 4.43803246 4.36204164]]
out_h: [[0.97130857 0.98548613 0.98831889 0.98740825]]
net_o: [[1.87503895]]
out_o: [[0.86704025]]
y_input: 0.8677
f(x): [[0.86704025]]
bobot input: [[0.20000398 0.30000265 0.40000283 0.50000363]
 [0.30000518 0.40000345 0.50000368 0.60000472]
 [0.40000881 0.50000588 0.60000626 0.70000803]
 [0.50000772 0.60000515 0.70000548 0.80000704]
 [0.60000301 0.70000201 0.80000214 0.90000274]
 [0.70000689 0.80000459 0.90000489 0.30000628]
 [0.80000441 0.90000294 0.30000314 0.50000402]]
bobot output: [[0.30071629]
 [0.40072591]
 [0.50072732]
 [0.60072676]]
Backward Pass karena y_input = [[0.86704025]] < 0.8677.
Total Error: [[2.17634607e-07]]
Update Bobot input: [[0.20000417 0.30000279 0.40000296 0.5000038 ]
 [0.30000543 0.40000363 0.50000385 0.60000494]
 [0.40000932 0.50000623 0.60000661 0.70000849]
 [0.50000823 0.6000055  0.70000583 0.80000749]
 [0.60000316 0.70000211 0.80000224 0.90000288]
 [0.70000727 0.80000486 0.90000516 0.30000662]
 [0.80000463 0.90000309 0.30000329 0.50000421]]
Updated Bobot output: [[0.30075322]
 [0.40076339]
 [0.5007649 ]
 [0.60076431]]
Forward-backward pass 3
net_h: [[3.52204833 4.21803228 4.43803427 4.36204399]]
out_h: [[0.97130864 0.98548616 0.98831891 0.98740828]]
net_o: [[1.87518605]]
out_o: [[0.86705721]]
y_input: 0.8677
f(x): [[0.86705721]]
bobot input: [[0.20000417 0.30000279 0.40000296 0.5000038 ]
 [0.30000543 0.40000363 0.50000385 0.60000494]
```

Gambar 4.31 Hasil Forward dan Backward Data Training

```
 [0.40000932 0.50000623 0.60000661 0.70000849]
 [0.50000823 0.6000055  0.70000583 0.80000749]
 [0.60000316 0.70000211 0.80000224 0.90000288]
 [0.70000727 0.80000486 0.90000516 0.30000662]
 [0.80000463 0.90000309 0.30000329 0.50000421]]
bobot output: [[0.30075322]
 [0.40076339]
 [0.5007649 ]
 [0.60076431]]
Backward Pass karena y_input = [[0.86705721]] < 0.8677.
Total Error: [[2.06591576e-07]]
Update Bobot input: [[0.20000436 0.30000291 0.40000309 0.50000396]
 [0.30000568 0.4000038  0.50000402 0.60000516]
 [0.40000982 0.50000657 0.60000696 0.70000893]
 [0.50000873 0.60000584 0.70000618 0.80000793]
 [0.60000331 0.70000221 0.80000235 0.90000301]
 [0.70000764 0.80000511 0.90000541 0.30000695]
 [0.80000484 0.90000323 0.30000343 0.5000044 ]]
Updated Bobot output: [[0.30078921]
 [0.4007999 ]
 [0.50080151]
 [0.60080089]]
Skip, Foto Selanjutnya

==========================================================================================================
Data Latih ke-22
Forward-backward pass 1
net_h: [[3.86005464 4.63003654 4.8400387  4.78004968]]
out_h: [[0.97936781 0.99033983 0.99215528 0.99167432]]
net_o: [[1.88418304]]
out_o: [[0.86809086]]
y_input: 0.8677
f(x): [[0.86809086]]
bobot input: [[0.20000436 0.30000291 0.40000309 0.50000396]
 [0.30000568 0.4000038  0.50000402 0.60000516]
 [0.40000982 0.50000657 0.60000696 0.70000893]
 [0.50000873 0.60000584 0.70000618 0.80000793]
 [0.60000331 0.70000221 0.80000235 0.90000301]
 [0.70000764 0.80000511 0.90000541 0.30000695]
 [0.80000484 0.90000323 0.30000343 0.5000044 ]]
bobot output: [[0.30078921]
 [0.4007999 ]
 [0.50080151]
 [0.60080089]]
Forward Pass ke foto selanjutnya.
```

Gambar 4.32 Hasil Forward dan Backward Data Training

```
==========================================================================================================
Data Latih ke-23
Forward-backward pass 1
net_h: [[3.66105121 4.38603424 4.49503626 4.51704656]]
out_h: [[0.97493873 0.98770309 0.98895899 0.98919675]]
net_o: [[1.87870479]]
out_o: [[0.86746229]]
y_input: 0.8677
f(x): [[0.86746229]]
bobot input: [[0.20000436 0.30000291 0.40000309 0.50000396]
 [0.30000568 0.4000038  0.50000402 0.60000516]
 [0.40000982 0.50000657 0.60000696 0.70000893]
 [0.50000873 0.60000584 0.70000618 0.80000793]
 [0.60000331 0.70000221 0.80000235 0.90000301]
 [0.70000764 0.80000511 0.90000541 0.30000695]
 [0.80000484 0.90000323 0.30000343 0.5000044 ]]
bobot output: [[0.30078921]
 [0.4007999 ]
 [0.50080151]
 [0.60080089]]
Backward Pass karena y_input = [[0.86746229]] < 0.8677.
Total Error: [[2.82540762e-08]]
Update Bobot input: [[0.20000443 0.30000296 0.40000314 0.50000403]
 [0.30000577 0.40000386 0.50000409 0.60000524]
 [0.40000998 0.50000668 0.60000708 0.70000907]
 [0.50000887 0.60000593 0.70000628 0.80000806]
 [0.60000337 0.70000225 0.80000239 0.90000306]
 [0.70000776 0.80000519 0.9000055  0.30000705]
 [0.80000493 0.90000329 0.3000035  0.50000448]]
Updated Bobot output: [[0.30080253]
 [0.40081339]
 [0.50081503]
 [0.60081441]]
Forward-backward pass 2
net_h: [[3.66105204 4.38603479 4.49503688 4.51704729]]
out_h: [[0.97493876 0.9877031  0.988959   0.98919676]]
net_o: [[1.87875787]]
out_o: [[0.86746839]]
y_input: 0.8677
f(x): [[0.86746839]]
bobot input: [[0.20000443 0.30000296 0.40000314 0.50000403]
 [0.30000577 0.40000386 0.50000409 0.60000524]
 [0.40000998 0.50000668 0.60000708 0.70000907]
 [0.50000887 0.60000593 0.70000628 0.80000806]
 [0.60000337 0.70000225 0.80000239 0.90000306]
 [0.70000776 0.80000519 0.9000055  0.30000705]
```

Gambar 4.33 Hasil Forward dan Backward Data Training

```
 [0.80000493 0.90000329 0.3000035  0.50000448]]
bobot output: [[0.30080253]
 [0.40081339]
 [0.50081503]
 [0.60081441]]
Backward Pass karena y_input = [[0.86746839]] < 0.8677.
Total Error: [[2.68222007e-08]]
Update Bobot input: [[0.20000451 0.30000301 0.4000032  0.50000409]
 [0.30000586 0.40000391 0.50000415 0.60000532]
 [0.40001014 0.50000678 0.6000072  0.70000921]
 [0.50000901 0.60000602 0.70000638 0.80000817]
 [0.60000343 0.70000229 0.80000244 0.90000312]
 [0.70000787 0.80000526 0.90000558 0.30000715]
 [0.80000501 0.90000335 0.30000356 0.50000455]]
Updated Bobot output: [[0.30081551]
 [0.40082654]
 [0.5008282 ]
 [0.60082758]]
Forward-backward pass 3
net_h: [[3.66105285 4.38603533 4.49503749 4.517048  ]]
out_h: [[0.97493878 0.9877031  0.988959   0.98919677]]
net_o: [[1.87880958]]
out_o: [[0.86747433]]
y_input: 0.8677
f(x): [[0.86747433]]
bobot input: [[0.20000451 0.30000301 0.4000032  0.50000409]
 [0.30000586 0.40000391 0.50000415 0.60000532]
 [0.40001014 0.50000678 0.6000072  0.70000921]
 [0.50000901 0.60000602 0.70000638 0.80000817]
 [0.60000343 0.70000229 0.80000244 0.90000312]
 [0.70000787 0.80000526 0.90000558 0.30000715]
 [0.80000501 0.90000335 0.30000356 0.50000455]]
bobot output: [[0.30081551]
 [0.40082654]
 [0.5008282 ]
 [0.60082758]]
Backward Pass karena y_input = [[0.86747433]] < 0.8677.
Total Error: [[2.54629945e-08]]
Update Bobot input: [[0.20000458 0.30000306 0.40000325 0.50000415]
 [0.30000594 0.40000397 0.50000422 0.60000539]
 [0.4000103  0.50000688 0.60000731 0.70000935]
 [0.50000914 0.60000611 0.70000648 0.80000829]
 [0.60000349 0.70000233 0.80000248 0.90000317]
 [0.70000798 0.80000533 0.90000566 0.30000724]
 [0.8000051  0.9000034  0.30000362 0.50000463]]
```

Gambar 4.34 Hasil Forward dan Backward Data Training

```
Updated Bobot output: [[0.30082816]
 [0.40083936]
 [0.50084102]
 [0.60084041]]
Skip, Foto Selanjutnya

==========================================================================================================
Data Latih ke-24
Forward-backward pass 1
net_h: [[3.7790542  4.53603622 4.67003847 5.12404918]]
out_h: [[0.97766592 0.98939781 0.99071511 0.99408334]]
net_o: [[1.88417523]]
out_o: [[0.86808997]]
y_input: 0.8677
f(x): [[0.86808997]]
bobot input: [[0.20000458 0.30000306 0.40000325 0.50000415]
 [0.30000594 0.40000397 0.50000422 0.60000539]
 [0.4000103  0.50000688 0.60000731 0.70000935]
 [0.50000914 0.60000611 0.70000648 0.80000829]
 [0.60000349 0.70000233 0.80000248 0.90000317]
 [0.70000798 0.80000533 0.90000566 0.30000724]
 [0.8000051  0.9000034  0.30000362 0.50000463]]
bobot output: [[0.30082816]
 [0.40083936]
 [0.50084102]
 [0.60084041]]
Forward Pass ke foto selanjutnya.
```

Gambar 4.35 Hasil Forward dan Backward Data Training

```
==========================================================================================================
Data Latih ke-25
Forward-backward pass 1
net_h: [[3.84105672 4.60903791 4.83104026 4.76005148]]
out_h: [[0.97898041 0.99013685 0.99208493 0.99150757]]
net_o: [[1.88400534]]
out_o: [[0.86807051]]
y_input: 0.8677
f(x): [[0.86807051]]
bobot input: [[0.20000458 0.30000306 0.40000325 0.50000415]
 [0.30000594 0.40000397 0.50000422 0.60000539]
 [0.4000103  0.50000688 0.60000731 0.70000935]
 [0.50000914 0.60000611 0.70000648 0.80000829]
 [0.60000349 0.70000233 0.80000248 0.90000317]
 [0.70000798 0.80000533 0.90000566 0.30000724]
 [0.8000051  0.9000034  0.30000362 0.50000463]]
bobot output: [[0.30082816]
 [0.40083936]
 [0.50084102]
 [0.60084041]]
Forward Pass ke foto selanjutnya.
```

Gambar 4.36 Hasil Forward dan Backward Data Training

4.1.7 Hasil Forward Data Testing

```
Testing Data Results:
Data Testing Ke-1
net_h: [[4.0520851  4.85105701 4.97106042 4.92907723]]
out_h: [[0.98291103 0.99224057 0.99311198 0.99281877]]
net_o: [[1.88879278]]
out_o: [[0.86861782]]
y_input: 0.8677
f(x): [[0.86861782]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-1: Benar

Data Testing Ke-2
net_h: [[3.71208028 4.45005379 4.642057   4.60407286]]
out_h: [[0.97615578 0.98845686 0.99045415 0.99008825]]
net_o: [[1.88226633]]
out_o: [[0.86787123]]
y_input: 0.8677
f(x): [[0.86787123]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-2: Benar
```

Gambar 4.37 Hasil Forward Data Testing

```
Data Testing Ke-3
net_h: [[3.63207789 4.34405218 4.5380553  4.42807069]]
out_h: [[0.974221   0.98718261 0.98941897 0.98820332]]
net_o: [[1.87952028]]
out_o: [[0.86755601]]
y_input: 0.8677
f(x): [[0.86755601]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-3: Salah

Data Testing Ke-4
net_h: [[3.60607664 4.31605134 4.46605441 4.38806955]]
out_h: [[0.97355988 0.98682344 0.98863801 0.98772779]]
net_o: [[1.87849972]]
out_o: [[0.86743871]]
y_input: 0.8677
f(x): [[0.86743871]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-4: Salah
```

Gambar 4.38 Hasil Forward Data Testing

```
Data Testing Ke-5
net_h: [[3.62707702 4.3440516  4.49405468 4.4450699 ]]
out_h: [[0.97409511 0.9871826  0.98894827 0.98839986]]
net_o: [[1.87936459]]
out_o: [[0.86753812]]
y_input: 0.8677
f(x): [[0.86753812]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-5: Salah

Data Testing Ke-6
net_h: [[4.08008647 4.88505793 5.02506139 4.94507848]]
out_h: [[0.98337506 0.99249802 0.99347172 0.99293196]]
net_o: [[1.88928418]]
out_o: [[0.86867389]]
y_input: 0.8677
f(x): [[0.86867389]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-6: Benar
```

Gambar 4.39 Hasil Forward Data Testing

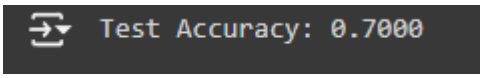
```
Data Testing Ke-7
net_h: [[3.8970829  4.66705554 4.82805885 4.74007523]]
out_h: [[0.98010289 0.99068763 0.99206148 0.9913377 ]]
net_o: [[1.88590698]]
out_o: [[0.86828814]]
y_input: 0.8677
f(x): [[0.86828814]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-7: Benar

Data Testing Ke-8
net_h: [[3.80807933 4.55005315 4.69005632 4.53807199]]
out_h: [[0.97829098 0.98954384 0.99089745 0.98941915]]
net_o: [[1.88316547]]
out_o: [[0.8679743]]
y_input: 0.8677
f(x): [[0.8679743]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-8: Benar
```

Gambar 4.40 Hasil Forward Data Testing

```
Data Testing Ke-9
net_h: [[4.30809174 5.16006146 5.31206513 5.26308325]]
out_h: [[0.98671954 0.99429143 0.99509247 0.99484738]]
net_o: [[1.89297495]]
out_o: [[0.86909436]]
y_input: 0.8677
f(x): [[0.86909436]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-9: Benar

Data Testing Ke-10
net_h: [[4.37508827 5.22505913 5.38906267 5.3360801 ]]
out_h: [[0.98756943 0.99464875 0.9954545  0.99520836]]
net_o: [[1.89377278]]
out_o: [[0.8691851]]
y_input: 0.8677
f(x): [[0.8691851]]
bobot input: [[0.20000662 0.30000443 0.4000047  0.500006  ]
 [0.30000863 0.40000578 0.50000613 0.60000784]
 [0.40001505 0.50001008 0.60001069 0.70001366]
 [0.50001331 0.60000892 0.70000944 0.80001207]
 [0.60000505 0.70000338 0.80000359 0.90000458]
 [0.70001159 0.80000776 0.90000823 0.30001052]
 [0.80000737 0.90000494 0.30000524 0.50000669]]
bobot output: [[0.30119247]
 [0.40120876]
 [0.50121122]
 [0.60121033]]
Hasil pengenalan foto ke-10: Benar
```

Gambar 4.41 Hasil Forward Data Testing

4.1.8 Hasil Hitung Akurasi



Gambar 4.42 Hasil Hitung Akurasi

4.2 Analisa

Hasilnya bisa salah dikarenakan out_o < y_input.

BAB V
KESIMPULAN

5.1 Kesimpulan

Proyek ini berhasil mengimplementasikan jaringan syaraf tiruan (JST) dengan algoritma propagasi balik untuk pengenalan wajah secara manual menggunakan bahasa Python. Dataset yang digunakan terdiri dari Foto B.J. Habibie dan Soeharto. Implementasi jaringan syaraf tiruan ini mampu mengenali dan membedakan wajah dari kedua tokoh tersebut dengan tingkat akurasi yang memadai. Melalui berbagai tahap pengujian, hasil menunjukkan bahwa algoritma propagasi balik dapat digunakan secara efektif untuk tujuan pengenalan wajah. Proyek ini juga memberikan kode Python yang terstruktur dan mudah dipahami, sehingga dapat dijadikan referensi untuk implementasi serupa. Analisis dan pengujian yang dilakukan menunjukkan bahwa metode ini dapat diterapkan untuk berbagai aplikasi pengenalan wajah dengan hasil yang akurat dan efisien.

REFERENSI

1) Haykin, S. (2008). Neural Networks and Learning Machines (3rd ed.). Pearson.
2) Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
3) Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural Network-Based Face Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1), 23-38.
4) Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face Recognition: A Literature Survey. ACM Computing Surveys (CSUR), 35(4), 399-458.
5) Wiskott, L., Fellous, J. M., Krüger, N., & von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 775-779.

LAMPIRAN

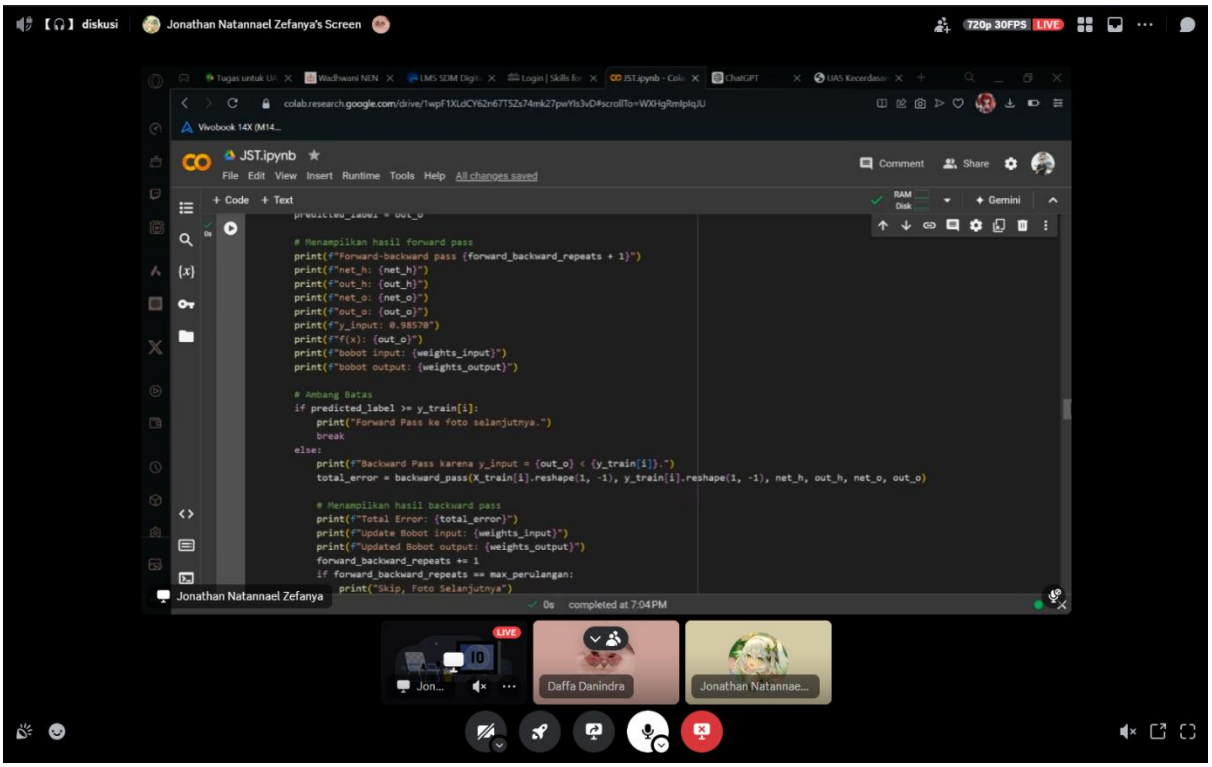
1) Foto B.J.Habibie

Gambar 5.1 Foto B.J.Habibie

2) Foto Soeharto

Gambar 5.2 Foto Soeharto

3) Melakukan Kerja Kelompok Melalui Discord



Gambar 5.3 Melakukan Kerja Kelompok Melalui Discord

4) Seluruh Data dan Foto yang digunakan untuk Data https://drive.google.com/drive/folders/1029XCU7gsLhGe5Qw5ZODYWVLW279aQ8s?usp=sharing

SIAPA MENGERJAKAN APA:

1. Jonathan Natannael Zefanya (1152200024) – Mengerjakan Program, Mengerjakan Soal Etika
2. Daffa Danindra (1152200028) – Mengambil Dan Membuat Seluruh Data Yang Dibutuhkan, Mengerjakan Laporan, Membuat PPT, Menganalisa dan Membantu Pembuatan program
3. Ikhwan El Faris (1152200009) – TIDAK MENGERJAKAN APA-APA
4. Dethalisa Aura Kirana (1152200037) – TIDAK MENGERJAKAN APA-APA