Seminar Nasional Fortei Regional 7

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

KLASIFIKASI pH AIR MENGGUNAKAN ARTIFICIAL NEURAL NETWORK BERDASARKAN WARNA RGB

[1] Anggar Kusuma, [2] Erdiaz Rakha Praditya, [3]Ilyas Hamzah Alkatiri, [4]Muhammad Faris Pradana, [5]Nasihol Fattah, [6]Novendra Setyawan

[1,2,3,4,5,6] Jurusan Teknik Elektro, Universitas Muhammadiyah Malang, Malang

[1]anggark38@gmail.com, [2]erdiazrakha@gmail.com, [3]mrilyasalkatiri@gmail.com, [4]fcharjaymania@gmail.com, [5]fattahsmart69@gmail.com

The purposes of this system design are: 1) to build a monitoring system to determine a pH level, 2) to make it easier for users to determine water pH levels more accurately.The method of making this system uses an Artificial Neural Network with a pH sample of 0-14. The research stages refer to the development research model of Rizky Satrio Wibowo and Muhammad Ali which consists of: 1)alat pengukur warna dari tael indikator universal pH yang diperbesar berbasis mikrokontroler arduino, 2) System making, 3) System testing, 4) implementation. The accuracy value of this system is 100% based on the experiments we have done on different input data repeatedly. The completion of this system uses the ANN (Artificial Neural Network) method with the Multilayer Neural Network architecture, where hidden layers located between the input layer and the output layer. In a multilayer neural network architecture, there can be more than one hidden layer. In this multi-layer neural network architecture, it can solve more complicated and complex problems, because it uses a non-linear activation function.

Keywords — ANN, pH air, Multilayer Neural Network, Arduino

Tujuan dari perancangan sistem ini adalah: 1) untuk membangun alat monitoring agar bisa mengetahui suatu kadar pH, 2) untuk mempermudah pengguna agar menentukan kadar pH air yang lebih akurat. Metode pembuatan sistem ini menggunakan Artificial Neural Network dengan sampel pH sebesar 0-14. Tahap-tahap penelitian mengacu pada model penelitian pengembangan Rizky Satrio Wibowo dan Muhammad Ali yang terdiri dari: 1) alat pengukur warna dari tael indikator universal pH yang diperbesar berbasis mikrokontroler arduino, 2) pembuatan alat, 3) pengujian alat, 4) implementasi. Nilai keakuratan alat ini sebesar 100% berdasarkan percobaan yang telah kami lakukan pada data input yang berbeda-beda secara berulang. Penyelesaian sistem ini menggunakan metode ANN (Artificial Neural Network) dengan arsitektur Multilayer Neural Network, dimana semua layer yang terletak diantara input layer dan output layer merupakan hidden layer. Pada arsitektur multilayer neural network hidden layer dapat lebih dari satu. Pada arsitektur multi layer neural network ini dapat menyelesaikan permasalahan yang lebih rumit dan kompleks, dikarenakan menggunakan fungsi aktivasi non-linier.

Kata Kunci : *ANN, pH air, Multilayer Neural Network, Arduino*

I. PENDAHULUAN

Asam dan basa adalah unsur kimia yang sangat penting. Istilah asam (acid) berasal dari bahasa Latin (*Acetum*) yang berarti cuka. Istilah bahasa (*Alkali*) berasal dari bahasa Arab yang berarti abu. Sifat asam basa dari suatu larutan dapat ditunjukkan dengan mengukur pH. pH adalah parameter yang berfungsi untuk menunjukkan tingkat keasaman larutan. Larutan asam memiliki pH kurang dari 7, sedangkan larutan netral memiliki pH sebesar 7, dan larutan basa memiliki pH lebih dari 7.

Cara yang digunakan untuk menentukan level pH biasanya menggunakan indikator. Cara menentukan indikator tersebut biasanya menggunakan kertas lakmus, larutan fenolftalein, brom timol biru, metil merah, dan metil orange. Ada beberapa cara yang digunakan oleh ilmuwan dan atau orang awam untuk mengukur pH diantaranya dengan menggunakan tabel indikator pH, pH meter, kertas lakmus ataupun perhitungan dengan mengetahui konsentrasi suatu larutan tersebut.[1]

pH air terbagi menjadi 3 jenis yaitu, asam, basa dan netral, masing-masing jenis tersebut dibedakan dengan warna, yaitu merah untuk asam, hijau untuk netral dan biru untuk basa.[2] Setiap warna memiliki batas yaitu dari 0 (minimum) hingga 255 (maximum). Semakin besar angka dari suatu warna maka kondisi air semakin merujuk ke label warna tersebut. Setiap jenis pH air juga memiliki level sendiri yaitu dari level 0 (minimum) hingga 14 (maximum). Untuk level 0-5 menandakan asam, semakin kecil angkanya maka semakin asam (0 lebih asam dibandingkan 5).

Gambar 1. Tabel warna asam

Untuk 6-8 menandakan netral, netral memiliki level yang paling sedikit dibanding asam dan basa, hanya terdapat 3 level, kondisi netral terbaik adalah pada level 7.

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

Gambar 2. Tabel warna netral

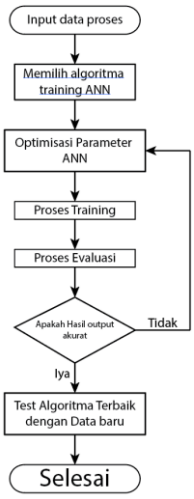
Dan yang terakhir level 9-14 menandakan basa, semakin besar angkanya maka menandakan bahwa semakin basa air tersebut (14 lebih basa dibandingkan 9).[3]

Gambar 3. Tabel Warna basa

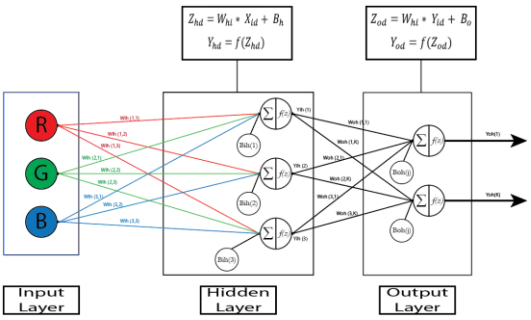
Masing-masing metode pengukuran memiliki kelebihan dan kekurangan, sehingga penelitian ini akan mecoba dengan metode mendeteksi pH warna RGB berbasis Artificial Neural Network, sehingga mampu meningkatkan nilai akurasi pengukuran dan menghindari terjadinya kesalahan baca nilai hasil pengukuran.

II. METODE PENELITIAN

Pada penelitian ini menggunakan Klasifikasi Ph air dilakukan dengan Artficial Neural Network dengan input adalah Warna RGB, inspirasi untuk model ANN berasal dari jaringan saraf otak manusia yang mana jaringan-jaringan tersebut mengetahui pengetahuan melalui proses pembelajaran.[4] kemampuan belajar jaringan saraf ini memberi keuntungan dalam memecahkan masalah kompleks yang solusi analitik dan numerik nya sulit didapat.[5] Pada penelitian kali ini kami menggunakan ANN dengan arsitektur Multilayer Neural Network, dimana semua layer yang terletak diantara input layer dan output layer merupakan hidden layer. Pada arsitektur multilayer neural network hidden layer dapat lebih dari satu.[6] Pada arsitektur ini dapat menyelesaikan permasalahan yang lebih rumit dan kompleks, dikarenakan menggunakan fungsi aktivasi non-linier. Proses pemrosesan data dapat dijelaskan dari flowchart dibawah ini



Gambar 4. Flowchart Metode Pelaksanaan



Gambar 5. Multi Layer Neural Network

Pada ANN, terdapat dua pemrosesan yang dilakukan pada Hidden Layer dan Output Layer, Pada Hidden Layer derdapat sebuah persamaan yaitu

$$Z_{hd} = W_{hi} * X_{id} + B_h \quad \text{.....(1)}$$

Keterangan :

Z_{hd} = hasil perkalian antara input layer (Variable Input) dikalikan pada bobot masing-masing neuron.

W_{hi} = merupakan suatu bobot yang ada diantara input layer dengan hidden layer (h = jumlah hidden, i = jumlah variable input)

X_{id} = ukuran dari data input (i = jumlah variable input, d = jumlah data)

B_h = merupakan bias pada hidden antara input layer dengan hidden layer

Setelah itu masuk pada aktivasi yang dituliskan dengan rumus

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

……..(2)

$$Y_{hd} = f(Z_{hd})$$

Dimana pada $f(Z_{hd})$ dapat berupa sigmoid, bipolar sigmoid ataupun bisa berupa linier, setelah itu pada ouput layer juga terdapat perhitungan dengan suatu rumus yang mana dapat dituliskan pada persamaan 3:

……(3)

$$Z_{od} = W_{oh} * Y_{hd} + B_o$$

Z_{od} = hasil perkalian antara hidden layer dengan bobot antara hidden layer dengan output Layer

W_{oh} = bobot yang menghubungkan antara ouput dengan hidden layer (o = jumlah output, h = jumlah hidden layer)

Y_{hd} = merupakan perkalian antara hidden layer (h = jumlah hidden, d = jumlah data)

B_o = merupakan bias pada hidden antara hidden layer dengan output

Setelah itu juga akan dilakukan aktivasi ulang dengan persamaan 4.

……..(4)

$$Y_{od} = f(Z_{od})$$

Dimana dari kedua aktivasi yang telah dilakukan yaitu pada hidden layer dengan input ( ) dan hidden layer dengan Output , dapat memilki fungsi yang berbeda-beda, seperti aktivasi input berupa sigmoid maka aktivasi output dapat juga berbeda dari sigmoid ataupun sama dengan sigmoid.

Y_{hd}

Y_{od}

Rumus diatas akan kita transformasi kedalam bentuk script coding menggunakan bahasa Python, terdapat dua program pada percobaan kali ini, dimana program pertama merupakan script algoritma dari ANN, dan yang kedua merupakan script dari klasifikasi.

Pertama kita membuat script coding untuk algoritma ANN menggunakan library numpy, dan membuat class baru yang bernama neural network

```
import numpy as np
class neuralnetwork:
```

Gambar 7. Perintah library

Lalu setelah itu kita menginisialisasi bobot, dengan menuliskan perintah pada Gambar 8 berikut :

```
    def __init__(self,Ninput,Nhidden,Noutput):
        self.Nin=Ninput
        self.Nhid=Nhidden
        self.Nout=Noutput
        self.InitializeWeight()
```

Gambar 8. Perintah inisialisasi bobot

Perintah Ninput untuk jumlah input, Nhidden untuk jumlah hidden dan Noutput untuk jumlah output nya, berikutnya kami menginisialisasi bobotnya dengan perintah sebagai berikut :

```
    def InitializeWeight(self):
        self.Whi=np.random.rand(self.Nhid,self.Nin)*0.1
        self.Woh=np.random.rand(self.Nout,self.Nhid)*0.1
        self.Bh=np.zeros((self.Nhid,1))
        self.Bo=np.zeros((self.Nout,1))
        self.Zhd=np.zeros((self.Nhid,1))
        self.Zod=np.zeros((self.Nout,1))
        self.Yhd=np.zeros((self.Nhid,1))
        self.mseLoging=np.array([])
```

Gambar 9. Perintah pemasukan bentuk matrix layer

Penulisan perintah tersebut merupakan bentuk rumus dari

$$Z_{hd} = W_{hi} * X_{id} + B_h$$
$$Y_{hd} = f(Z_{hd})$$
$$Z_{od} = W_{oh} * Y_{hd} + B_o$$
$$Y_{od} = f(Z_{od})$$

Setelah itu kita memasukkan perintah fungsi dari sigmoid

```
    def sigmoid(self,valin):
        return (1/(1+np.exp(-(valin))))
```

Gambar 10. Perintah fungsi sigmoid

Kemudian menuliskan juga turunan dari fungsi sigmoid

```
    def derSigmoid(self,valin):
        return (valin*(1-valin))
```

Gambar 11. Perintah fungsi sigmoid

Dimana valin*(1-valin) berarti , lalu pada perintah berikutnya kami menggunakan dua buah keadaan bias yang mana, kondisi pertama menggunakan bias atau BIAS = 1, dan kondisi kedua tidak menggunakan BIAS atau BIAS = 0.

$y * (1 - y)$

```
    def Forward(self,inputData,bias=False):
        if bias==True:
            self.Zhd=np.dot(self.Whi, inputData)+self.Bh
            self.Yhd=self.sigmoid(self.Zhd)
            self.Zod=np.dot(self.Woh, self.Yhd)+self.Bo
            self.Yod=self.sigmoid(self.Zod)
        else:
            self.Zhd=np.dot(self.Whi, inputData)
            self.Yhd=self.sigmoid(self.Zhd)
            self.Zod=np.dot(self.Woh,self.Yhd)
            self.Yod=self.sigmoid(self.Zod)
        return self.Yod.T
```

Gambar 12. Printah klasifikasi Bias

Namun pada script ini, data belum di transpose. Transpose merupakan perubahan kolom menjadi baris, dan baris menjadi

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

kolom pada matrix, maka harus kita transpose pada script main, lalu kita menuliskan script perintah untuk proses training neural networknya. Script beserta keterangannya dapat dilihat pada gambar dibawah ini :

```
def Train(self,inDataSet,outDataSet,maxIt,lr,bias=False):
        for it in range(maxIt):
            Yn=self.Forward(inDataSet,bias)#proses forward neuron
            error=(outDataSet - Yn)#menghitung error
            mse=np.square(error).mean()#menghitung mean square error (MSE)
            self.mseLoging=np.append(self.mseLoging,mse)#Logging mse
            deYo=error*self.derSigmoid(Yn)
            #Menghitung Turunan Bobot pada Hidden to Output
            dWoh=np.dot(self.Yhd, deYo)
            dBo=np.mean(deYo,axis=0,keepdims=True).T
            #Menghitung Turunan Bobot pada Input to Hidden
            dBh=(np.dot(deYo, self.Woh)*self.derSigmoid(self.Yhd).T)
            dWhi=np.dot(inDataSet, dBh)
            dBh=(np.mean(dBh,axis=0,keepdims=True)).T
            #Update Bobot
            self.Woh=self.Woh+lr*dWoh.T # .T adalah operator transpose
            self.Whi=self.Whi+lr*dWhi.T # .T adalah operator transpose
            #Update bias jika digunakan bias=True
            if bias==True:
                self.Bo=self.Bo+lr*dBo
                self.Bh=self.Bh+lr*dBh
            if it%10==0:
                print("Iteration: {} MSE: {}".format(it,mse))
        return self.mseLoging, mse
```

Gambar 13. Perintah Training

Pada training kali ini kami menggunakan arsitektur Feed Foward neural network. Informasi masuk dari input layer menuju hidden layer sampai output layer dalam satu arah ‘forward’, tidak melakukan proses loop seperti pada Recurrent Neural Network. Pada arsitektur Feed Forward neural network kita menggunakan Multi Layer Preceptron yang melibatkan banyak layer yang saling terhubung dengan cara feed forward, dimana tiap neuron pada layer terhubung dengan semua neuron di layer selanjutnya. Banyak implementasi Multi Layer Perceptron menggunakan Sigmoid function sebagai activation funtion-nya.

Multi Layer Perceptron menggunakan back propagation dalam proses training yang akan menghitung gradient dari loss function yang berhubungan dengan ‘weight’ tiap koneksi neuron dan akan meminimalkan loss saat ‘weight’ diubah.

Setelah menuliskan script coding algoritma neural network, maka dibutuhkan suatu script lagi untuk mengeksekusi algoritma neural network dengan data-data yang sudah disiapkan, fungsi dari script ini untuk memasukkan data yang sudah disiapkan dan mengklasifikasi data tersebut dengan algoritma neural network yang sudah dibuat pada script sebelumnya.

```
import numpy as np
from neuralnetwork import neuralnetwork
import matplotlib.pyplot as pylot

inDataSet=np.genfromtxt("phair.csv",delimiter=";",skip_header=1,usecols=(0,1,2))
outDataSet=np.genfromtxt("phair.csv",delimiter=";",skip_header=1,usecols=(3,4))
inDatatest=np.genfromtxt("phairtes.csv",delimiter=";",skip_header=1,usecols=(0,1,2))

maxVal=255
minVal=0
maxph=14
minph=0

inDataSetNorm=(inDataSet-minVal)/(maxVal-minVal)
outDataSetNorm=(outDataSet-minph)/(maxph-minph)
inDatatestNorm=(inDatatest-minVal)/(maxVal-minVal)

nn=neuralnetwork(3,20,2)
mseLoging,mse=nn.Train(inDataSetNorm.T,outDataSetNorm,10000 , 0.4, bias=True)

outtest=nn.Forward(inDatatestNorm.T, True)
outtestnorm=(outtest*(maxph-minph))+minph
outClassification=np.round(outtestnorm)
print(outClassification)
pylot.plot(mseLoging)
pylot.show()
```

Gambar 14. Listing main program

Dengan menggunakan library yang sama yaitu numpy, dan matplotlib.pylot, Setelah itu menuliskan perintah diatas untuk input data yang berupa file csv, Karena akan terdapat dua output maka masukan nilai maksimal dan minimum dari kedua output tersebut dengan menuliskan perintah seperti pada baris ke 9 hingga 12 pada script main program diatas, Dikarenakan input dan output dalam sigmoid hanya bisa pada range 0 hingga 1, sedangkan data input dan output lebih dari 1 maka kita melakukan proses normalisasi untuk input datatest, output datatest dan input datatest agar range angkanya dapat memunculkan angka yang lebih bervariasi tidak hanya terpaku pada angka 0 dan 1 saja. Maka kita diharuskan menuliskan perintah seperti pada baris ke 14 hingga 16 pada script programmain diatas. Setelah itu dilakukannya proses setting parameter, dalam parameter tersebut terdapat hidden layer, iterasi, dan learning rate. Pengaturan parameter dilakukan berulang kali dengan kombinasi parameter yang berbeda-beda sampai ditemukan hasil yang paling memuaskan atau tingkat akurasi ouput yang akurat, untuk melakukan proses setting parameter maka kami menuliskan perintah seperti pada baris 18 hingga 19. Selanjutnya untuk menampilkan hasil dari perintah diatas maka kami menuliskan perintah seperti pada baris ke 21 hingga 26 .

A. Gambar dan Tabel

B	G	R	level	label	B	G	R	level	label
36	27	231	0	1	36	27	231	0	Asam
36	84	250	1	1	36	84	250	1	Asam
37	164	255	2	1	37	164	255	2	Asam
22	205	255	3	1	22	205	255	3	Asam
38	223	221	4	1	38	223	221	4	Asam
29	214	148	5	1	29	214	148	5	Asam
0	181	76	6	2	0	181	76	6	Netral
13	156	0	7	2	13	156	0	7	Netral
92	166	0	8	2	92	166	0	8	Netral
184	191	0	9	3	184	191	0	9	Basa
200	136	1	10	3	200	136	1	10	Basa
198	76	0	11	3	198	76	0	11	Basa
180	38	51	12	3	180	38	51	12	Basa
181	22	72	13	3	181	22	72	13	Basa
140	17	57	14	3	140	17	57	14	Basa

Tabel 1. Data pH air

III. HASIL DAN PEMBAHASAN

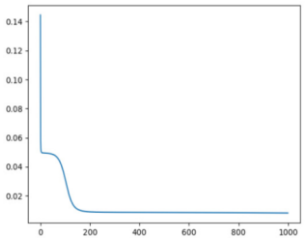
Pada percobaan kaliini kami melakukan proses learning yang mana prosestersebut bertujuan untuk melatih sistem kecerdasan buatan tersebut untuk memahami data data input yan gakan diberikan dan meprosesnya dengan algoritma neural network, disini kami melakukan beberapa testing dengan beragam variasi parameter yang berbeda beda seperti hidden layer, iterasi, dan learning rate. Berikut merupakan beberapa hasil percobaan yang telah kami lakukan :

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

```
maxVal=255
minVal=0
maxph=14
minph=0

inDataSetNorm=(inDataSet-minVal)/(maxVal-minVal)
outDataSetNorm=(outDataSet-minph)/(maxph-minph)
inDatatestNorm=(inDatatest-minVal)/(maxVal-minVal)

nn=neuralnetwork(3,10,2)
mseLoging,mse=nn.Train(inDataSetNorm.T,outDataSetNorm,1000 , 0.3, bias=True)

outtest=nn.Forward(inDatatestNorm.T, True)
outtestnorm=(outtest*(maxph-minph))+minph
outClassification=np.round(outtestnorm)
#print(outClassification)
pylot.plot(mseLoging)
pylot.show()
```

```
Iteration: 920 MSE: 0.00827938547611
Iteration: 930 MSE: 0.00826946769347
Iteration: 940 MSE: 0.00825927590766
Iteration: 950 MSE: 0.00824880138514
Iteration: 960 MSE: 0.00823803520887
Iteration: 970 MSE: 0.00822696828037
Iteration: 980 MSE: 0.00821559132239
Iteration: 990 MSE: 0.00820389488213
```

Gambar 21. Training dengan Hidden layer 10

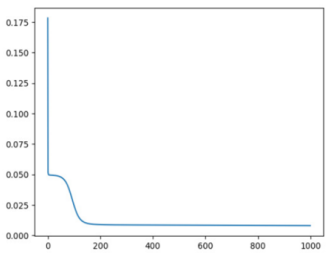


Gambar 22. Gambar Grafik Training dengan Hidden layer 10

```
maxVal=255
minVal=0
maxph=14
minph=0

inDataSetNorm=(inDataSet-minVal)/(maxVal-minVal)
outDataSetNorm=(outDataSet-minph)/(maxph-minph)
inDatatestNorm=(inDatatest-minVal)/(maxVal-minVal)

nn=neuralnetwork(3,20,2)
mseLoging,mse=nn.Train(inDataSetNorm.T,outDataSetNorm,1000 , 0.3, bias=True)

outtest=nn.Forward(inDatatestNorm.T, True)
outtestnorm=(outtest*(maxph-minph))+minph
outClassification=np.round(outtestnorm)
#print(outClassification)
pylot.plot(mseLoging)
pylot.show()
```

```
Iteration: 920 MSE: 0.00820563115895
Iteration: 930 MSE: 0.00819402418663
Iteration: 940 MSE: 0.00818213763383
Iteration: 950 MSE: 0.00816996535791
Iteration: 960 MSE: 0.00815750118371
Iteration: 970 MSE: 0.00814473898881
Iteration: 980 MSE: 0.00813167230436
Iteration: 990 MSE: 0.00811829512801
```

Gambar 23. Training dengan Hidden layer 20

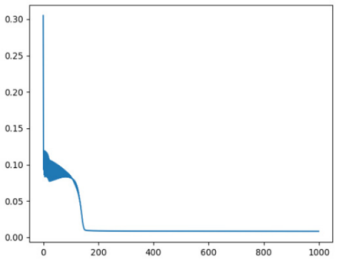


Gambar 24. Gambar Grafik Training dengan Hidden layer 20

```
maxVal=255
minVal=0
maxph=14
minph=0

inDataSetNorm=(inDataSet-minVal)/(maxVal-minVal)
outDataSetNorm=(outDataSet-minph)/(maxph-minph)
inDatatestNorm=(inDatatest-minVal)/(maxVal-minVal)

nn=neuralnetwork(3,50,2)
mseLoging,mse=nn.Train(inDataSetNorm.T,outDataSetNorm,1000 , 0.3, bias=True)

outtest=nn.Forward(inDatatestNorm.T, True)
outtestnorm=(outtest*(maxph-minph))+minph
outClassification=np.round(outtestnorm)
#print(outClassification)
pylot.plot(mseLoging)
pylot.show()
```

```
Iteration: 920 MSE: 0.00840131239333
Iteration: 930 MSE: 0.00839542220124
Iteration: 940 MSE: 0.0083894036188
Iteration: 950 MSE: 0.00838325343622
Iteration: 960 MSE: 0.00837696837465
Iteration: 970 MSE: 0.00837054508519
Iteration: 980 MSE: 0.00836398014801
Iteration: 990 MSE: 0.00835727007147
```

Gambar 25. Training dengan Hidden layer 50



Gambar 26. Gambar Grafik Training dengan Hidden layer 50

Setelah melakukan percobaan dari beberapa variasi hidden layer kami mendapatkan hasil sebagai berikut : hidden layer 10 mendapatkan nilai MSE sebesar 0.008203897488213, hidden layer 20 mendapatkan MSE sebesar 0.00811829512801 dan hidden layer 50 mendapatkan nilai MSE sebesar 0.00835727007147. Dari ketiga data tersebut hidden layer dengan nilai MSE terkecil adalah 20 dengan MSE sebesar 0.008118295.

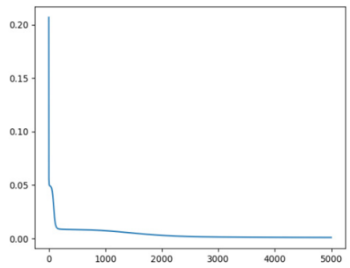
Selanjutnya kami melakukan pengujian terhadap iterasi dan didapatkan data sebagai berikut :

```
maxVal=255
minVal=0
maxph=14
minph=0

inDataSetNorm=(inDataSet-minVal)/(maxVal-minVal)
outDataSetNorm=(outDataSet-minph)/(maxph-minph)
inDatatestNorm=(inDatatest-minVal)/(maxVal-minVal)

nn=neuralnetwork(3,25,2)
mseLoging,mse=nn.Train(inDataSetNorm.T,outDataSetNorm,5000 , 0.3, bias=True)

outtest=nn.Forward(inDatatestNorm.T, True)
outtestnorm=(outtest*(maxph-minph))+minph
outClassification=np.round(outtestnorm)
#print(outClassification)
pylot.plot(mseLoging)
pylot.show()
```

```
Iteration: 4920 MSE: 0.0010350999525
Iteration: 4930 MSE: 0.001034127419
Iteration: 4940 MSE: 0.0010331604953
Iteration: 4950 MSE: 0.0010321991271?
Iteration: 4960 MSE: 0.00103124326094
Iteration: 4970 MSE: 0.00103029284353
Iteration: 4980 MSE: 0.00102934782242
Iteration: 4990 MSE: 0.00102840814564
```

Gambar 27. Training dengan Iterasi 5000

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

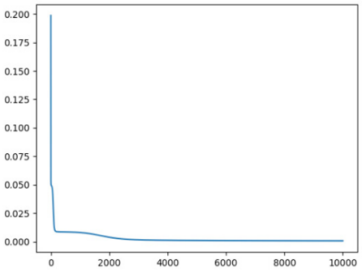


Gambar 28. Gambar Grafik Training dengan Iterasi 5000

```
maxVal=255
minVal=0
maxph=14
minph=0

inDataSetNorm=(inDataSet-minVal)/(maxVal-minVal)
outDataSetNorm=(outDataSet-minph)/(maxph-minph)
inDatatestNorm=(inDatatest-minVal)/(maxVal-minVal)

nn=neuralnetwork(3,25,2)
mseLoging,mse=nn.Train(inDataSetNorm.T,outDataSetNorm,10000 , 0.3, bias=True)

outtest=nn.Forward(inDatatestNorm.T, True)
outtestnorm=(outtest*(maxph-minph))+minph
outClassification=np.round(outtestnorm)
#print(outClassification)
pylot.plot(mseLoging)
pylot.show()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Iteration: 9920 MSE: 0.000752036712511
Iteration: 9930 MSE: 0.000751627776889
Iteration: 9940 MSE: 0.000751218977354
Iteration: 9950 MSE: 0.000750810312844
Iteration: 9960 MSE: 0.000750401782306
Iteration: 9970 MSE: 0.000749993384696
Iteration: 9980 MSE: 0.000749585118981
Iteration: 9990 MSE: 0.000749176984137
```

Gambar 29. Training dengan Iterasi 10000

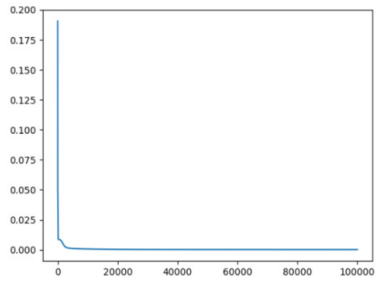


Gambar 30. Gambar Grafik Training dengan Iterasi 10000

```
maxVal=255
minVal=0
maxph=14
minph=0

inDataSetNorm=(inDataSet-minVal)/(maxVal-minVal)
outDataSetNorm=(outDataSet-minph)/(maxph-minph)
inDatatestNorm=(inDatatest-minVal)/(maxVal-minVal)

nn=neuralnetwork(3,25,2)
mseLoging,mse=nn.Train(inDataSetNorm.T,outDataSetNorm,100000 , 0.3, bias=True)

outtest=nn.Forward(inDatatestNorm.T, True)
outtestnorm=(outtest*(maxph-minph))+minph
outClassification=np.round(outtestnorm)
#print(outClassification)
pylot.plot(mseLoging)
pylot.show()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Iteration: 99920 MSE: 0.000162136270888
Iteration: 99930 MSE: 0.000162132814232
Iteration: 99940 MSE: 0.000162129358101
Iteration: 99950 MSE: 0.000162125902478
Iteration: 99960 MSE: 0.000162122447368
Iteration: 99970 MSE: 0.000162118992772
Iteration: 99980 MSE: 0.000162115538869
Iteration: 99990 MSE: 0.000162112085121
```

Gambar 31. Training dengan Iterasi 100000



Gambar 32. Gambar Grafik Training dengan Iterasi 100000

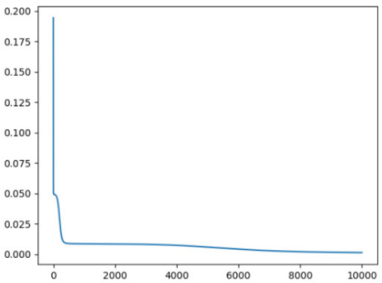
Setelah melakukan percobaan dari beberapa variasi iterasi kami mendapatkan hasil sebagai berikut : iterasi 5000 mendapatkan nilai MSE sebesar 0.00102846814564, iterasi 10000 mendapatkan MSE sebesar 0.000749176984137 dan iterasi 100000 mendapatkan nilai MSE sebesar 0.000162112085121. Dari ketiga data tersebut iterasi dengan nilai MSE terkecil adalah 10000 dengan MSE sebesar 0.000749176984137.

Selanjutnya kami melakukan pengujian terhadap learning rate dan didapatkan data sebagai berikut.

```
maxVal=255
minVal=0
maxph=14
minph=0

inDataSetNorm=(inDataSet-minVal)/(maxVal-minVal)
outDataSetNorm=(outDataSet-minph)/(maxph-minph)
inDatatestNorm=(inDatatest-minVal)/(maxVal-minVal)

nn=neuralnetwork(3,25,2)
mseLoging,mse=nn.Train(inDataSetNorm.T,outDataSetNorm,10000 , 0.12, bias=True)

outtest=nn.Forward(inDatatestNorm.T, True)
outtestnorm=(outtest*(maxph-minph))+minph
outClassification=np.round(outtestnorm)
#print(outClassification)
pylot.plot(mseLoging)
pylot.show()
```

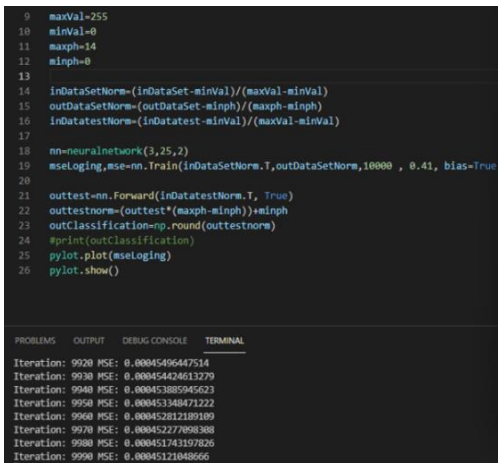
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Iteration: 9920 MSE: 0.00143298824467
Iteration: 9930 MSE: 0.00143091082182
Iteration: 9940 MSE: 0.00142892267859
Iteration: 9950 MSE: 0.00142694174758
Iteration: 9960 MSE: 0.00142497395487
Iteration: 9970 MSE: 0.00142301323775
Iteration: 9980 MSE: 0.00142106152698
Iteration: 9990 MSE: 0.00141911875563
```

Gambar 33. Training dengan Learning rate 0.12

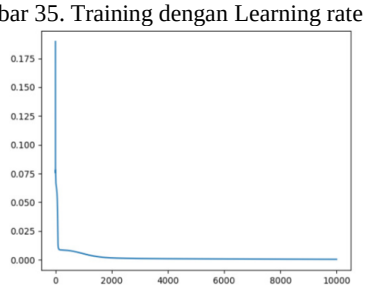


Gambar 34. Gambar Grafik Training Learning rate 0.12

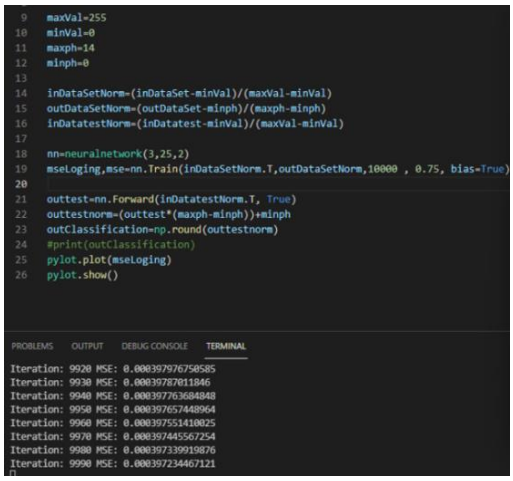
ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551



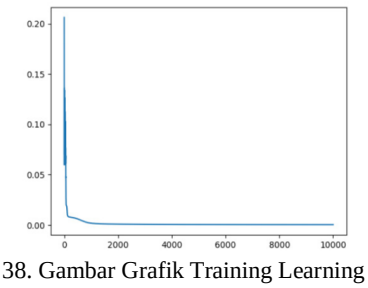
Gambar 35. Training dengan Learning rate 0.41



Gambar 36. Gambar Grafik Training Learning rate 0.41



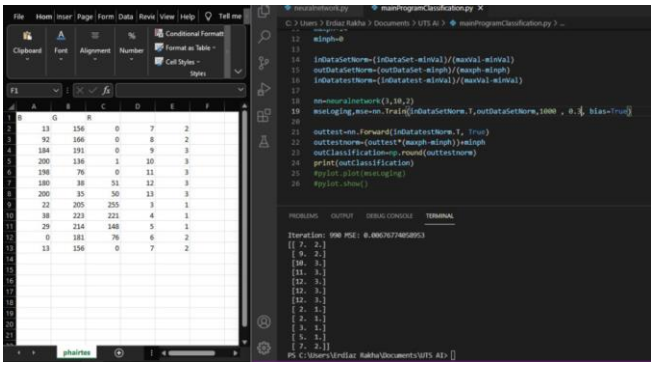
Gambar 37. Training dengan Learning rate 0.75

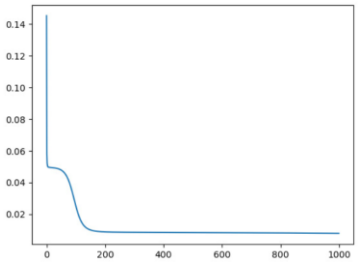


Gambar 38. Gambar Grafik Training Learning rate 0.75

Setelah melakukan percobaan dari beberapa variasi learning rate kami mendapatkan hasil sebagai berikut: learning rate 0.12 mendapatkan nilai MSE sebesar 0.00141911875563, learning rate 0.41 mendapatkan MSE sebesar 0.00045121048666 dan learning rate 0.75 mendapatkan nilai MSE sebesar 0.000397234467121. Dari ketiga data tersebut learning rate dengan nilai MSE terkecil adalah 0.75 dengan MSE sebesar 0.000397234467121.

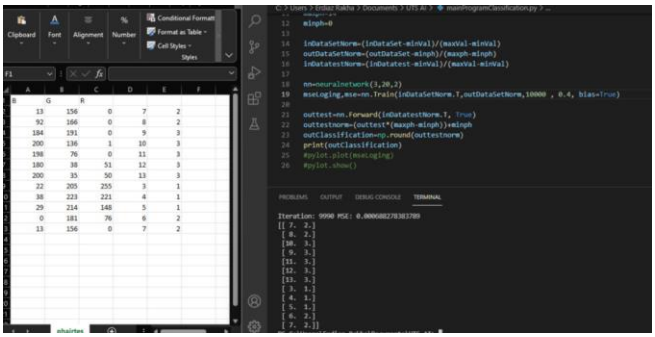
Setelah itu kami melakukan pengujian sembari memilih parameter yang paling akurat, kami melakukan 4 kali percobaan dan didapatkan hasil sebagai berikut


Gambar 39. Gambar Percobaan dengan Hasil Testing ke-1

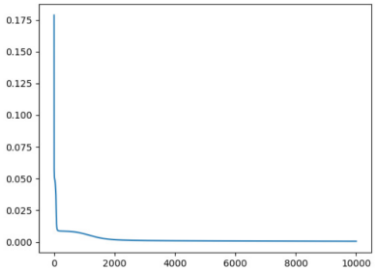


Gambar 40. Gambar Grafik Percobaan dengan Hasil Testing ke-1

Pada testing pertama kita menggunakan hidden layer 10, iterasi 1000 dan learning rate 0.3, menghasilkan MSE sebesar 0.000438790519135 dan dari 12 data yang diberikan menghasilkan tingkat akurasi sebesar 35% dengan 9 data salah dan 3 benar.

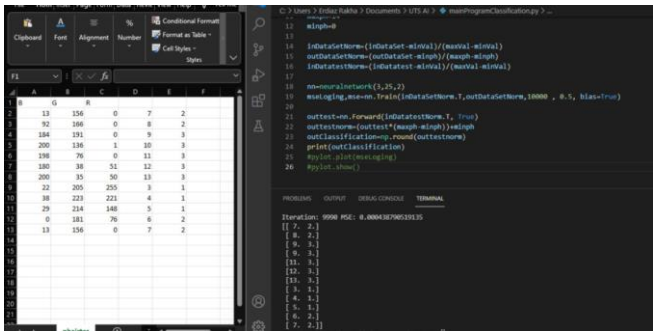

Gambar 41. Gambar Percobaan dengan Hasil Testing ke-2

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

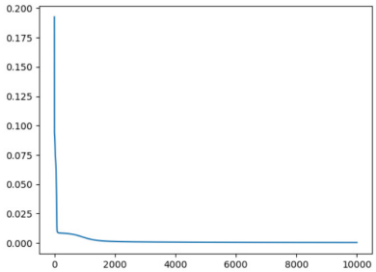


Gambar 42. Gambar Grafik Percobaan dengan Hasil Testing ke-2

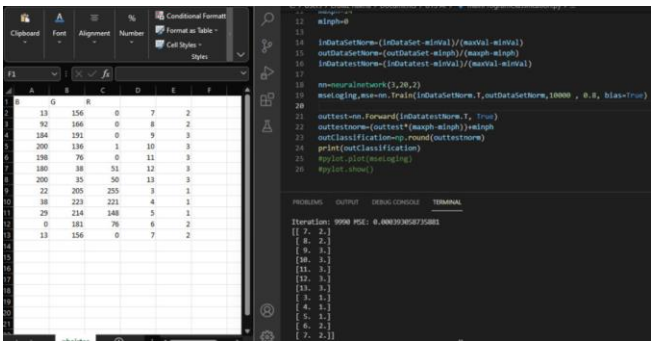
Pada testing kedua kita menggunakan hidden layer 20, iterasi 10000 dan learning rate 0.4, menghasilkan MSE sebesar 0.00676774058953 dan dari 12 data yang diberikan menghasilkan tingkat akurasi sebesar 83,33% dengan 2 data salah dan 10 benar.



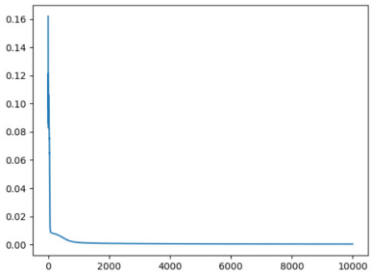
Gambar 43. Gambar Percobaan dengan Hasil Testing ke-3



Gambar 44. Gambar Grafik Percobaan dengan Hasil Testing ke-3



Gambar 45. Gambar Percobaan dengan Hasil Testing ke-4



Gambar 46. Gambar Grafik Percobaan dengan Hasil Testing ke-4

Pada testing keempat kita menggunakan hidden layer 20, iterasi 10000 dan learning rate 0.8, dari 12 data yang diberikan menghasilkan MSE sebesar 0.000393058735881dan menghasilkan tingkat akurasi sebesar 100% dengan 0 kesalahan data dan 12 data benar.

Pada tabel data yang akan kami uji, kami menandakan warna merah dengan kolom R(Red), warna hijau dengan kolom G(Green) dan warna biru dengan kolom B(Blue), kolom tersebut akan memberikan informasi seberapa dominan level pH pada air dengan cara memberikan informasi berupa angka yang memiliki range dari 0 sampai dengan 255. Dan untuk level pH diberikan pada indikator level dengan range angka 0 hingga dengan 14, dimana angka-angka tersebut didapat dari klasifikasi pengetesan pH berdasarkan angka yang diberikan pada kolom RGB dan angka 0 hingga 14 tersebut akan menandakan warna apa yang keluar dari proses klasifikasi tersebut. Hal ini berkaitan pada kolom setelahnya yaitu indikator label, dimana jika pada indikator level mengeluarkan angka 0 hingga 5 maka indikator label akan mengenalinya sebagai asam dan akan memberikan output angka 1 yang menandakan bahwa kondisi air asam. Jika pada indikator level mengeluarkan angka 6 hingga 8 maka pada indikator label akan memiliki output dengan angka 2 yang menandakan bahwa kondisi air netral dan untuk indikator level yang memiliki angka dari 9 hingga 14, maka indikator label akan memiliki output angka 3 yang menandakan bahwa kondisi air basa.

Secara sederhana cara kerja dari pemrosesan pengenalan pH air ini adalah dengan mengalkulasi angka yang diberikan pada kolom R,G, dan B, kemudian diklasifikasi ke dalam indikator level dan setelah itu diberi label pada indikator label apakah asam, netral atau basa dengan memberi label angka 1 untuk asam, 2 untuk netral, dan 3 untuk basa.

IV. KESIMPULAN

Perancangan dan pembangunan sistem klasifikasi warna dari tabel indikator universal pH yang berbasis artificial neural network berdasarkan warna RGB, dirancang sesuai dengan tujuan dan disusun dengan dua buah script program utama yaitu, script program algoritma neural network dan script main program clasification. Data yang digunakan adalah dataset kadar warna pH. Output hasil pengolahan data

ISSN (Print) : 2621-3540
ISSN (Online) : 2621-5551

ditampilkan oleh sebuah terminal. Nilai keakuratan sistem ini sebesar 100 % berdasarkan percobaan yang telah kami lakukan pada data input yang berbeda-beda secara berulang. Sistem ini mampu mengklasifikasi warna dengan akurat untuk pemindaian warna antara pH 0 sampai pH 14.

V. DAFTAR PUSTAKA

[1]. M. A. Rizky Satrio Wibowo, "ALAT PENGUKUR WARNA DARI TABEL INDIKATOR UNIVERSAL PH YANG DIPERBESAR BERBASIS MIKROKONTROLER ARDUINO," Jurnal Edukasi Elektro, pp. 99-109, 2019.

[2]. S. I. Zulfian Azmi, "SISTEM PENGHITUNG PH AIR PADA TAMBAK IKAN BERBASIS MIKROKONTROLLER," Jurnal SAINTIKOM, vol. 15, no. 2, pp. 101-108, 2016.

[3]. B. S. S. Jufriadi Karangan, "UJI KEASAMAN AIR DENGAN ALAT SENSOR pH DI STT MIGAS BALIKPAPAN," JURNAL KEILMUAN TEKNIK SIPIL, vol. 2, no. 1, pp. 65-72, 2019.

[4]. Kusumadewi, S., 2004, Membangun Jaringan Saraf Tiruan Menggunakan MATLAB & Excel Link. Yogyakarta: Penerbit Graha Ilmu.

[5]. Prathama, A. Y. (2018). PENDEKATAN ANN (ARTIFICIAL NEURAL NETWORK) UNTUK PENENTUAN PROSENTASE BOBOT PEKERJAAN DAN ESTIMASI NILAI PEKERJAAN STRUKTUR PADA RUMAH SAKIT PRATAMA. *Jurnal Teknosains*, *7*(1), 14. https://doi.org/10.22146/teknosains.30139

[6]. Yuniarti Trisna, I. R. T. R. H. M. M. (n.d.). PENGGUNAAN ARTIFICIAL NEURAL NETWORK (ANN) UNTUK MEMODELKAN VOLUME EKSPOR CRUDE PALM OIL (CPO) DI INDONESIA. *Ready Star-2*, 252.