

## Simulasi Jaringan Saraf Tiruan dengan *Neural Network Fitting Tool* (NFTool)

Yohanes Pracoyo Widi Prasetyo<sup>1✉</sup>

<sup>1</sup> Teknik Sipil, Teknik, Universitas 17 Agustus 1945 Banyuwangi, Indonesia

### Informasi Artikel

#### Riwayat Artikel

**Diserahkan** : 17-10-2023

**Direvisi** : 20-10-2023

**Diterima** : 23-10-2023

#### Kata Kunci:

*Neural Network Fitting Tool;*  
*Jaringan Saraf Tiruan;*  
*Learning rate neuron*

#### Keywords :

*Neural Network Fitting Tool;*  
*Artificial Neural Networks;*  
*Learning rate neuron.*

### ABSTRAK

Jaringan Saraf Tiruan (JST) merupakan model komputasi dengan mengadopsi jaringan sistem saraf manusia, jaringan ini tersusun dari *neuron* yang saling terhubung melalui sinapsis. JST digunakan untuk memecahkan berbagai masalah, seperti klasifikasi, regresi, dan pengenalan pola. *Neural Network Fitting Tool* (nftool) adalah alat yang disediakan oleh Matlab untuk membuat dan mengevaluasi JST dengan mudah. Nftool memungkinkan pengguna untuk mengatur topologi JST, memilih fungsi aktivasi dan pelatihan, serta mengevaluasi performa JST. Dalam penggunaannya, pengguna perlu memasukkan data pelatihan dan data validasi, dan menentukan jumlah lapisan dan jumlah neuron dalam setiap lapisan. Nftool juga dapat membantu pengguna dalam memilih parameter pelatihan, seperti kecepatan belajar dan momentum. Implementasinya, *nftool* bisa digunakan untuk memecahkan masalah regresi dan klasifikasi, namun pengguna perlu memahami konsep dasar JST dan memilih parameter yang sesuai untuk mendapatkan hasil yang akurat.

### ABSTRACT

*Artificial Neural Network (ANN) is a computational model that adopts the human nervous system network, this network is composed of neurons that are connected to each other via synapses. ANN is used to solve various problems, such as classification, regression, and pattern recognition. Neural Network Fitting Tool (nftool) is a tool provided by Matlab to easily create and evaluate ANNs. Nftool allows users to set the ANN topology, select activation and training functions, and evaluate ANN performance. To use it, users need to enter training data and validation data, and determine the number of layers and the number of neurons in each layer. Nftool can also assist users in selecting training parameters, such as learning rate and momentum. In its implementation, nftool can be used to solve regression and classification problems, but users need to understand the basic concepts of ANN and choose appropriate parameters to get accurate results.*

#### Corresponding Author :

Yohanes Pracoyo Widi Prasetyo

Teknik Sipil, Fakultas Teknik, Universitas 17 Agustus 1945 Banyuwangi

Jl. Adi Sucipto 26 Banyuwangi 68416

Email: widiprasetyo@untag-banyuwangi.ac.id

## PENDAHULUAN

Jaringan Saraf Tiruan (JST), karena kemampuannya untuk memodelkan dan memecahkan masalah yang kompleks, menjadi salah satu metode yang sangat dikenal dalam ilmu data dan pemrosesan sinyal, termasuk untuk mengoptimalkan dan meningkatkan rasio rekonstruksi citra dari data DCT yang dikompresi (Prasetyo, 2021). Namun, penggunaan JST sering kali memerlukan keahlian dan pengalaman khusus dalam pemrograman dan matematika. Untuk membantu pengguna yang kurang berpengalaman dalam membangun JST, Matlab menyediakan *Neural Network Fitting Tool (nftool)* sebagai salah satu fitur yang dapat digunakan dengan mudah (Du et al., 2019). Dengan *nftool*, pengguna dapat membangun dan mengevaluasi JST dengan cepat dan mudah tanpa harus memiliki pengetahuan yang mendalam tentang JST (Luo et al., 2019).

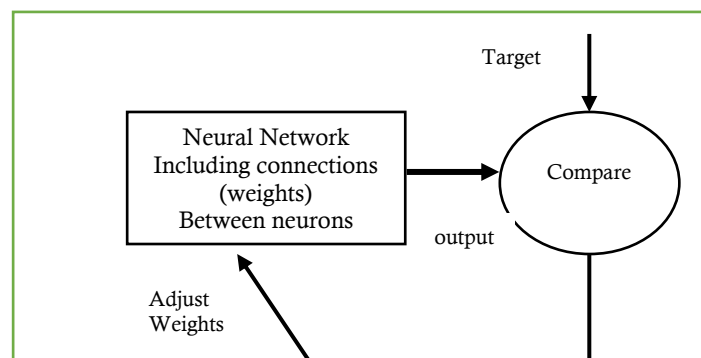
Sebagai alat bantu yang telah diuji dan terbukti, *nftool* telah digunakan oleh banyak peneliti dan praktisi dalam berbagai bidang, seperti ilmu data, teknik, dan biologi (Drouillet et al., 2016). Oleh karena itu, *nftool* menjadi salah satu alat yang paling penting dalam membangun JST dan memecahkan berbagai masalah di bidang yang berbeda (Smith, 2018).

### a) *Neural Network*

*Artificial Neural Network* memiliki 2 jenis metode belajar, pertama metode terbimbing dan metode tidak terbimbing. Bersama dengan metode *Adaline/Madaline*, maka metode *perceptron* sebagai metode pembelajaran terbimbing awal pada jaringan syaraf tiruan, oleh metode *backpropagation* dijadikan sebagai dasar pengembangan lebih lanjut (Bao et al., 2018). *Perceptron* sebagai bentuk sederhana dari JST ditanamkan pola prosedur algoritma pelatihan, yang mencakup *neuron* tunggal dengan *synaptic weights* dan fungsi aktivasi *hard limit* diatur sebagai pemanfaatan metode ini (Dong & Tsai, 2021).

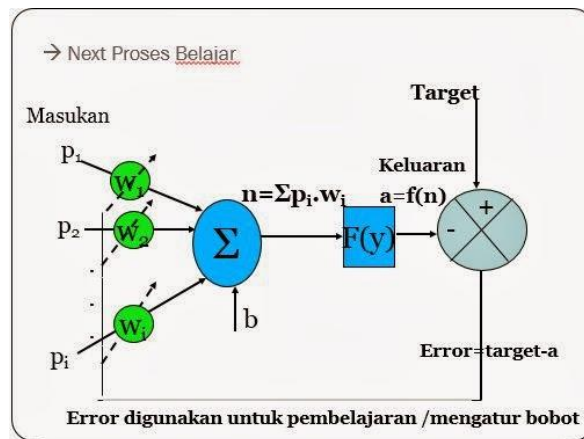
### b) *Perceptron*

Algoritma belajar *perceptron* lebih teruji dibandingkan algoritma *Hebbrule*, yang asumsinya sesuai dengan prosedur belajar iteratifnya, hal ini bisa diuji pada konvergennya dengan bobot yang tepat, yaitu bobot dengan kemungkinan jaringannya untuk mendapat nilai luaran yang sesuai dengan model dan input pembelajaran. Arsitektur dan algoritma *perceptron* mengidentifikasi model dengan algoritma pelatihan terbimbing (He et al., 2022). Perhatikan gambar 1 berikut ini :



**Gambar 1. Arsitektur *Perceptron***  
(sumber : He et al, 2022)

Target sama dengan nilai yang dkehendaki, *Output* sama dengan nilai yang dihasilkan oleh *neuron*, untuk membandingkan antara *output* dengan target maka proses *Compare* dipakai. Jika ditemukan hal berbeda, *weigh* diatur sampai nilai *ouput* sama dengan nilai target atau nilai yang mendekatinya (Azzizi & Zaatri, 2016). Gambar 2 berikut ini bisa membantu memahami proses belajar *perceptron*.



**Gambar 2. Proses Belajar Pada Perceptron**  
(sumber :Azzizi&Zaatri, 2016)

p = input jaringan  
w = bobot  
b = bias

Masukan pada jaringan akan diberikan bobot ( $w$ ) selanjutnya diproses elemen, dimana nilai input akan dikali dengan nilai bobotnya. Pelatihan *perceptron* menggunakan sekumpulan model yang diberikan kepada pelatihan dan dilakukan secara berulang selama proses latihan berlangsung (He et al., 2022). Setiap model yang diberikan adalah pasangan pola input dan pola yang dikehendaki menjadi target. Penjumlahan terhadap tiap masukannya dilakukan oleh *Perceptron* dan perhitungan keluarannya akan menggunakan fungsi ambang (Jakšić et al., 2023). Hasil *output* ini selanjutnya dibandingkan dengan hasil yang dikehendaki, perbedaan yang didapatkan dari perbandingan tersebut dipergunakan untuk mengganti bobot bobot dalam system jaringan. Hal ini dilakukan secara berulang dengan proses yang berlanjut untuk menghasilkan *output* yang tepat dengan hasil yang dikehendaki.

## METODE PENELITIAN

Jaringan saraf tiruan dengan *neural network fitting tool* (nftool) menggunakan metode sebagai berikut:

1. Analisis persiapan data: data pelatihan harus terdiri dari *input* dan *output* yang tepat, dan data validasi harus digunakan untuk mengevaluasi performa JST (Zhou et al., 2019). Perangkat lunak yang dipakai adalah Matlab R2022a dengan fitur *toolbox neural network*, *guide matlab* sebagai *user interface* dan *m-file editor* sebagai fitur *coding*. Data input didasarkan pada 3 variabel yaitu infrastruktur pada nilai 1- 5, dengan tingkat semakin tinggi maka ketersediaan infrastruktur memadai, Skala tingkat akses berada pada rentang 0–10, semakin tinggi nilai indeks maka semakin cepat, dan sebaliknya jika rendah indeks maka akses relatif lambat. variabel trafik 0 untuk tidak efektif dan 1 adalah efektif.
2. Konfigurasi JST: *nftool* juga memungkinkan pengguna untuk menentukan parameter pelatihan seperti *learning rate* dan momentum (Dong & Tsai, 2021). Pada *nftool* dikonfigurasi dengan seleksi *output* dalam matrix, dan sistem pelatihan dibuat sampling 8 *record*, untuk validasi data variable infrastruktur dijarikan sebagai input dan variable trafik sebagai target.
3. Pelatihan JST: Setelah konfigurasi JST dilakukan, pelatihan JST dilakukan dengan menggunakan data pelatihan. *nftool* akan menghasilkan grafik yang menunjukkan bagaimana performa JST berubah selama proses pelatihan (Janiesch et al., 2021). Prosedur pelatihan pada *nftool* memanfaatkan fitur *training*, *validation* dan *testing*. Data pada table diuji dengan

memasukkan pada fitur *input* untuk diproses dengan *hidden layer* dan *output layer*, dengan *number hidden neuron* adalah 8.

4. Evaluasi Performa JST: Setelah JST dilatih, performa JST dievaluasi menggunakan data validasi. *nftool* akan menghasilkan berbagai metrik evaluasi, yaitu dengan nilai 15% pada validasi dengan 1 sampel, testing dengan nilai 15% dengan menggunakan 1 sampel juga. seperti rata-rata kesalahan kuadrat (MSE) dan koefisien determinasi ( $R^2$ ), yang menunjukkan seberapa akurat JST dalam memprediksi hasil yang benar (Huang et al., 2015).
5. Penggunaan JST: Setelah JST dilatih dan diuji, JST dapat digunakan untuk memprediksi nilai keluaran yang belum diketahui dari input yang diberikan (Luo et al., 2019). Setelah proses pelatihan maka hasil pengujian dengan *epoch* interaksi sejumlah 4 dan jumlah data 1000 akan menunjukkan performance dibawah 1.

## Analisa

Dalam keseluruhan metode tersebut, penggunaan *nftool* sangat membantu dalam memudahkan pengguna dalam membangun dan mengevaluasi JST dengan cepat dan mudah (McDonald et al., 2017). Hal ini terlihat dari data yang dimasukkan secara terstruktur pada variabel infrastruktur dengan jumlah sampel 8 dan indeks penilaian 1-5, variabel tingkat akses dengan indeks 0-10 dan variabel trafik dengan 0 dan 1, akan mendukung model pelatihan *algoritma* *progress* yaitu *epoch* dengan 4 iterasi, *performance* < 0.5, *gradient* dengan nilai 1.00-1.00e-10 kemudian dengan *validation check* 0-6 dan berada pada nilai 3, ini memberikan hasil pengujian lebih tepat.

## HASIL DAN PEMBAHASAN

Matlab memiliki fitur jaringan saraf tiruan, dengan *tools* yang cukup memadai yaitu *fitting*, *Pattern recognition* dan *clustering*. Untuk melakukan simulasi dalam melatih data dengan *nftool*, harus dirancang terlebih dahulu data menggunakan program Microsoft excel, kemudian data diletakkan ke *workspace* melalui perintah *xlsread*. Beri penamaan filenya 'inputdata.xlsx'. Misal *field* A Infrastruktur, *field* B tingkat akses dan kolom C trafik pilihannya adalah efektif dan tidak efektif (efektif = 1, tidak efektif = 0).

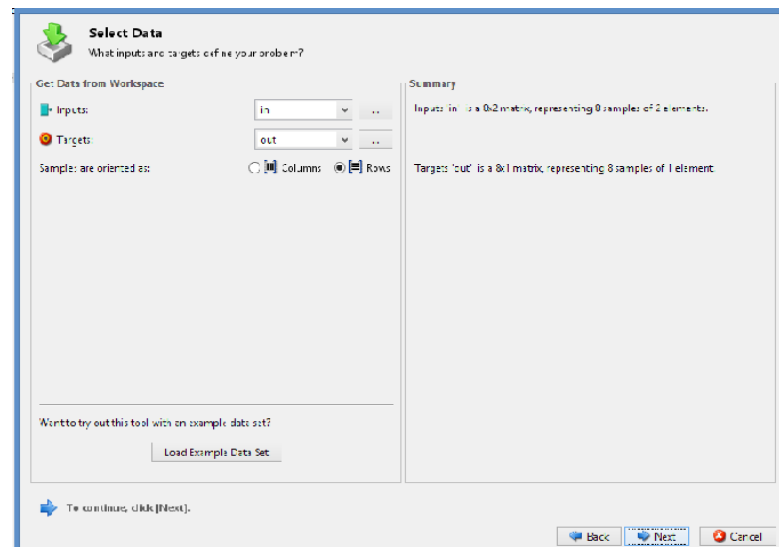
Tabel 1. Variable Melatih Data

| Infrastruktur | Tingkat Akses | Trafik |
|---------------|---------------|--------|
| 5             | 9             | 1      |
| 4             | 8             | 1      |
| 3,5           | 5             | 0      |
| 2             | 4             | 0      |
| 3             | 8             | 1      |
| 5             | 7             | 1      |
| 4             | 6             | 1      |
| 3             | 4             | 0      |

Berikutnya lakukan akses *Command Window*, tulis perintah dan *syntax* berikut ini untuk menciptakan data input dan target yang akan dipakai untuk permodelan dan pelatihan .

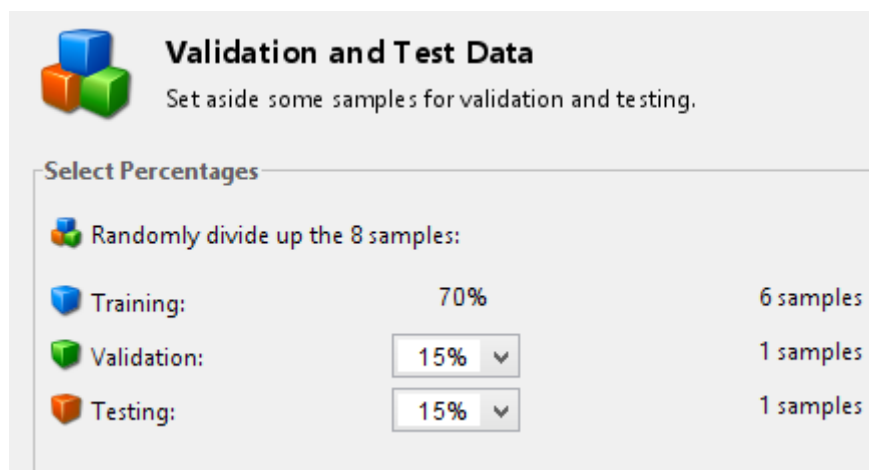
- *input=xlsread('input.xlsx');*
- *in=[input(:,1) input(:,2)];*
- *out=input(:,3);*

Selanjutnya apabila *in* dan *out* telah berada di *workspace*, tampilkan *nftool* untuk melakukan pelatihan dan membuat jaringan saraf tiruan atau JST. Ketik perintah *nftool* di *command window*, lakukan pengisian *input* dan *output* sesuai variabel yang telah ditentukan dan pilih jumlah *record* sesuai dengan kolom atau baris.



Gambar 4. Pemilihan Data Input

Isikan data *input* dan target sesuai dengan variabel yang sudah ditentukan pada tabel di excel. Dalam hal ini variable pada kolom A sebagai input dan variabel C sebagai target dengan *rule based* adalah sebagai berikut



Gambar 5. Data Validasi Variable

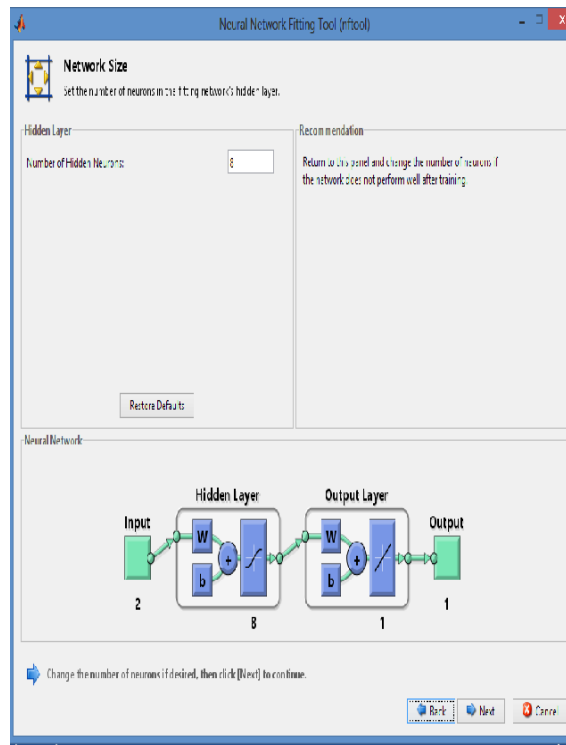
Pada gambar 5 menunjukkan disajikan ke jaringan selama pelatihan , dan jaringan disesuaikan sesuai dengan kesalahannya, pada validasi 15% Ini digunakan untuk mengukur jaringan generalisasi , dan untuk menghentikan pelatihan ketika generalisasi berhenti membaik, sedangkan proses testing 15% Ini tidak berpengaruh pada pelatihan dan memberikan ukuran independen terhadap kinerja jaringan selama dan setelah pelatihan .

Tabel 2. Rule Base Pada JST

| No | Variabel Infrastruktur | Var Perilaku Pengguna | Frekuensi akses |
|----|------------------------|-----------------------|-----------------|
| 1  | 5                      | 9                     | 1               |
| 2  | 4                      | 8                     | 1               |
| 3  | 2                      | 4                     | 0               |
| 4  | 3                      | 8                     | 1               |
| 5  | 5                      | 7                     | 1               |

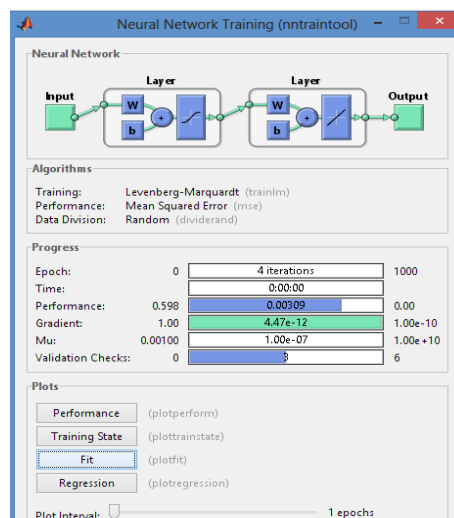
Tabel 2 ini menyatakan variabel infrastruktur dan variabel perilaku menjadi input bagi JST sementara frekuensi akses menjadi target. Pada variabel infrastruktur memiliki *range* 0-5, variabel perilaku *range* 0-10, variabel frekuensi *range* 0-5. Dengan aturan :

- jika variabel infrastruktur tinggi dan variabel perilaku tinggi maka efektif dengan kode "1"
- jika variabel infrastruktur rendah dan variabel perilaku rendah maka tidak efektif dengan kode "0"
- ketentuan berlaku jika  $\geq 5$  maka efektif dan jika  $< 5$  maka tidak efektif.



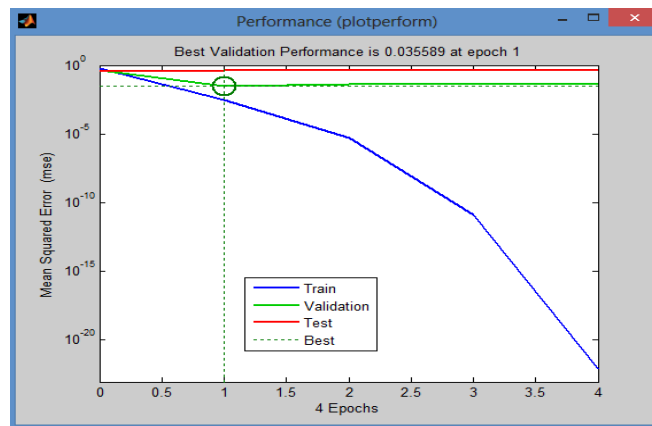
**Gambar 6. Training Dengan Backpropagation**

Gambar 6 diberikan ukuran pengujian sebanyak 8 record sehingga menunjukkan *hidden layer* menjadi 9 dengan target output 1. Melalui pengujian 2 variabel memberikan hasil pengujian yang lebih tepat.



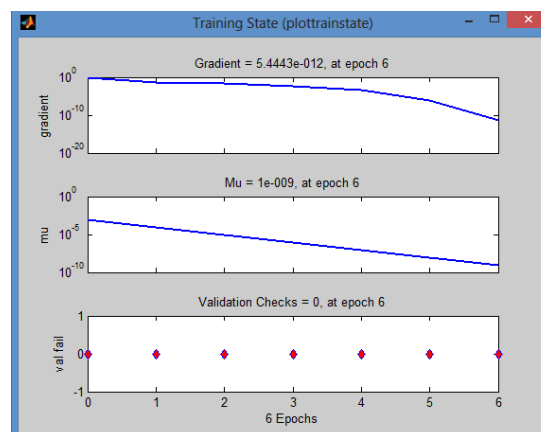
**Gambar 7. Hasil Pengujian 2 Variabel**

Hasil pengujian menunjukkan dengan *epoch* 4 iterasi dan jumlah data dari 1000 maka menunjukkan tingkat performance 0.00309



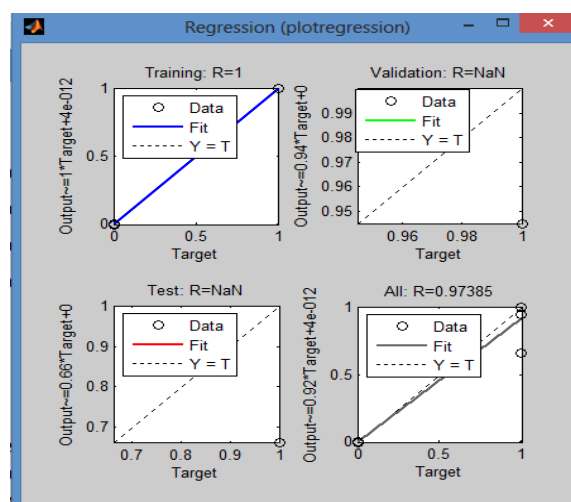
Gambar 8. Plot Performance

Pada gambar 8 grafik *performance* ditunjukkan pada titik terbaik yaitu 0.03559 pada *epoch* 1, pembelajaran dimulai pada  $10^0$  dan menuju pada titik  $<10^{-20}$ .



Gambar 9. Plot Training State

Tampilan pembelajaran bagian dari *gradient* pada *epoch* 6 mencakup Mu dan *validasi check* di gambar 9 bisa dihasilkan garis dan simbol perubahan pada koordinat lebih kecil dari  $10^5$ .



Gambar 10. Plot Regresi



Gambaran dari grafik regresi masing-masing pembelajaran  $R=1$  maka target menghasilkan gambar yang jelas berada pada koordinat 0-1 demikian juga untuk validasi dan keseluruhan nilai  $R$  bergantung pada *ouput* dan target yang dihasilkan gambar 10.

#### Simulasi *Surface*

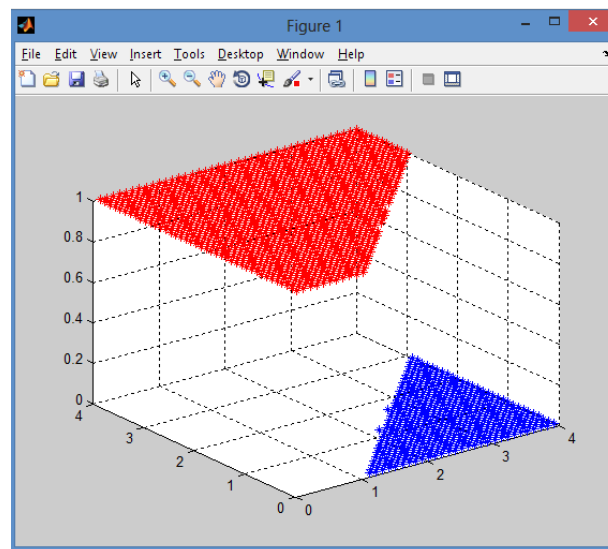
```

grid
plot3(1,1,1)
hold
%Current plot held
grid
for i=1:40
    for j=1:40
        x=i/10;y=j/10;
        k= sim(net,[x;y])
        if k >0.5
            plot3(x,y,k,'r*')
        else
            plot3(x,y,k,'b*')
        end
    end
end
end

```

Gambar 11. Kode *Surface*

Dalam *syntax* program di gambar 11, fungsi *plot* memberikan identifikasi untuk  $r$  adalah *red* sedangkan  $b$  adalah *blue*, serta menunjukkan pola dari *sim (net, koordinat x dan y*. Warna merah ditunjukkan untuk kode efektif sedang kode warna biru untuk tidak efektif



Gambar 12. *Plot Surface*

Area yang berwarna merah dengan acuan kode diatas maka terlihat sebagaimana dalam gambar 12, termasuk kategori efektif “1” sedangkan area warna biru termasuk kategori tidak efektif “0”.

## KESIMPULAN DAN SARAN

### Kesimpulan

Pada simulasi dengan variable pembanding dengan target pada *range* yang ditentukan pada *rule based target* dengan menggunakan *toolbox* Matlab, menunjukkan bentuk dari jaringan syaraf tiruan menggunakan metode *fitting* menunjukkan grafik *best validaton performance* pada titik akurat,



namun juga akan terjadi perubahan pada tiap pembelajaran dengan menunjukkan iterasi yang berubah-ubah. Pada perubahan nilai *epoch* dan iterasi akan menghasilkan *best validasi* yang berbeda namun tidak terlalu jauh rentang perbedaannya. Pada tampilan regresi dan nilai R secara keseluruhan akan menempati posisi optimasi.

### Saran

Berdasarkan kesimpulan penelitian ini maka dimasa datang perlu tantangan dalam bidang data yang kompleks dan dengan jumlah data yang besar, sebagai rekomendasi adalah data hasil kumpulan sensor pada data besar. Gunakan data yang telah diseleksi untuk melatih jaringan yang sudah ditentukan. *Neural Network Fitting Tool* akan melakukan proses pelatihan dan menampilkan statistik hasil pelatihan.

### UCAPAN TERIMA KASIH

Terimakasih kepada keluarga yang sudah mendukung kegiatan penelitian ini, dan juga para rekan kerja di Fakultas Teknik Universitas 17 Agustus 1945 Banyuwangi.

### REFERENSI

- Azzizi, N., & Zaatri, A. (2016). A learning process of multilayer Perceptron for speech recognition. *International Journal of Pure and Applied Mathematics*, 107(4). <https://doi.org/10.12732/ijpam.v107i4.18>
- Bao, C., Pu, Y., & Zhang, Y. (2018). Fractional-Order Deep Backpropagation Neural Network. *Computational Intelligence and Neuroscience*, 2018. <https://doi.org/10.1155/2018/7361628>
- Dong, H., & Tsai, S. B. (2021). An Empirical Study on Application of Machine Learning and Neural Network in English Learning. *Mathematical Problems in Engineering*, 2021. <https://doi.org/10.1155/2021/8444858>
- Drouillet, C., Karandikar, J., Nath, C., Journeaux, A. C., El Mansori, M., & Kurfess, T. (2016). Tool life predictions in milling using spindle power with the neural network technique. *Journal of Manufacturing Processes*, 22. <https://doi.org/10.1016/j.jmapro.2016.03.010>
- Du, M., Wang, P., Wang, J., Cheng, Z., & Wang, S. (2019). Intelligent Turning Tool Monitoring with Neural Network Adaptive Learning. *Complexity*, 2019. <https://doi.org/10.1155/2019/8431784>
- He, H., Huang, G., Zhang, B., & Zheng, Z. (2022). Research on DoS Traffic Detection Model Based on Random Forest and Multilayer Perceptron. *Security and Communication Networks*, 2022. <https://doi.org/10.1155/2022/2076987>
- Huang, G., Huang, G. Bin, Song, S., & You, K. (2015). Trends in extreme learning machines: A review. In *Neural Networks* (Vol. 61). <https://doi.org/10.1016/j.neunet.2014.10.001>
- Jakšić, O., Jakšić, Z., Guha, K., Silva, A. G., & Laskar, N. M. (2023). Comparing artificial neural network algorithms for prediction of higher heating value for different types of biomass. *Soft Computing*, 27(9). <https://doi.org/10.1007/s00500-022-07641-4>
- Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3). <https://doi.org/10.1007/s12525-021-00475-2>
- Luo, C., Tan, C., Wang, X., & Zheng, Y. (2019). An evolving recurrent interval type-2 intuitionistic fuzzy neural network for online learning and time series prediction. *Applied Soft Computing Journal*, 78. <https://doi.org/10.1016/j.asoc.2019.02.032>
- McDonald, B. W., Hall, L., & Zhang, X. P. (2017). Who is most likely to offend in my store now? Statistical steps towards retail crime prevention with Auror. *ANZIAM Journal*, 57. <https://doi.org/10.21914/anziamj.v57i0.10507>

- Prasetyo, Y. P. W. (2021). Image Compression Using Method On Dct (Discrete Cosine Transform) Low Resolution Camera. *International Journal of Education and Social Science Research*, 04(03). <https://doi.org/10.37500/ijessr.2021.4302>
- Smith, L. N. (2018). Disciplined Approach To Neural Network. *ArXiv:1803.09820v2 [Cs.LG]*.
- Zhou, T., Ruan, S., & Canu, S. (2019). A review: Deep learning for medical image segmentation using multi-modality fusion. *Array*, 3–4. <https://doi.org/10.1016/j.array.2019.100004>