

ANALISIS PRODUKTIVITAS PEKERJAAN *PREFABRICATED VERTICAL DRAIN* MENGGUNAKAN JARINGAN SYARAF TIRUAN

Eka Nur Fitriani¹⁾, Budi Susetyo²⁾

¹⁾Program Studi Teknik Sipil, Institut Sains Teknologi Al-Khamal

²⁾Program Studi Magister Teknik Sipil, Fakultas Teknik, Universitas Mercu Buana Jakarta

¹⁾email: eka.010410@gmail.com, ²⁾email: budisusetyo@mercubuana.ac.id

Abstract

Prefabricated vertical drain is a technique of soil improvement with the concepts of release pore water content in the soil so as to accelerate settlement time of the soil. Currently, prefabricated vertical drain method is many used for construction of building at soft soil. During this time, prefabricated vertical drain work installation time is determined based on prefabricated vertical drain work experience in previous projects. Currently, artificial neural networks have been used in engineering and construction management for prediction, risk analysis, decision making, resource optimization, classification, and selection. For calculation of productivity of prefabricated vertical drain work by using artificial neural network (ANN) used backpropagation algorithm model. The backpropagation architecture consists of an input screen, a hidden screen, and an output screen. Backpropagation training consists of three phases: advanced phase, reverse phase, and weight modification. Artificial neural network (ANN) training backpropagation using matlab software. From the analysis results obtained that artificial neural network (ANN) backpropagation can be used on calculation of productivity of prefabricated vertical drain work.

Key words: Artificial neural network (ANN), backpropagation, prefabricated vertical drain, productivity

Abstrak

*Prefabricated vertical drain merupakan teknik perbaikan tanah dengan prinsip mengeluarkan kadar air pori di dalam tanah sehingga mempercepat waktu penurunan tanah. Saat ini metode *prefabricated vertical drain* banyak digunakan untuk pembangunan khususnya yang berdiri di atas tanah lunak. Selama ini, waktu pekerjaan *prefabricated vertical drain* ditentukan berdasarkan pengalaman pekerjaan *prefabricated vertical drain* di proyek-proyek terdahulu. Saat ini, jaringan syaraf tiruan telah digunakan dalam teknik dan manajemen konstruksi untuk prediksi, analisis risiko, pengambilan keputusan, optimasi sumber daya, klasifikasi, dan seleksi. Untuk perhitungan produktivitas pekerjaan *prefabricated vertical drain* dengan menggunakan jaringan syaraf tiruan (JST) digunakan model algoritma *backpropagation*. Arsitektur *backpropagation* terdiri dari layar masukan, layar tersembunyi, dan layar keluaran. Pelatihan *backpropagation* terdiri dari tiga fase yaitu fase maju, fase mundur, dan modifikasi bobot. Pelatihan jaringan syaraf tiruan (JST) *backpropagation* menggunakan software matlab. Dari hasil analisis didapat bahwa jaringan syaraf tiruan (JST) *backpropagation* dapat digunakan pada perhitungan produktivitas pekerjaan *prefabricated vertical drain*.*

Kata kunci: Jaringan syaraf tiruan (JST), *backpropagation*, *prefabricated vertical drain*, produktivitas.

1. PENDAHULUAN

Saat ini metode *prefabricated vertical drain* banyak digunakan untuk pembangunan khususnya yang berdiri di atas tanah lunak, karena jika suatu bangunan berdiri diatas tanah lunak, biasanya ada dua masalah geoteknik yang harus dipecahkan yaitu daya dukung yang rendah dan penurunan serta beda penurunan yang besar (Aspar & Fitriani, 2016). Pemasangan *prefabricated vertical drain* menggunakan alat berat mesin hidrolik sebagai alat untuk memancang material *prefabricated vertical drain* ke dalam tanah.

Terdapat 2 bagian utama dalam alat pancang *prefabricated vertical drain* yaitu *rig* dan *mandrel*.

Pekerjaan *prefabricated vertical drain* mulai banyak dilakukan untuk perbaikan tanah dasar pada proyek-proyek reklamasi, jalan tol, PLTU, dan bangunan lainnya yang berdiri diatas tanah lunak. Akan Tetapi, penelitian mengenai tingkat produktivitas pekerjaan tersebut belum banyak dilakukan. Selama ini waktu pekerjaan *prefabricated vertical drain* ditentukan berdasarkan pengalaman pekerjaan

prefabricated vertical drain di proyek-proyek terdahulu.

Saat ini, banyak model pendekatan yang digunakan untuk menghitung produktivitas diantaranya model statistik, *generalized linear mixed models*, model jaringan syaraf tiruan. Model statistik digunakan dalam perhitungan produktivitas pada pekerjaan pengeboran pondasi bor pile (Dhiva, 2014). *Generalized Linear Mixed Model* (GLMM) memiliki potensi untuk meningkatkan performansi model estimasi produktivitas bulldozer (Rashidi dkk., 2014) Pendekatan model jaringan syaraf tiruan digunakan pada penentuan faktor-faktor yang mempengaruhi produktivitas pekerja (Heravi dan Eslamdoost, 2015; Al-Zwainy dkk., 2012); penentuan penggunaan bulldozer yang optimal pada pekerjaan tanah (Rashidi dkk., 2014); perhitungan produktivitas alat (Warsito dkk., 2016; Ok dan Sinha, 2006).

2. LANDASAN TEORI

Produktivitas

Secara umum produktivitas diartikan sebagai suatu perbandingan antara hasil keluaran dan masuk atau *output: input*. Masukan sering dibatasi dengan masukan tenaga kerja, material, sedangkan keluaran diukur dalam satuan bentuk fisik atau nilai.

Besar kecilnya produktivitas yang dihasilkan suatu pekerjaan pada proyek konstruksi dipengaruhi oleh banyak faktor, diantaranya adalah:

1. sumber daya;
2. kondisi lingkungan kerja;
3. metode kerja.

Estimasi produktivitas didasarkan pada asumsi bahwa adanya hubungan antara beberapa faktor dan produktivitas suatu pekerjaan pada masa lalu. Untuk itu nilai produktivitas suatu pekerjaan dapat diperkirakan dengan menentukan hubungan antara hubungan tersebut dan menentukan nilai faktor-faktor yang berpengaruh. Terdapat 2 pendekatan yang dapat digunakan dalam memperkirakan produktivitas yaitu model statistik dan model jaringan syaraf tiruan (JST).

Model jaringan syaraf tiruan (JST) saat ini telah banyak digunakan pada penelitian konstruksi khususnya untuk memprediksi

mengenai produktivitas. Teknik jaringan syaraf tiruan meniru proses pengenalan pola otak manusia (Wasserman, 1989).

Kelebihan jaringan syaraf di atas teknik statistik adalah kemampuan untuk menangkap hubungan non-linear multivariabel yang kompleks, dan keterkaitan antar faktor-faktor yang berpengaruh.

Pekerjaan Prefabricated Vertical Drain

Pemasangan *prefabricated vertical drain* yaitu dengan menggunakan alat berat mesin hidrolik sebagai alat untuk memancang material *prefabricated vertical drain* ke dalam tanah. Terdapat 2 bagian utama dalam alat pancang *prefabricated vertical drain* yaitu *rig* dan *mandrel*.

Pemilihan *rig* instalasi *prefabricated vertical drain* sangat penting untuk pelaksanaan proyek *prefabricated vertical drain* dan didasarkan terutama pada faktor-faktor berikut:

1. kekuatan tanah *platform* instalasi *prefabricated vertical drain*;
2. kedalaman instalasi;
3. tipe tanah;
4. kapasitas produksi *rig*

Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) adalah sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi (Agustin & Prahasto, 2012). Jaringan syaraf tiruan (JST) dibentuk sebagai generalisasi model matematika dari jaringan-jaringan syaraf biologi, dengan asumsi:

1. Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron)
2. Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung
3. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal
4. Untuk menentukan *output*, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linear) yang dikenakan pada jumlahan *input* yang diterima. Besarnya *output* ini selanjutnya dibandingkan dengan suatu ambang batas (*threshold*).

Jaringan syaraf tiruan (JST) ditentukan oleh 3 (tiga) hal yaitu (Siang, 2005):

1. Pola hubungan antar neuron (disebut arsitektur jaringan)
2. Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*)
3. Fungsi aktivasi

Beberapa arsitektur jaringan yang sering digunakan dalam jaringan syaraf tiruan antara lain :

1. Jaringan Layar Tunggal (*single layer network*). Dalam jaringan ini, sekumpulan *input* neuron dihubungkan langsung dengan sekumpulan *output*-nya.
2. Jaringan Layar Jamak (*multi layer network*). Jaringan layar jamak merupakan perluasan dari layar tunggal. Dalam jaringan ini, selain unit *input* dan *output*, ada unit-unit lain (sering disebut layar tersembunyi). Dimungkinkan pula ada beberapa layar tersembunyi. Sama seperti pada unit *input* dan *output*, unit-unit dalam satu layar tidak saling berhubungan.
3. Jaringan *Reccurent*. Model jaringan *reccurent* mirip dengan jaringan layar tunggal ataupun ganda. Hanya saja, ada neuron *output* yang memberikan sinyal pada unit *input* (sering disebut *feedback loop*).

Dalam jaringan syaraf tiruan, fungsi aktivasi digunakan untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah net masukan (kombinasi linear masukan dan bobotnya).

Jika $net = \sum x_i w_i$, maka fungsi aktivasinya adalah $f(net) = f(\sum x_i w_i)$.

Beberapa fungsi aktivasi yang sering digunakan adalah sebagai berikut :

1. Fungsi *threshold* (ambang batas)

$$f(x) = \begin{cases} 1 & \text{jika } x \geq \alpha \\ 0 & \text{jika } x < \alpha \end{cases} \quad (1)$$

Untuk beberapa kasus fungsi *threshold* yang dibuat tidak bernilai 0 atau 1, tapi berharga -1 atau 1 (sering disebut *threshold bipolar*). Jadi :

$$f(x) = \begin{cases} 1 & \text{jika } x \geq \alpha \\ -1 & \text{jika } x < \alpha \end{cases} \quad (2)$$

2. Fungsi sigmoid

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

Fungsi sigmoid sering digunakan karena nilai fungsinya yang terletak antara 0 dan 1 serta dapat diturunkan dengan mudah.

$$f'(x) = f(x)(1 - f(x)) \quad (4)$$

3. Fungsi identitas

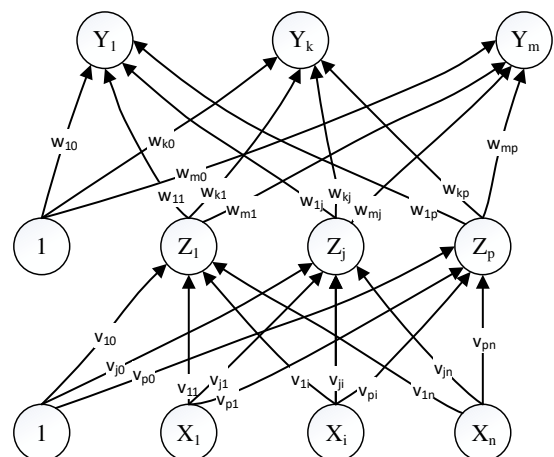
$$f(x) = x$$

fungsi identitas sering digunakan apabila menginginkan keluaran jaringan berupa sembarang bilangan riil (bukan hanya pada range $[0,1]$ atau $[-1,1]$).

Backpropagation Model

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih dari layar tersembunyi. Gambar 1 adalah arsitektur *backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah layar tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran.

v_{ij} merupakan bobot garis dari unit masukan x_i ke unit layar tersembunyi z_j (v_{j0} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi z_j). w_{kj} merupakan bobot dari unit layar tersembunyi z_j ke unit keluaran y_k (w_{k0} merupakan bobot dari bias di layar tersembunyi ke unit keluaran z_k).



Gambar 1 Arsitektur *Backpropagation*

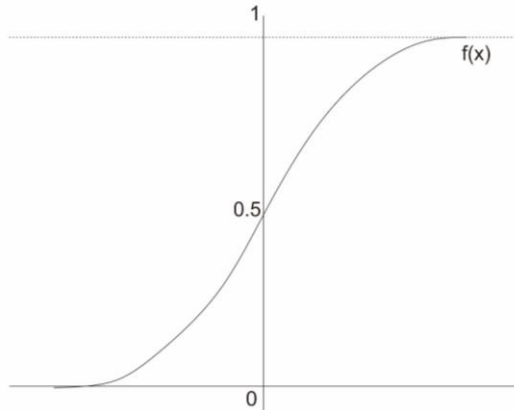
Dalam *backpropagation*, fungsi aktivasi yang digunakan harus memenuhi beberapa syarat yaitu : kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun, salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering digunakan adalah fungsi sigmoid biner yang memiliki range (0,1).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

dengan turunan

$$f'(x) = f(x)(1 - f(x)) \quad (6)$$

Grafik fungsinya dapat dilihat pada gambar 2.



Gambar 2 Grafik Fungsi Sigmoid Biner

Fungsi lain yang sering digunakan adalah fungsi sigmoid bipolar yang bentuk fungsinya mirip dengan fungsi sigmoid biner, tapi dengan range $(-1,1)$.

$$f(x) = \frac{2}{1 + e^{-x}} - 1 \quad (7)$$

dengan turunan

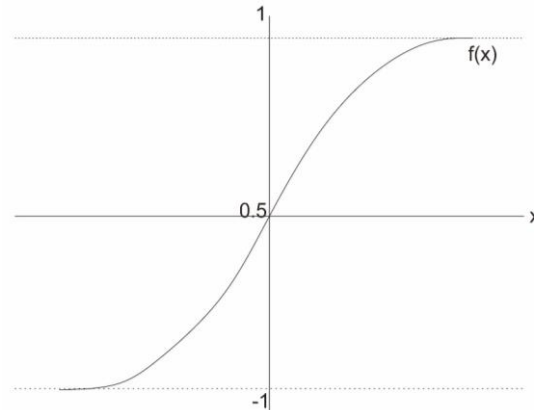
$$f'(x) = \frac{(1 + f(x))(1 - f(x))}{2} \quad (8)$$

Grafik fungsinya tampak pada gambar 3

Fungsi sigmoid memiliki nilai maksimum = 1. Maka untuk pola yang targetnya > 1 , pola masukan dan keluaran harus lebih dahulu ditransformasi sehingga semua polanya memiliki *range* yang sama seperti fungsi sigmoid yang digunakan. Alternatif lain adalah menggunakan fungsi aktivasi sigmoid hanya pada layar yang bukan layar keluaran. Pada layar keluaran, fungsi aktivasi yang digunakan adalah fungsi identitas : $f(x) = x$

Pelatihan *backpropagation* memiliki 3 fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan

tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.



Gambar 3 Grafik Fungsi Sigmoid Bipolar

Algoritma pelatihan untuk jaringan *backpropagation* (dengan aktivasi sigmoid biner dan fungsi identitas pada keluaran) adalah sebagai berikut :

A. Kalkulasi masukan normal untuk unit x_1 sampai dengan x_9 . Masukan normal didapat dengan melakukan percobaan terus menerus sampai dicapai keadaan eror yang paling minimum.

B. Fase I Propagasi Maju

1. Hitung semua keluaran di unit tersembunyi layar 1

$$H_net_m = b_{hm} + \sum_{i=1}^i x_i w_{inm} \quad (9)$$

$$H_m = f(H_net_m) = \frac{1}{1 + e^{-H_net_m}} \quad (10)$$

dimana,

H_net_m = nilai masukan pada setiap unit tersembunyi layar 1

H_m = nilai keluaran pada setiap unit tersembunyi layar 1

b_{hm} = nilai bobot bias dari untuk unit tersembunyi layar 1

x_1 = nilai *input* unit masukan

w_{inm} = nilai bobot dari unit masukan ke unit tersembunyi layar 1

2. Hitung semua keluaran di unit tersembunyi layar 2

$$H_net_n = b_{hn} + \sum_{h=1}^h H_m w_{hnm} \quad (11)$$

$$H_n = f(H_{net_n}) = \frac{1}{1 + e^{-H_{net_n}}} \quad (12)$$

dimana,

H_{net_n} = nilai masukkan pada setiap unit tersembunyi layar 2

H_n = nilai keluaran pada setiap unit tersembunyi layar 2

b_{hm} = nilai bobot bias dari untuk unit tersembunyi layar 2

H_m = nilai *input* unit tersembunyi layar 1

w_{hnm} = nilai bobot dari unit tersembunyi layar 1 ke unit tersembunyi layar 2

3. Hitung keluaran pada output Y

$$Y_{net} = b_o + \sum_{o=1}^o H_n w_{om} \quad (13)$$

$$Y = f(Y_{net}) = Y_{net} \quad (14)$$

dimana,

Y_{net} = nilai masukkan pada unit keluaran

Y = nilai keluaran pada unit keluaran

b_o = nilai bobot bias dari untuk keluaran

H_n = nilai *input* unit tersembunyi layar 2

w_{om} = nilai bobot dari unit tersembunyi layar 2 ke unit keluaran

C. Fase II Propagasi Mundur

4. Hitung faktor δ unit keluaran berdasarkan kesalahan pada unit keluaran Y

$$\delta = (t - Y)Y_{net} \quad (15)$$

dimana ,

δ = faktor kesalahan unit keluaran

t = target yang harus dicapai

Y = nilai pada unit keluaran

Y_{net} = nilai masukkan pada unit keluaran

Hitung suku perubahan bobot w_{om} (yang akan digunakan untuk merubah bobot w_{om})

$$\Delta w_{om} = \alpha \delta H_n \quad (16)$$

dimana,

Δw_{om} = suku perubahan nilai bobot dari unit tersembunyi layar 2 ke unit keluaran

α = laju percepatan

H_n = nilai unit tersembunyi layar 2

δ = faktor kesalahan unit keluaran

5. Hitung faktor δ unit tersembunyi layar 2 berdasarkan kesalahan disetiap unit tersembunyi layar 2 H_n

$$\delta_{net_n} = \sum_{o=1}^o \delta w_{om} \quad (17)$$

$$\delta_n = \delta_{net_n} f'(H_{net_n}) = \delta_{net_n} H_n (1 - H_n) \quad (18)$$

$$\Delta w_{hnm} = \alpha \delta_n H_m \quad (19)$$

dimana,

δ_{net_n} = fungsi faktor kesalahan nilai masukkan unit tersembunyi layar 2

δ = faktor kesalahan unit keluaran

δ_n = faktor kesalahan unit tersembunyi layar 2

H_n = nilai input unit tersembunyi layar 2

w_{om} = nilai bobot dari unit tersembunyi layar 2 ke unit keluaran

Δw_{om} = suku perubahan nilai bobot dari unit tersembunyi layar 1 ke unit tersembunyi layar 2

H_m = nilai *input* unit tersembunyi layar 1

w_{hnm} = nilai bobot dari unit tersembunyi layar 1 ke unit tersembunyi layar 2

6. Hitung faktor δ unit tersembunyi layar 1 berdasarkan kesalahan disetiap unit tersembunyi layar 1 H_m

$$\delta_{net_m} = \sum_{o=1}^o \delta w_{hnm} \quad (20)$$

$$\delta_m = \delta_{net_m} f'(H_{net_m}) = \delta_{net_m} H_m (1 - H_m) \quad (21)$$

$$\Delta w_{inn} = \alpha \delta_m x_i \quad (22)$$

dimana,

δ_{net_m} = fungsi faktor kesalahan nilai masukkan unit tersembunyi layar 1

δ = faktor kesalahan unit keluaran

δ_m = faktor kesalahan unit tersembunyi layar 1

H_m = nilai *input* unit tersembunyi layar 1

w_{hnm} = nilai bobot dari unit tersembunyi layar 1 ke unit tersembunyi layar 2

Δw_{inn} = suku perubahan nilai bobot dari unit masukan ke unit tersembunyi layar 1

x_1 = nilai *input* unit masukan

w_{inn} = nilai bobot dari unit masukan ke unit tersembunyi layar 1

D. Fase III Perubahan Bobot

7. Hitung semua perubahan bobot

Perubahan bobot garis yang menuju unit keluaran :

$$w_{om}(\text{baru}) = w_{om}(\text{lama}) + \Delta w_{om}$$

Perubahan bobot garis yang menuju unit tersembunyi layar 2 :

$$w_{hnm}(\text{baru}) = w_{hnm}(\text{lama}) + \Delta w_{hnm}$$

Perubahan bobot garis yang menuju unit tersembunyi layar 1 :

$$w_{inn}(\text{baru}) = w_{inn}(\text{lama}) + \Delta w_{inn} \quad (23)$$

3. METODE

Untuk menganalisis jaringan syaraf tiruan dengan model *backpropagation* dilakukan dengan bantuan perangkat lunak Matlab R2016b.

Adapun tahapan-tahapan dalam penelitian ini yaitu:

1. pengumpulan data, data yang dikumpulkan merupakan data produktivitas total pekerjaan *prefabricated vertical drain*;
2. analisa data;
3. analisa sistem jaringan syaraf tiruan dengan menggunakan model *backpropagation*;
4. membangun sistem pada *software* Matlab

dengan desain GUIDe;

5. implementasikan sistem;
6. melatih data yang ada dengan sistem yang telah dibangun dengan desain GUIDe hingga didapat MSE mendekati nol dan R^2 mendekati 1;
7. menguji data yang ada dengan sistem yang telah dibangun;
8. mengevaluasi sistem jaringan syaraf tiruan.

Variabel dan Arsitektur Jaringan Syaraf Tiruan

Variabel terikat pada penelitian ini yaitu produktivitas pekerjaan *prefabricated vertical drain*, sedangkan untuk variabel bebas dikelompokkan kedalam 3 kelompok yaitu kondisi mesin, kondisi tanah dan lokasi pemasangan, dan kondisi pekerja. Ketiga variabel bebas tersebut diuraikan dan dapat dilihat pada tabel 1 dibawah ini.

Untuk pemodelan jaringan syaraf tiruannya dapat dilihat pada gambar 4 di bawah ini.

Tabel 1 Variabel Penelitian

Variabel	Kelompok	Uraian
x1	Kondisi Mesin	Jenis mesin dasar
x2		Daya penetrasi
x3		Ketinggian rig
x4	Kondisi perencanaan dan lokasi pemasangan	Jarak PVD
x5		Pola PVD
x6		Cuaca
x7	Kondisi pekerja	Pengalaman dan keterampilan pekerja
x8		Jumlah tenaga kerja pembantu
x9		Sikap pekerja

4. HASIL DAN DISKUSI

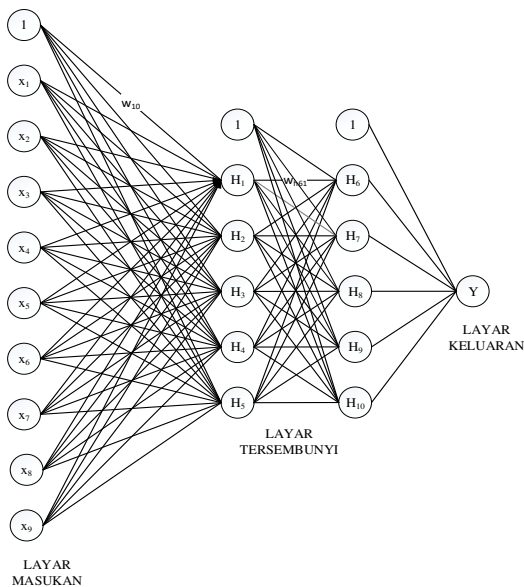
Perancangan Sistem

Untuk memprediksi produktivitas pekerjaan *prefabricated vertical drain* dalam penelitian ini dirancang suatu program khusus yang dibangun dengan menggunakan system *toolbox* yang telah disediakan oleh MATLAB dengan menggunakan *Graphical User Interface Guide* (GUIDe).

Untuk melakukan pelatihan jaringan syaraf tiruan yang telah dibuat sebelumnya dengan program yang telah dibuat pada

toolbox yang ada pada MATLAB, maka diperlukan parameter sebagai berikut :

Net size : *Input Layer* : 10 sel
Hidden Layer : 100 sel dan 1 sel
Output Layer : 1 sel
Maksimum epoch / iterasi : 50000
Error goals : 0,000001
Learning rate : 0,1
Training function : *trainlm*
Fungsi aktivasi *hidden layer* : sigmoid biner
Fungsi aktivasi *output layer* : linear
Bobot -1 sampai dengan 1



Gambar 4 Pemodelan Jaringan Syaraf Tiruan

Arsitektur Jaringan Syaraf Tiruan

Gambar 5 menunjukkan bahwa arsitektur jaringan yang sudah dibuat dalam bentuk GUIDE, terdiri dari satu lapisan *input* yang terdiri dari 9 (sembilan) unit sel dan 1 (satu) unit bias, 2(dua) lapisan tersembunyi dengan jumlah unit sel yang ditentukan yaitu sebanyak 100 dan 1, dan satu lapisan output berjumlah satu unit sel sebagai target.

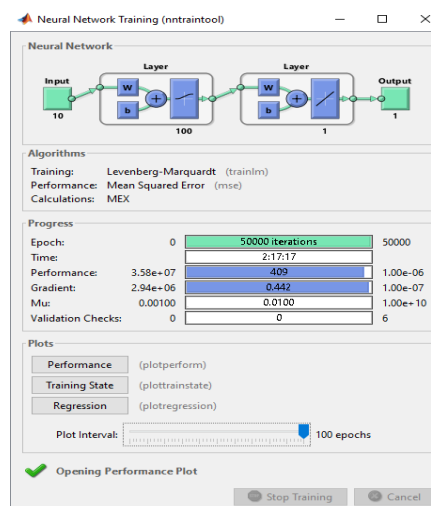
Prediksi Produktivitas dengan menggunakan Jaringan Syaraf Tiruan

Data yang ada di masukkan ke dalam program yang sebelumnya di rancang (Gambar 5) dengan bobot yang dipilih yaitu random untuk mencari nilai bobot yang dapat membuat jaringan syaraf tiruan menjadi lebih akurat.

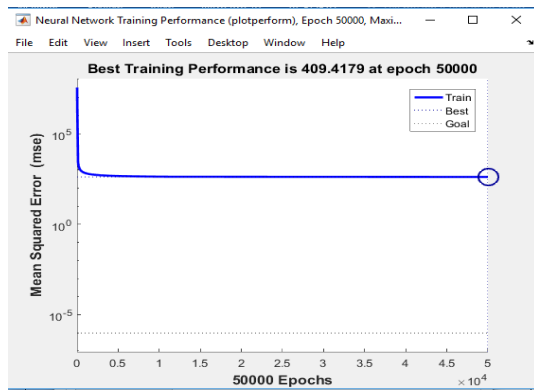
Data yang dilatih akan menghasilkan *neural network training* yang dapat dilihat pada Gambar 6 sampai dengan Gambar 8.

Pada Gambar 6 menunjukkan proses pembelajaran pada setiap epoch. Pada proses ini, iterasi dihentikan pada epoch ke-50000, karena batas dari epoch yang diinginkan sudah tercapai dan ($MSE = 409,4 > 0.000001$) dimana MSE ini merupakan MSE yang muncul ketika pelatihan selesai dilakukan sesuai dengan iterasi yang ditentukan. Gambar 8 menunjukkan hubungan antara target dengan output jaringan pada data pelatihan. Dari pengujian pada data pelatihan untuk kecocokan antara output jaringan dengan target diperoleh koefisien korelasi (R) bernilai $0,99993 \approx 1$ dimana untuk hasil terbaik adalah bernilai 1.

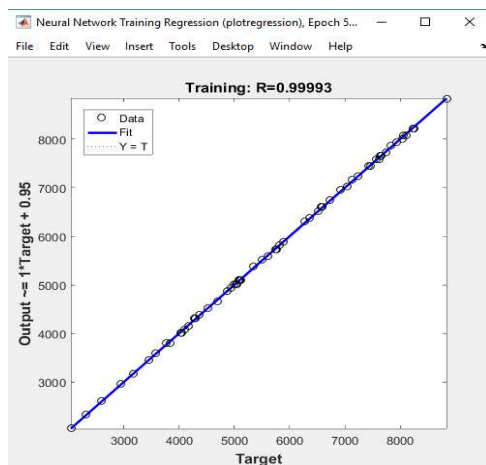
Gambar 5 Tampilan Aplikasi Pelatihan JST dalam GUIDE



Gambar 6 Arsitektur Jaringan Syaraf Tiruan pada Program Matlab



Gambar 7 Hasil *Neural Network* Berupa *Plot Perform*



Gambar 8 Hasil *Neural Network* Berupa *Plot Regression*

Hasil dari produktivitas target pada jaringan di atas di masukkan ke dalam tabel dan dicari nilai *mean square error* (MSE).

Tabel 2 Data Hasil Keluaran

Produktivitas Aktual m	Produktivitas Target m	Error
2320	2329	-9
4516	4517	-1
3176	3166	10
4048	4017	31
6585	6597	-12
3456	3445	11
5024	5018	6
4368	4373	-5
2592	2607	-15
4105	4089	16
5764	5737	27
8227	8217	10
8112	8079	33
4096	4089	7
5744	5737	7
8064	8079	-15
4864	4876	-12

[Lanjutan Tabel 2]

Produktivitas Aktual m	Produktivitas Target m	Error
3568	3589	-21
5104	5090	14
6576	6597	-21
2944	2954	-10
5348	5378	-30
5888	5881	7
3840	3803	37
4032	4017	15
6928	6953	-25
6732	6740	-8
4276	4301	-25
6524	6525	-1
5501	5522	-21
5763	5737	26
5049	5018	31
5814	5809	5
7463	7449	14
5117	5090	27
8840	8836	4
8840	8836	4
4692	4660	32
8245	8217	28
7752	7730	22
7242	7236	6
7650	7659	-9
4284	4301	-17
7123	7165	-42
7633	7659	-26
7565	7589	-24
4522	4516	6
6358	6383	-25
6273	6311	-38
7939	7939	0
7837	7869	-32
2040	2051	-11
4947	4946	1
3774	3803	-29
7038	7024	14
5083	5090	-7
7429	7449	-20
7616	7589	27
8041	8009	32
5610	5594	16
4998	5018	-20
4165	4160	5

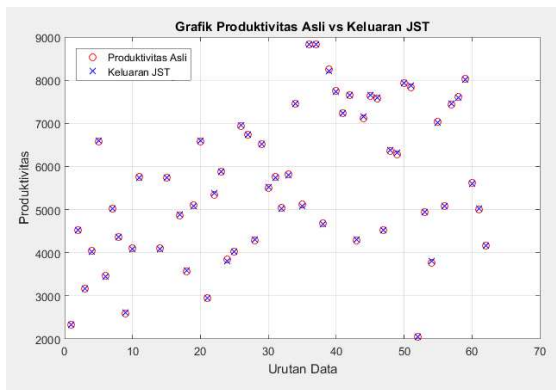
Dari pelatihan jaringan yang dilakukan, didapatkan nilai MSE (Mean Square Error) yaitu:

$$MSE = \frac{\sum_{i=1}^{62} e^2}{n} = \frac{25316}{62} = 408,3$$

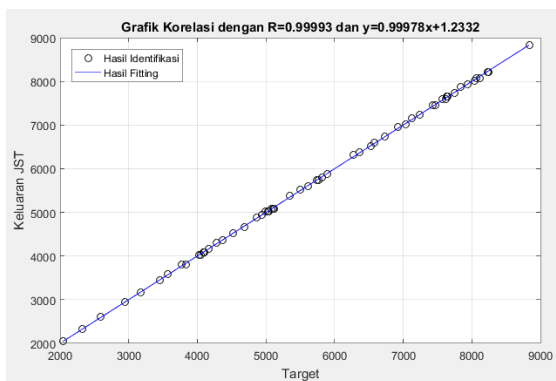
Hal ini berarti bahwa jaringan sudah mampu belajar dengan baik untuk menghasilkan target yang diinginkan.

Prediksi Produktivitas dengan Model Jaringan Syaraf Tiruan

Data yang digunakan pada tahapan pengujian jaringan syaraf tiruan sama dengan data yang digunakan pada pelatihan tujuannya untuk mengetahui perbandingan dan korelasi antara produktivitas asli dan target.



Gambar 9 Grafik Produktivitas Asli VS Keluaran JST



Gambar 10 Grafik Korelasi Produktivitas Asli VS Keluaran JST

Gambar 9 dan Gambar 10 merupakan hasil dari keluaran data pelatihan yang menunjukkan perbandingan antara target dengan *output* jaringan pada data pelatihan. Pada gambar dapat dilihat bahwa sebagian besar *output* jaringan (o) dan target (x) sudah berdekatan (hampir menempati posisi yang sama) hal ini dikarenakan tingkat error yang tidak signifikan.

5. KESIMPULAN

Dari hasil pembahasan dan pengujian dengan menggunakan Software Matlab, maka dapat ditarik kesimpulan sebagai berikut:

- 1) Jaringan yang digunakan terdiri dari tiga buah lapisan yaitu lapisan *input*, lapisan tersembunyi dan lapisan keluaran.
- 2) Algoritma *backpropagation* dapat melakukan proses prediksi pada perhitungan nilai produktivitas pekerjaan *prefabricated vertical drain*.
- 3) Perangkat lunak Matlab telah berhasil melakukan serangkaian tahapan yang diperlukan dalam prediksi nilai produktivitas pada pekerjaan *prefabricated vertical drain*.

REFERENSI

- Agustin, M., & Prahasto, T. (2012). Penggunaan Jaringan Syaraf Tiruan Backpropagation untuk Seleksi Penerimaan Mahasiswa Baru pada Jurusan Teknik Komputer di Politeknik Sriwijaya. *Universitas Diponegoro*, 02, 4–32.
- Al-Zwainy, F. M. S., Rasheed, H. A., & Ibraheem, H. F. (2012). Development of the Construction Productivity Estimation Model using Artificial Neural Network for Finishing Works for Floors with Marble. *ARPJ Journal of Engineering and Applied Sciences*, 7(6), 714–722.
- Aspar, W. A. N., & Fitriani, E. N. (2016). Pengaruh Jarak dan Pola Prefabricated Vertical drain (PVD) Pada Tanah Lunak. *Majalah Ilmiah Pengkajian Industri*, 10 Nomer 1, 41–50.
- Dhiva, I. G. N. A. (2014). Analisis Produktivitas Pengeboran Pondasi Bor Pile dengan Menggunakan Mesin Bor Ringan. *Jurnal Rekayasa Sipil/Volume 8*, No. 2, 8(2), 128–135.
- Heravi, G., & Eslamdoost, E. (2015). Applying Artificial Neural Networks for Measuring and Predicting Construction-Labor Productivity. *Journal of Construction Engineering and Management*, (11), 1–11.
- Ok, S. C., & Sinha, S. K. (2006). Construction Equipment Productivity Estimation using Artificial Neural Network Model. *Construction Management and Economics*, 24(10), 1029–1044.
- Rashidi, A., Nejad, H. R., & Maghiar, M.

- (2014). Productivity Estimation of Bulldozers using Generalized Linear Mixed Models. *KSCE Journal of Civil Engineering*, 18(6), 1580–1589.
- Siang, J. J. (2005). *Jaringan Syaraf Tiruan & Pemogramannya Menggunakan Matlab*. Andi Yogyakarta.
- Warsito, J. Y. E., Hatmoko, J. U. D., Wibowo, M. A., & HA, R. (2016). An Artificial Neural Network Model of Hydraulic Static Pile Driver Productivity in Silt Soil. *Australian Journal of Basic and Applied Sciences*, 10(1), 489–497. Retrieved from <https://ssrn.com/abstract=2791462>
- Wasserman, P. D. (1989). *Neural Computing: Theory and Practice*. New York: Van Nostrand Reinhold.