

## Keefektifan Neural Network dalam Memprediksi Respon Eksperimen Ortogonal Array Sebagai Alternatif Pendekatan Taguchi Klasik

M. Arbi Hadiyat

Jurusan Teknik Industri, Universitas Surabaya  
Jl. Raya Kalirungkut, Surabaya 60293, Indonesia  
Email: [arbi@ubaya.ac.id](mailto:arbi@ubaya.ac.id)

Kestrilia Rega Priianti

Fakultas Teknobiologi (konsentrasi Bioinformatika), Universitas Surabaya  
Jl. Raya Kalirungkut, Surabaya 60293, Indonesia  
Email: [kestrilia@ubaya.ac.id](mailto:kestrilia@ubaya.ac.id)

### Abstrak

*Tujuan dari penelitian ini adalah untuk mengevaluasi efektivitas neural network dalam memprediksi respon eksperimen sebagaimana telah diaplikasikan oleh beberapa peneliti. Beberapa penelitian menunjukkan bahwa neural network biasanya digunakan untuk memprediksi data dengan rekaman data historis maupun cross-section yang cukup. Dalam eksperimen Taguchi, orthogonal array yang disediakan terdiri dari kombinasi level yang disusun untuk eksperimen. Kombinasi level tersebut beserta responnya dapat diperlakukan sebagai data training untuk neural network. Bagaimanapun, permasalahan muncul berkaitan dengan kekompleksan model neural network, yakni overestimate dan penentuan banyaknya hidden node dan layer. Selain neural network, pendekatan Taguchi klasik yang sederhana dapat dipertimbangkan untuk menghasilkan prediksi yang lebih efektif, dengan hanya perbedaan yang kecil meski neural network lebih baik. Penelitian ini menggunakan kasus-kasus eksperimen Taguchi yang diambil dari beberapa penelitian, dan membandingkan kedua metode tersebut.*

**Kata kunci:** neural networks, Taguchi, prediksi, orthogonal array

### Abstract

*The aim of this research is to evaluate the effectiveness of Neural Networks in predicting Taguchi experiment response as applied by some experimenters. As shown in many research, neural networks is used to predict data with enough historical or cross-sectional records. In Taguchi experiment, the orthogonal array provided by the design consists of level combinations conducted for performing experiment. It is considerable to treat the level combinations and experiment response as training data for neural networks prediction, However, the problems are arises within the complexity of neural networks modeling, that is the overestimate and the determination of hidden node and layer. Instead of neural networks modeling, the simple classical Taguchi predicting is considerably more effective, with only small differences compared by a little better neural networks predicting. This research uses cases from Taguchi experiment data that is conducted by some experimenters, and comparing the result between both method.*

**Keywords:** neural networks, Taguchi, predicting, orthogonal array

## 1. Pendahuluan

Beberapa dekade terakhir, ketika metode Taguchi menjadi salah satu alternatif untuk mendapatkan setting mesin yang optimal, berbagai penelitian tentang keefektifan metode ini

bermunculan, mulai dari penerapan sederhana, hingga membandingkannya dengan metode-metode lainnya [1, 2]. Metode Taguchi, secara sederhana terbukti efektif dalam proses optimasi mesin di samping metode *Respon surface* yang selama ini telah mendominasi bidang ini.

Meskipun Taguchi telah menjadi rujukan bagi para peneliti dalam kasus-kasus optimasi, bukan berarti metode ini tanpa kekurangan. Beberapa penelitian menunjukkan bahwa Taguchi tidak sepenuhnya efektif dalam melibatkan adanya interaksi antar-faktor [3], terutama untuk interaksi lebih dari dua faktor. Kelemahan ini terkadang tidak menjadi sebuah masalah yang cukup pelik, karena bagaimanapun, interaksi lebih dari dua faktor menjadi sesuatu yang dihindari oleh para peneliti [4], dan sebuah penelitian Taguchi akan berusaha memastikan bahwa interaksi lebih dari dua faktor tidak terjadi.

Dalam perkembangannya, selain menentukan optimasi parameter mesin, beberapa penelitian tentang optimasi mengarah pada penentuan model untuk memprediksi respon. Ketika sebuah respon akan diprediksi, maka diperlukan sebuah model yang mampu mewakili kondisi setting parameter mesin yang sesungguhnya, sehingga perubahan-perubahan pada parameter tersebut menjadi dasar untuk melihat perilaku responnya. Beberapa penelitian tentang ini antara lain adalah [5, 6]. *Neural Networks* menjadi salah satu model yang digunakan untuk prediksi respon dari proses optimasi Taguchi.

Terkait dengan kompleksitas model *Neural Networks*, data-data yang dihasilkan oleh desain Taguchi, yang kemudian digunakan sebagai *training set*, menjadi permasalahan yang perlu mendapat perhatian khusus. *Neural Networks*, bagaimanapun adalah sebuah model matematis nonlinier dalam parameter dan variabel, yang dalam proses estimasinya memerlukan proses iteratif yang kompleks. Banyak atau sedikitnya data *training set*, terkadang menjadi salah satu pertimbangan apakah *Neural Networks* dengan baik dapat menjadi model yang dapat dipakai untuk prediksi.

## 2. Kajian Literatur

### 2.1. Desain Eksperimen Taguchi

Desain eksperimen telah banyak digunakan sebagai alat yang ampuh dalam proses perbaikan kualitas produk dan proses terutama dalam bidang manufaktur, sejak Sir Ronald Fisher pertama kali menerapkannya. Saat ini, berbagai area penelitian telah banyak yang menerapkan desain eksperimen, bahkan dalam bidang jasa [7]. Desain eksperimen klasik menjadi trendsetter untuk diterapkan oleh para peneliti, hingga akhirnya ditemukan metode Taguchi. Taguchi mendasarkan desain percobaannya pada Desain eksperimen klasik (khususnya desain *Respon surface*), namun memberikan pengalaman praktis yang lebih sederhana dalam menentukan kombinasi level faktor yang optimal [8].

Belavendram [8] menyatakan bahwa Taguchi lebih mengarah ke *robust design*, sebuah desain yang melibatkan variasi proses di dalamnya. Taguchi menjalankan prosedurnya melalui tiga tahapan, yakni *system design*, *parameter design* dan *tolerance design*. Pada tahapan *system design*, pengetahuan keteknisian digunakan untuk menentukan konfigurasi dasar produk (atau proses) oleh *engineer*. Pada tahapan *parameter design*, nilai spesifik untuk parameter sistem ditentukan. *Parameter design*, yang merupakan tahapan antara *system design* dan *tolerance design*, memberikan nilai terbaik untuk parameter sistem. Tujuan dari *parameter design* adalah identifikasi *setting* parameter yang meminimumkan variasi respon, serta mengaturnya untuk mencapai target.

*Tolerance design* digunakan untuk menentukan *tolerance parameter* [9]. Taguchi merekomendasikan metode desain eksperimen sebagai salah satu alat untuk *quality improvement*, khususnya selama tahapan *parameter design* dan *tolerance design* untuk meminimalkan biaya manufaktur dan meningkatkan kualitas. Di dalam proses analisis Taguchi, terdapat tiga macam karakteristik kualitas, mengacu pada target yang akan dicapai, yakni *SN ratio smaller-the-better*, *SN ratio nominal-the-best*, *SN ratio larger-the-better*, *SN ratio signed-target*, dan *SN ratio-fraction defective* [8]. Tabel orthogonal array yang merupakan pengembangan dari desain faktorial [3], digunakan dalam Taguchi untuk mendesain eksperimennya.

Penemuan fenomenal yang dilakukan oleh Taguchi adalah besaran *signal-to-noise ratio* (S/N *ratio*), yang mengukur seberapa besar variabilitas produk/proses dengan adanya gangguan noise factor. Semakin tinggi nilai S/N *ratio*, maka semakin baik karakteristik kualitas yang didapatkan. Penelitian ini menggunakan S/N *ratio* untuk *fraction defective*, dengan persamaan:

$$\eta = -10 \log_{10} \left[ \frac{1}{p} - 1 \right] \quad (1)$$

di mana  $p$  adalah fraksi cacat. Menurut Belavendram [8], pada *SNR fraction-defective* yang digunakan untuk menunjukkan fraksi cacat, nilai terbaik untuk  $p$  adalah nol, namun terkadang  $p$  digunakan untuk menunjukkan hasil dimana nilai terbaiknya adalah 1 (100%). Maka dari itu, Phadke [10] merumuskan *SNR fraction-defective* dengan tujuan memaksimalkan nilai SNR untuk mencapai nilai terbaik  $p$  yaitu nol, dengan rumusan sebagai berikut:

$$\eta = -10 \log_{10} \left[ \frac{p}{1-p} \right] \quad (2)$$

## 2.2. Neural Networks

Dalam dekade terakhir, berbagai penelitian tentang Neural Networks banyak dilakukan, mulai dari penerapan untuk analisis data, hingga aplikasi engineering. Neural Networks merupakan sebuah model matematis yang fleksibel untuk memodelkan perilaku data secara nonlinier. Beberapa peneliti di bidang matematika dan informatika memasukkan Neural Networks sebagai model kompleks yang mampu secara buatan mencerminkan sistem syaraf manusia, sehingga dapat dianggap sebagai sebuah model kecerdasan buatan [11].

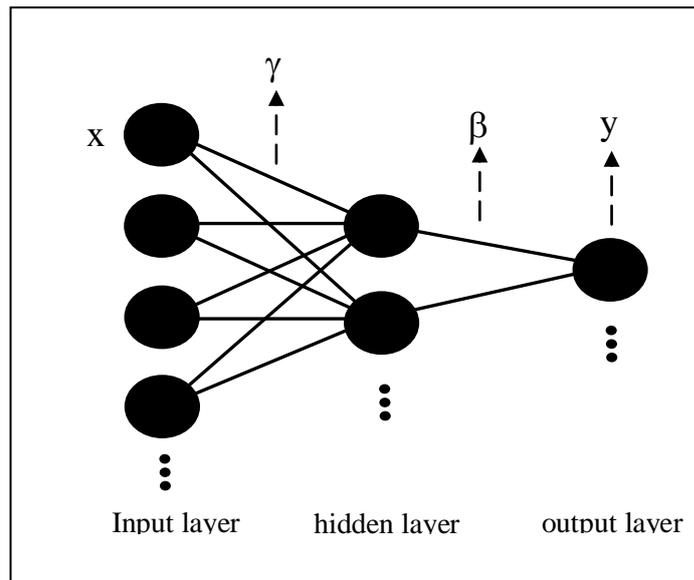
Di antara sekian banyak jenis model Neural Networks yang dikembangkan, Feedforward Neural Networks menjadi salah satu model yang banyak diterapkan, seperti pada pemrosesan citra [12], peramalan [13], dan berbagai aplikasi lainnya. Skema jaringan syaraf yang diadopsi oleh model Neural Network ini, secara umum membentuk arsitektur Feedforward Neural Networks dengan sketsa seperti pada gambar 1.

Setiap node di masing-masing layer, terhubung dengan node-node di layer terdekat. Garis penghubung menggambarkan weight (bobot) hubungan antar node. Model umum untuk Feedforward Neural Networks adalah:

$$y = \sum \beta_j f \left( \sum \gamma_{ij} x_i \right) + \varepsilon \quad (3)$$

di mana  $i$  adalah banyaknya input, dan  $j$  adalah banyaknya hidden nodes, dan  $f$  adalah fungsi aktivasi yang biasanya berbentuk fungsi logistik. Arsitektur Neural Networks yang digunakan

dalam penelitian ini mempunyai input node sebanyak faktor utama yang menjadi parameter mesin, satu hidden layer, dan output berupa data proporsi produk cacat.



Gambar 1. Arsitektur umum Neural Networks

Proses estimasi (dalam beberapa referensi disebut sebagai *learning process*) weight  $\gamma$  dan  $\beta$  pada pemodelan Neural Networks didasarkan pada data-data training set.

### 3. Metodologi

Penelitian menggunakan data sekunder, yang didapatkan dari hasil optimasi mesin injection molding menggunakan Taguchi [14], dengan respon penelitian adalah proporsi produk cacat baik dalam bentuk prosentase (rasio) maupun dalam bentuk S/N ratio. Optimasi mesin telah dilakukan, dan setting parameter optimal telah didapatkan untuk menekan prosentase produk cacat.

Tahap awal, berdasarkan hasil optimasi tersebut, data yang telah ada menjadi acuan untuk membentuk model Neural Networks. Model ini diharapkan dapat membuat prediksi respon ketika setting parameter harus diubah sesuai keinginan peneliti. Ketika nilai respon harus diprediksi, maka model Neural Networks harus secara matematis dapat menangkap pola perilaku data. Kendala yang mungkin terjadi adalah data hasil percobaan yang terlalu sedikit, sehingga model Neural Networks kemungkinan besar akan jatuh pada kondisi *overfitting* [12].

Langkah selanjutnya, membangun model Neural Networks. Beberapa arsitektur model Neural Networks diuji kemampuan prediksinya untuk data-data di atas. Pemilihan arsitektur terbaik menggunakan kondisi error terkecil yang diukur menggunakan besaran *Mean Square Error* (MSE), sebagaimana dikemukakan oleh Bishop [11]. Arsitektur Neural Network terbaik menjadi sebuah model prediksi, khususnya dalam studi kasus mesin injection molding di atas.

### 4. Hasil dan Diskusi

Data yang digunakan untuk penelitian ini adalah sebagai berikut:

Tabel 1. Faktor dan *level* dalam eksperimen

Kode	Faktor	Nilai riil Level 1	Nilai riil Level 2
A	<i>nozzle temp</i>	195°C	205°C
B	tekanan <i>inject</i>	30 Bar	35 Bar
C	<i>injection timer</i>	3.5 sekon	4 sekon
D	<i>cooling timer</i>	13 sekon	17 sekon
AxB	interaksi <i>nozzle temp</i> dan tekanan <i>inject</i>		
AxD	interaksi <i>nozzle temp</i> dan <i>cooling timer</i>		
BxD	interaksi tekanan <i>inject</i> dan <i>cooling timer</i>		

Tabel 2. Hasil eksperimen Taguchi

Exp	Faktor							Jenis Cacat			Jumlah Produk Cacat	Jumlah Produksi
	A	B	AxB	D	AxD	BxD	C	Tidak Utuh	Retak	Kasar		
1	1	1	1	1	1	1	1	23	0	0	23	500
2	1	1	1	2	2	2	2	18	11	0	29	500
3	1	2	2	1	1	2	2	10	0	0	10	500
4	1	2	2	2	2	1	1	8	11	0	19	500
5	2	1	2	1	2	1	2	1	0	14	15	500
6	2	1	2	2	1	2	1	4	0	0	4	500
7	2	2	1	1	2	2	1	0	0	15	15	500
8	2	2	1	2	1	1	2	0	0	1	1	500
TOTAL								64	22	30	116	4000

Rancangan Taguchi menggunakan Orthogonal Array L<sub>8</sub>. Sebagai respon, proporsi cacat dihitung untuk setiap eksperimen. Hasil setting optimal yang didapatkan oleh Rosiawan, dkk [14] adalah A<sub>2</sub>, A<sub>2</sub> D<sub>2</sub>, D<sub>2</sub>, dan B<sub>2</sub>. Hasil ini masih dapat melibatkan adanya interaksi, namun ketika model Neural Networks akan diimplementasikan, maka interaksi antar-faktor tidak diikutsertakan.

Input model Neural Networks adalah nilai riil (bukan kode) masing-masing level dari setiap faktor, sebagaimana yang tercantum pada tabel 2. Sebanyak 8 kali eksperimen yang telah dilakukan tersebut, menjadi data training set bagi model Neural Networks yang akan dibangun. Model Neural Networks yang akan dibangun hanya terdiri dari satu *hidden layer* saja, mengingat data training set tidak terlalu banyak. Input model Neural Networks adalah nilai-nilai riil dari 4 faktor utama (faktor A, B, C, D), dan sebagai output adalah persentase produk cacat. Tabel 3 adalah training set yang akan ditangkap polanya oleh model Neural Networks.

Mengacu pada Tabel 1, maka arsitektur Neural Networks menggunakan 4 input node, dan 1 output node. Banyaknya node pada *hidden layer* dikombinasikan antara 2 hingga 5 node untuk mendapatkan model Neural Networks yang paling optimal.

Hasil pemodelan Neural Networks difokuskan pada kemampuan untuk memprediksi ada pada Tabel 4.

Ukuran kebaikan model didasarkan pada nilai MSE yang terkecil. Secara sederhana, Arsitektur model Neural Networks dengan 5 dan 4 *hidden nodes* memberikan prediksi dengan nilai MSE terkecil, bahkan cenderung mendekati nol. Bishop [11] menyebutkan bahwa jika sebuah model Neural Networks berhasil melakukan learning dengan data *training set* yang tidak terlalu banyak, maka terdapat kemungkinan yang cukup besar terjadinya *overfitting*.

Nilai MSE yang terlalu kecil pada arsitektur Neural Networks dengan 5 dan 4 hidden node, mengindikasikan terjadinya *overfitting*. Ketika *overfitting* terjadi, maka model neural networks sangat baik prediksinya namun hanya berlaku untuk data training set yang diacu, dan tidak dapat digeneralisasi untuk input-input lainnya.

Tabel 3. Training set

Data ke	Faktor				Proporsi Produk Cacat
	Input node 1 A	Input node 2 B	Input node 3 C	Input node 4 D	
1	195	30	3.5	13	0.046
2	195	30	4	17	0.058
3	195	35	3.5	17	0.02
4	195	35	4	13	0.038
5	205	30	3.5	17	0.03
6	205	30	4	13	0.008
7	205	35	3.5	13	0.03
8	205	35	4	17	0.002

Tabel 4. Hasil Prediksi Neural Networks dan Taguchi

Target (respon)	Prediksi Taguchi	Prediksi NN 2 hidden nodes	Prediksi NN 3 hidden nodes	Prediksi NN 4 hidden nodes	Prediksi NN 5 hidden nodes
0.046	0.051	0.052	0.084	0.046	0.046
0.058	0.043	0.052	0.060	0.058	0.058
0.020	0.035	0.020	0.073	0.020	0.020
0.038	0.033	0.038	0.058	0.038	0.038
0.030	0.025	0.030	0.048	0.030	0.030
0.008	0.023	0.008	0.025	0.008	0.008
0.030	0.015	0.030	0.079	0.030	0.030
0.002	0.007	0.002	0.003	0.002	0.002
MSE	$1.037 \cdot 10^{-5}$	$4.40 \cdot 10^{-5}$	0.000953	$8.16 \cdot 10^{-27}$	$2.47 \cdot 10^{-30}$

Sementara itu, prediksi yang dilakukan secara klasik menggunakan metode Taguchi, memberikan hasil yang tidak mengalami *overfitting*. Perhitungan prediksi ini dilakukan secara linier, menggunakan input kombinasi level-level setiap faktor [15]

## 5. Kesimpulan

Efektivitas Neural Network yang digunakan untuk memprediksi respon hasil eksperimen Taguchi masih memerlukan pengkajian yang lebih dalam. Keinginan untuk dapat menjadikan Neural Networks sebagai model simulasi ketika seorang peneliti mencoba untuk merubah level faktor-faktor sehingga prediksi respon dapat diperkirakan (tanpa melakukan eksperimen) berpeluang untuk diteliti lebih lanjut. Kendala utama adalah data-data eksperimen Taguchi (rancangan Orthogonal Array) yang biasanya tidak terlalu banyak, sementara Neural Networks membutuhkan training set yang cukup banyak untuk bisa *learning* dengan baik.

## 6. Daftar Rujukan

- [1] Lo, Y. & Tsao, C.C. (2002). Integrated Taguchi Method and Neural Network Analysis of Physical Profiling in the Wirebonding Process, *IEEE Transaction on Components and Packaging Technologies*, 25 (2), 270-277.
- [2] Zeydan, M. (2008). Modelling the Woven Fabric Strength Using Artificial Neural Networks and Taguchi Methodologies, *International Journal of Clothing Science and Technology*, 20 (2), 104-118.
- [3] Roy, R. (1990). *A Primer on Taguchi Method*, Van Nostrand Reinhold International Company Ltd, London.
- [4] Park, S.H. (1996). *Robust Design and Analysis for Quality Engineering*, Chapman & Hall, London.
- [5] Sukthomya, W. & Tannock, J.D.T. (2005). Taguchi experimental design for manufacturing process optimisation using historical data and network process model, *International Journal of Quality & Reliability Management*, 22 (5), 485-502.
- [6] Sanjari, M., Taheri, K. & Movahedi, M.R. (2009). An Optimization Method for Radial Forging Process using ANN and Taguchi Method, *International Journal of Advanced Manufacturing Technology*, 40, 776-784.
- [7] Taner, T. & Anthony, J. (2006). Applying Taguchi methods to healthcare, *Leadership in Health Services*, 19 (1), 26-35.
- [8] Belavendram, N. (1995). *Quality by Design*, Prentice Hall International, Hertfordshire.
- [9] Montgomery, D.G. (1997). *Design and Analysis of Experiments*, 4<sup>th</sup> ed., Wiley, NY.
- [10] Phadke, M.S. (1989). *Quality Engineering Using Robust Design*, Prentice Hall International, USA.
- [11] Bishop, C.M. (1999). *Neural Networks for Pattern Recognition*, Oxford Univ Press, NY.
- [12] Ripley, B.D. (1996). *Pattern Recognition and Neural Networks*, Cambridge University Press, NY.
- [13] Suhartono & A. Hadiyat (2003). Building the optimal architecture of neural network models for time series as statistical modeling, *Proceedings of International Conference on Mathematics and Its Applications*, SEAMS-Gadjah Mada University.
- [14] Rosiawan, M., Alvina, B. & Hadiyat, M.A. (2011). Optimisasi Parameter Vertical Injection Molding Menggunakan Metode Taguchi Untuk Data Persentase Cacat, to appear on *Proceedings of 6th National Industrial Engineering Conference*, Universitas Surabaya.
- [15] Peace, G.S. (1993). *Taguchi Methods: A Hands on Approach*, Addison Wesley Inc. Massachusetts.