

PERBANDINGAN ANTARA MODEL *NEURAL NETWORK* DAN MODEL *DUANE* UNTUK EVALUASI KETEPATAN PREDIKSI WAKTU KERUSAKAN SUATU KOMPONEN

Tan Te Feng¹⁾, Suhartono²⁾, Julius Mulyono³⁾

E-mail: frans_tanoni@yahoo.com, suhartono@statistika.its.ac.id, seeyong@wima.ac.id

ABSTRAK

Artificial Neural Network atau yang dikenal dengan jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia. Sedangkan model *Duane* adalah model yang dikenal baik untuk sistem perbaikan dengan asumsi waktu antar kerusakan saat ini berkaitan dengan waktu antar kerusakan satu waktu sebelumnya, atau bahkan dengan waktu-waktu antar kerusakan sebelumnya. Dalam analisis keandalan, kedua model tersebut dapat dipergunakan untuk menentukan waktu pergantian, atau perawatan suatu komponen. Keuntungan melakukan pergantian, atau perawatan secara berkala adalah dapat mengurangi, atau menekan kerugian yang akan ditimbulkan bila suatu proses produksi berhenti.

Dalam artikel ini akan dilakukan kajian perbandingan antara model *Neural Network* dengan model *Duane* untuk analisis keandalan berkaitan dengan penentuan jadwal pergantian, atau perawatan suatu komponen dalam memprediksi terjadinya kegagalan. Pemilihan model terbaik berdasarkan pada model yang menghasilkan tingkat kesalahan prediksi yang paling kecil untuk suatu data kerusakan dengan membandingkan nilai *Root Mean Square Error (RMSE)*.

Kata kunci : model *Neural Network*, model *Duane*, prediksi, waktu kerusakan

PENDAHULUAN

Analisis keandalan adalah suatu metode yang secara umum banyak digunakan dalam perawatan, dan pergantian suatu komponen. Hal ini dilakukan dengan harapan suatu komponen, atau peralatan dalam suatu proses produksi dapat berjalan dengan stabil. Perbaikan atau pergantian komponen pada suatu proses produksi seringkali menghasilkan kondisi yang tidak dapat dikatakan “*as good as new*”. Dengan melakukan pergantian, atau perawatan komponen secara berkala dapat mengurangi, atau menekan kerugian yang akan ditimbulkan bila suatu proses produksi berhenti. Kendala yang seringkali dihadapi adalah menentukan waktu pergantian, atau perawatan yang tepat dari suatu komponen.

Ada beberapa model yang dapat dipergunakan untuk memprediksi waktu antar kegagalan yang tidak random, di antaranya model *Neural Network*, model *Duane*, dan model *ARIMA*. Kajian perbandingan yang telah dilakukan adalah antara model *ARIMA* dan *Duane*^[1], serta antara *ARIMA* dan *Neural Network*^[2]. Dalam artikel ini akan dilakukan kajian perbandingan antara model *Neural Network* dengan model *Duane* untuk analisis keandalan berkaitan dengan penentuan jadwal

pergantian, atau perawatan suatu komponen dalam memprediksi terjadinya kegagalan.

Pemilihan model terbaik berdasarkan pada model yang menghasilkan tingkat kesalahan prediksi yang paling kecil untuk suatu data kerusakan pada suatu komponen, atau sistem. Sebagai studi kasus akan menggunakan tiga macam data sekunder yaitu data kerusakan *Equipment* dari PT.Petro Kimia Gresik dan data waktu antar kerusakan dari artikel tulisan Xie dan Ho (1999) yaitu data waktu antar kerusakan *software*, dan data waktu antar kerusakan mesin fotokopi^[1].

TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan teori-teori yang berkaitan dengan model-model keandalan, yaitu model *Neural Network* dan model *Duane*.

Model *Neural Network*

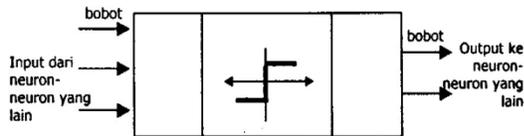
Artificial Neural Network atau yang dikenal dengan jaringan syaraf tiruan adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia. Ada beberapa tipe jaringan syaraf, namun demikian, hampir semuanya memiliki komponen-komponen yang sama seperti halnya

¹⁾ Mahasiswa di Fakultas Teknik Jurusan Teknik Industri Universitas Katolik Widya Mandala Surabaya

²⁾ Staf Pengajar di Jurusan Statistika Institut Teknologi Sepuluh Nopember Surabaya

³⁾ Staf Pengajar di Fakultas Teknik Jurusan Teknik Industri Universitas Katolik Widya Mandala Surabaya

otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, dan ada hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan ke luarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Gambar 1 menunjukkan struktur neuron pada jaringan syaraf.



Gambar 1. Struktur neuron jaringan syaraf

Jika dilihat, neuron buatan ini sebenarnya mirip dengan sel neuron biologis. Neuron-neuron buatan tersebut bekerja dengan cara yang sama pula dengan neuron-neuron biologis. Informasi (disebut dengan *input*) akan dikirim ke neuron dengan bobot kedatangan tertentu. *Input* ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila *input* tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan *output* melalui bobot-bobot *output*nya ke semua neuron yang berhubungan dengannya. Demikian seterusnya.

Pada jaringan syaraf, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layers*). Biasanya neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelumnya, dan sesudahnya (kecuali lapisan *input* dan lapisan *output*). Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan *input* sampai ke lapisan *output* melalui lapisan yang lainnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*). Tergantung pada algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan secara mundur pada jaringan.

Terdapat beberapa macam fungsi aktivasi yang digunakan dalam model *Neural Network*

antara lain adalah adalah fungsi *sigmoid biner* (*logsig*). Fungsi *sigmoid biner* memiliki nilai pada kisaran 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan model *Neural Network* yang terletak pada interval 0 sampai 1. Fungsi *sigmoid biner* dirumuskan sebagai:

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

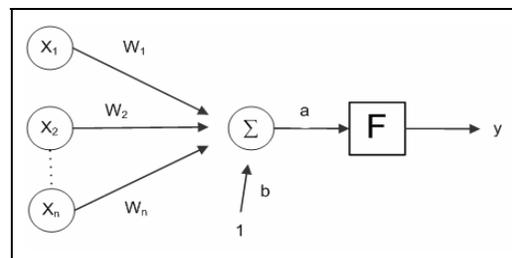
Sebelum pelatihan biasanya dikenal proses penskalaan nilai yang disebut sebagai *premnmx*. Penskalaan ini berfungsi untuk mengubah nilai *input* awal ke dalam kisaran nilai antara -1 hingga 1, di mana persamaan *premnmx* adalah sebagai berikut:

$$P_n = \frac{2x(p - \min p)}{(\max p - \min p) - 1} \quad (2)$$

Setelah proses pelatihan didapatkan, maka proses untuk mengembalikan nilai penskalaan ke bentuk asal dikenal sebagai *post processing* yang persamaan:

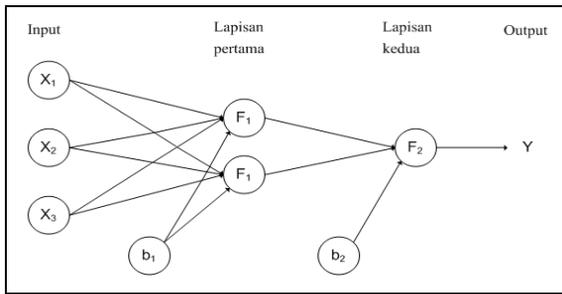
$$p = 0,5(pn + 1)x(\max p - \min p) + \min p \quad (3)$$

Multi layer perceptron (*Feed Forward Neural Network*) merupakan algoritma pembelajaran yang terawasi, dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Gambaran *perceptron* secara sederhana ditampilkan pada Gambar 2 berikut.



Gambar 2. Perceptron secara sederhana

Pada Gambar 2 *input* jaringan dilambangkan dengan X, sedangkan bobot *input* dilambangkan dengan W, bobot bias dilambangkan dengan b, dan fungsi aktivasi yang akan digunakan dilambangkan dengan F, sedangkan contoh gambar *multi layer perceptron* ditampilkan pada Gambar 3 berikut.



Gambar 3. Multi layer perceptron

Model Duane

Model *Duane* banyak diketahui sebagai model untuk sistem perbaikan (Duane 1964). Model *Duane* sesuai dengan proses model *non-homogeneous poisson*. Model ini telah digunakan, dan dipelajari secara intensif^[3,4]. Model *Duane* sangat fleksibel, dan dapat ditunjukkan sebagai berikut:

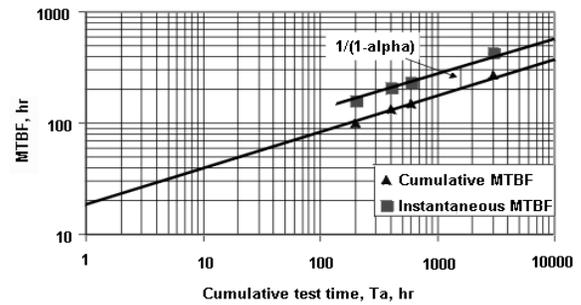
$$m(t) = at^b, t \geq 0 \quad (4)$$

dengan a dan b adalah nilai konstan yang lebih dari nol.

Pada tahun 1962, J. T. Duane menunjukkan sebuah hasil yang menunjukkan data kegagalan dari beberapa sistem yang berbeda selama program pembangunannya, menganalisis data, dan mengamati bahwa kumulatif dari *MTBF* terhadap waktu kumulatif pengoperasian mendekati garis lurus jika diplotkan pada sebuah kertas. Contoh urutan kegagalan disajikan pada Gambar 4 dan hubungan antara *cumulative test time* terhadap *MTBF cumulative MTBF* disajikan pada Gambar 5.



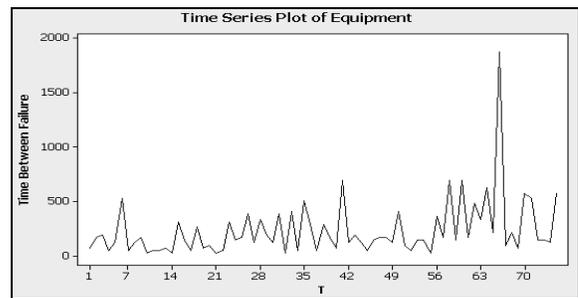
Gambar 4. Contoh urutan kegagalan untuk sistem yang diperbaiki



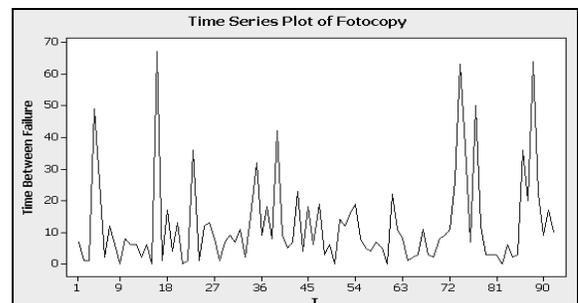
Gambar 5. Hubungan antara *cumulative MTBF* terhadap *cumulative test time* yang dikemukakan *Duane*

METODE PENELITIAN

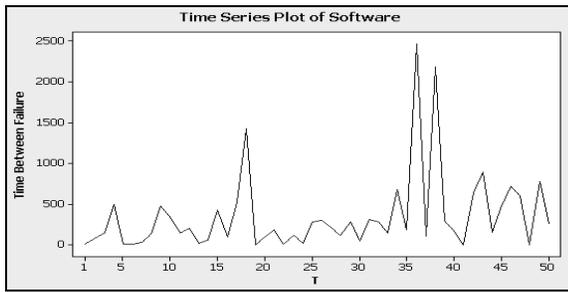
Pada penelitian ini akan dilakukan kajian perbandingan ketepatan ramalan antara model *Neural Network* dan model *Duane*. Sebagai studi kasus, data riil yang digunakan adalah data waktu antar kerusakan *Equipment* dari PT. Petro Kimia Gresik, data waktu antar kerusakan dari paper Xie dan Ho (1999) yaitu data waktu antar kerusakan *software*, dan data waktu antar kerusakan mesin fotokopi. Plot *time series* data waktu antar kerusakan untuk masing-masing kasus ditampilkan pada Gambar 6 berikut.



(a). Kerusakan equipment



(b). Kerusakan *software*



(c). Kerusakan mesin fotokopi

Gambar 6. Plot series data kerusakan *equipment*, plot series data kerusakan *Software* dan plot series data kerusakan mesin fotokopi

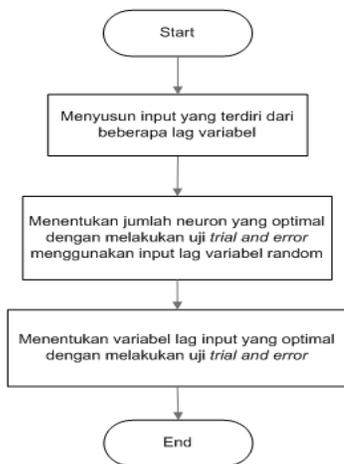
Evaluasi perbandingan ketepatan ramalan dilakukan dengan membagi masing-masing data menjadi dua bagian, yaitu data *training* (untuk pemodelan), dan data *testing* (untuk validasi). Kriteria pemilihan model yang terbaik digunakan *Root Mean Squares Error (RMSE)* terutama pada data *testing*.

HASIL PENELITIAN DAN PEMBAHASAN

Pada bagian ini akan diberikan hasil-hasil perbandingan antara model *Neural Network* dan *Duane* pada data riil kerusakan *equipment*, *software*, dan mesin fotokopi.

Hasil pada data kerusakan equipment di PT. Petrokimia, Gresik

Untuk mendapatkan hasil peramalan terbaik dengan menggunakan model *Neural Network*, maka perlu disusun strategi pemodelan yang ditampilkan pada Gambar 7 berikut agar hasil *output* yang optimal bisa dicapai.



Gambar 7. Diagram alir strategi pemodelan untuk model *Neural Network*

Penjelasan langkah-langkah pada diagram alir di atas adalah sebagai berikut:

Langkah 1: menyusun *input* menjadi beberapa variabel *lag*, dalam kasus ini variabel *lag input* disusun menjadi 12 variabel *lag*. Berikut pada Tabel 1 akan ditampilkan contoh susunan variabel *lag input*.

Tabel 1. Susunan variabel *lag input*

| No | TBF | | | | |
|----|---------|-----------|-----------|-----------|-----------|
| | Y_t | Y_{t-1} | Y_{t-2} | Y_{t-3} | Y_{t-4} |
| 1 | 72.000 | * | * | * | * |
| 2 | 168.000 | 72.000 | * | * | * |
| 3 | 192.000 | 168.000 | 72.000 | * | * |
| 4 | 48.000 | 192.000 | 168.000 | 72.000 | * |
| 5 | 120.000 | 48.000 | 192.000 | 168.000 | 72.000 |
| 6 | 528.000 | 120.000 | 48.000 | 192.000 | 168.000 |
| 7 | 48.000 | 528.000 | 120.000 | 48.000 | 192.000 |
| 8 | 120.000 | 48.000 | 528.000 | 120.000 | 48.000 |
| 9 | 168.000 | 120.000 | 48.000 | 528.000 | 120.000 |
| 10 | 24.000 | 168.000 | 120.000 | 48.000 | 528.000 |

Langkah 2: menentukan jumlah neuron optimal dengan melakukan uji *trial and error* dengan *input* yang digunakan adalah random, sebagai contoh pada repetisi pertama digunakan *input* Y_{t-1} , Y_{t-2} dan Y_{t-3} , lalu untuk repetisi kedua digunakan *input* Y_{t-1} , Y_{t-2} hingga Y_{t-5} , sedangkan untuk repetisi ketiga digunakan *input* Y_{t-1} , Y_{t-2} hingga Y_{t-12} . Hasil uji *trial & error* menunjukkan bahwa jumlah neuron yang optimal adalah 1.

Langkah 3: menentukan variabel *lag input* optimal dengan uji *trial and error* dengan menggunakan neuron pada lapisan tersembunyi sebanyak 1. Contoh uji *trial and error* penentuan variabel *lag input* yang optimal ditampilkan pada Tabel 2, di mana pada kolom *input* berisikan variabel *lag input* mulai dari Y_{t-1} hingga Y_{t-12} , sedangkan pada kolom *RMSE testing* berisikan hasil perhitungan validasi data, yaitu selisih nilai prediksi *output* model *Neural Network* dengan data riil. Hasil penentuan variabel *lag* yang terpilih adalah yang memiliki nilai *RMSE testing* yang terkecil.

Tabel 2. Uji *trial and error* penentuan *input* optimal dengan 1 *input* data kerusakan *equipment*

| Input ke-1 | MSE | | RMSE |
|------------|----------|------------|---------|
| | Training | Testing | Testing |
| Y_{t-1} | 0,088 | 25195,213 | 158,730 |
| Y_{t-2} | 0,075 | 81361,858 | 285,240 |
| Y_{t-3} | 0,087 | 24339,120 | 156,010 |
| Y_{t-4} | 0,075 | 74278,052 | 272,540 |
| Y_{t-5} | 0,088 | 69095,380 | 262,860 |
| Y_{t-6} | 0,067 | 140362,623 | 374,650 |
| Y_{t-7} | 0,089 | 17540,354 | 132,440 |
| Y_{t-8} | 0,078 | 47820,942 | 218,680 |
| Y_{t-9} | 0,087 | 96490,997 | 310,630 |
| Y_{t-10} | 0,081 | 22970,434 | 151,560 |
| Y_{t-11} | 0,085 | 37322,376 | 193,190 |
| Y_{t-12} | 0,088 | 23994,010 | 154,900 |

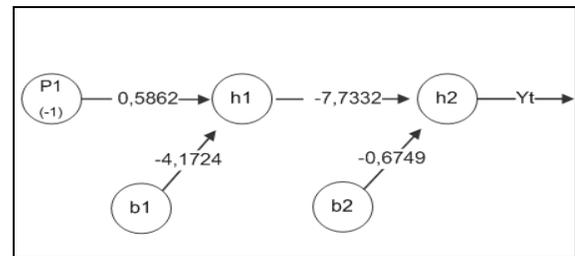
Hasil uji *trial and error* menunjukkan bahwa variabel *lag* yang optimal adalah Y_{t-7} karena memiliki nilai *RMSE testing* terkecil. Langkah selanjutnya adalah melakukan uji *trial and error* menggunakan dua variabel *lag input* di mana *input* pertama adalah Y_{t-7} . Hasil uji dengan menggunakan dua variabel *input* ditampilkan Tabel 3 berikut.

Tabel 3. Uji *trial and error* penentuan *input* optimal menggunakan 2 *input* data kerusakan

| Input ke-1 | Input ke-2 | MSE | | RMSE |
|------------|------------|----------|------------|---------|
| | | Training | Testing | Testing |
| Y_{t-7} | Y_{t-1} | 0,088 | 19433,754 | 139,405 |
| | Y_{t-2} | 0,074 | 71624,746 | 267,628 |
| | Y_{t-3} | 0,088 | 504021,903 | 709,945 |
| | Y_{t-4} | 0,075 | 46172,555 | 214,878 |
| | Y_{t-5} | 0,086 | 20321,073 | 142,552 |
| | Y_{t-6} | 0,066 | 150843,685 | 388,386 |
| | Y_{t-8} | 0,078 | 47823,567 | 218,686 |
| | Y_{t-9} | 0,087 | 50344,141 | 224,375 |
| | Y_{t-10} | 0,080 | 21598,711 | 146,965 |
| | Y_{t-11} | 0,084 | 28361,928 | 168,410 |
| | Y_{t-12} | 0,084 | 23934,565 | 154,708 |

Hasil uji *trial and error* menunjukkan bahwa nilai *RMSE testing* dengan menggunakan dua variabel *lag input* tidak sebaik dengan menggunakan satu variabel *lag input*, oleh karena itu uji *trial and error* dihentikan dan variabel *lag input* yang terpilih adalah Y_{t-7} .

Berdasarkan hasil uji *trial and error* didapatkan variabel *lag input* optimal sebanyak 1 yaitu Y_{t-7} , dan jumlah neuron optimal di lapisan tersembunyi sebanyak 1, maka bentuk arsitektur model *Neural Network* dapat disimbolkan sebagai NN(1,1,1). Pada Gambar 8 berikut akan ditampilkan contoh gambar jalannya interpretasi *output perceptron*, di mana *input* Y_{t-7} dilambangkan sebagai P1, sedangkan neuron di lapisan tersembunyi pada lapisan pertama sebanyak 1 dilambangkan sebagai h1, lapisan pada lapisan kedua sebanyak 1 dilambangkan sebagai h2 dan bobot bias pada lapisan pertama, dan kedua masing-masing dilambangkan sebagai b1 dan b2.



Gambar 8. Interpretasi *perceptron* dengan 1 *input*

Langkah selanjutnya adalah melakukan penskalaan nilai yang dikenal dengan *premnmx*, sebagai contoh perhitungan data target pada saat Y_{76} bernilai 192, diketahui nilai maksimal p adalah 192, sedangkan nilai minimal p adalah 24, maka dengan persamaan (2) dapat dihitung Pn sebagai berikut:

$$Pn = 2 \times \frac{(192 - 24)}{(192 - 24)} - 1 = 1$$

Setelah nilai penskalaan telah didapatkan, langkah selanjutnya adalah melakukan perhitungan untuk mendapatkan nilai prediksi pada saat Y_{76} dengan menggunakan *input* Y_{t-7} yaitu pada saat Y_{69} , di mana pada saat Y_{69} bernilai 72, setelah dilakukan proses penskalaan dengan *premnmx* memiliki nilai -1 dengan bobot *input* sebesar 0,5862, dan diketahui nilai bobot bias (b1) untuk lapisan pertama (h1) adalah -4,1724. Perhitungan untuk mendapatkan nilai *output* (h1) di atas dapat diselesaikan dengan persamaan (1) sebagai berikut:

$$h_1 = \frac{1}{1 + e^{-(-4,1724 + ((-1) \times 0,5862))}} = 0,0085046$$

Sedangkan perhitungan untuk mendapatkan nilai prediksi Y_{76} sebagai berikut :

$$Y_{76} = -0,6749 + ((-7,7332) \times 0,0085046) = -0,7407$$

Setelah nilai Y_{76} didapatkan maka perhitungan *post processing* dapat dilakukan untuk mengembalikan nilai penskalaan *premmx* ke bentuk asal pada saat Y_{76} sebagai berikut:

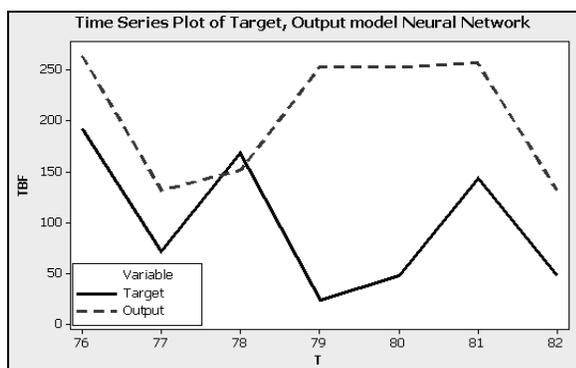
$$p = 0,5((-0,7407) + 1) \times (1872 - 24) + 24 = 263,6$$

Diketahui nilai *max p* adalah 1872 dan nilai *min p* adalah 24 yang didapat dari nilai data terbesar, dan nilai data yang terkecil. Hasil prediksi saat Y_{76} adalah sebesar 263,6 di mana nilai prediksi tersebut akan dibandingkan dengan data riil pada Tabel 4 berikut.

Tabel 4. Perbandingan nilai antara target dengan *output* jaringan

| T | Target | Output | Error |
|------|---------|---------|-----------|
| 76 | 192,000 | 263,605 | -71,605 |
| 77 | 72,000 | 131,766 | -59,766 |
| 78 | 168,000 | 151,623 | 16,377 |
| 79 | 24,000 | 252,637 | -228,637 |
| 80 | 48,000 | 252,637 | -204,637 |
| 81 | 144,000 | 256,495 | -112,495 |
| 82 | 48,000 | 131,766 | -83,766 |
| MSE | | | 17540,354 |
| RMSE | | | 132,440 |

Untuk mempermudah pembacaan antara target dengan *output* jaringan, maka pada Gambar 9 berikut akan ditampilkan dalam bentuk plot garis.



Gambar 9. Perbandingan antara target dengan *output*

Hasil plot perbandingan pada Gambar 9 di atas menunjukkan bahwa prediksi model *Neural Network* memiliki pola yang berfluktuasi naik, dan turun pada tiap waktu yang digambarkan dengan garis lurus putus-putus, sedangkan data

riil digambarkan dengan garis penghubung lurus tanpa putus-putus.

Sebagai pembanding model *Neural Network* dalam memprediksi, berikut akan ditampilkan perhitungan dengan menggunakan model *Duane*. Persamaan umum yang digunakan untuk Model *Duane* adalah:

$$Mc_{(T)} = bT^\alpha \tag{4}$$

dengan:

$Mc_{(T)}$ = nilai kumulatif dari data waktu antar kegagalan pada waktu ke-T yang dikenal dengan $TBF_{(T)}$ di mana nilai $Mc_{(T)}$ akan semakin meningkat seiring dengan bertambahnya waktu T.

Contoh perhitungan dengan menggunakan persamaan $Mc_{(T)} = bT^\alpha$ pada kasus ini dapat dilihat pada Tabel 5 berikut.

Tabel 5. Hasil Perhitungan untuk sistem

| T | $TBF_{(T)}$ | $Mc_{(T)}$ |
|----|-------------|------------|
| 1 | 72 | 72 |
| 2 | 168 | 240 |
| 3 | 192 | 432 |
| 4 | 48 | 480 |
| 5 | 120 | 600 |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |
| 73 | 144 | 16872 |
| 74 | 120 | 16992 |
| 75 | 576 | 17568 |

Untuk mendapatkan nilai b, dan α , maka lebih mudah jika persamaan $Mc_{(T)} = bT^\alpha$ dilinierkan dengan transformasi ln, yaitu:

$$\ln Mc_{(T)} = \ln b + \alpha \ln T \tag{5}$$

b, dan α dapat dicari dengan menggunakan regresi linier atau persamaan linear. Hasil analisis regresi disajikan pada Tabel 6 berikut.

Tabel 6. Hasil analisis Regresi Data Waktu antar Kerusakan *Equipment* dari segi mekanik

| Persamaan Regresi $\ln Mc_{(T)} = 4,486 + 1,18 \ln T$ | | | | |
|---|---------|---------|-------|-------|
| Predictor | Coef | SE Coef | T | P |
| ln b | 4,48665 | 0,05917 | 75,83 | 0,000 |
| ln T | 1,18191 | 0,01701 | 69,5 | 0,000 |

Dari hasil regresi dapat dilihat bahwa $\ln b = 4,486$ dan $\alpha = 1,18$ dengan p-value untuk $\ln b$ adalah 0,000, maka H_0 ditolak yang berarti $\ln b$ tidak sama dengan 0, dan p-value untuk α adalah 0,000, maka H_0 ditolak yang berarti α tidak sama dengan 0. Karena $\ln b$, dan α tidak sama dengan nol, maka keduanya berarti dalam model dan ditemukan $b = 88,823$ dan $\alpha = 1,18$. $\hat{M}c_{(T)}$ dicari dengan menggunakan b dan α dari hasil regresi tersebut dengan persamaan:

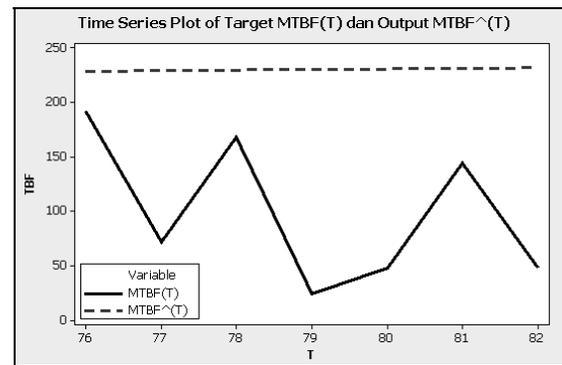
$$\hat{M}c_{(T)} = 88,82 \times T^{1,18} \quad (6)$$

Oleh karena persamaan baru telah didapatkan, maka untuk mendapatkan nilai $\hat{M}c_{(T)}$ saat Y_{76} hingga ke Y_{82} dengan menggunakan persamaan di atas dapat dilakukan dan hasil dari persamaan tersebut akan dibandingkan dengan data riil. Hasil prediksi *output* dengan menggunakan model *Duane* diatas dilambangkan sebagai $\overline{MTBF}_{(T)}$, sedangkan data riil dilambangkan sebagai $MTBF_{(T)}$. Pada Tabel 7 akan ditampilkan hasil perbandingan antara nilai data riil dengan hasil *output* dengan menggunakan persamaan di atas.

Tabel 7. Perbandingan target ($MTBF_{(T)}$) dengan *output* ($\overline{MTBF}_{(T)}$)

| T | $MTBF_{(T)}$ | $\overline{MTBF}_{(T)}$ | $Error\ MTBF_{(T)}$ |
|------------------|--------------|-------------------------|---------------------|
| 76 | 192,000 | 228,200 | -36,200 |
| 77 | 72,000 | 228,800 | -156,800 |
| 78 | 168,000 | 229,400 | -61,400 |
| 79 | 24,000 | 229,900 | -205,900 |
| 80 | 48,000 | 230,400 | -182,400 |
| 81 | 144,000 | 230,900 | -86,900 |
| 82 | 48,000 | 231,400 | -183,400 |
| <i>MSE MTBF</i> | | | 20931,203 |
| <i>RMSE MTBF</i> | | | 144,676 |

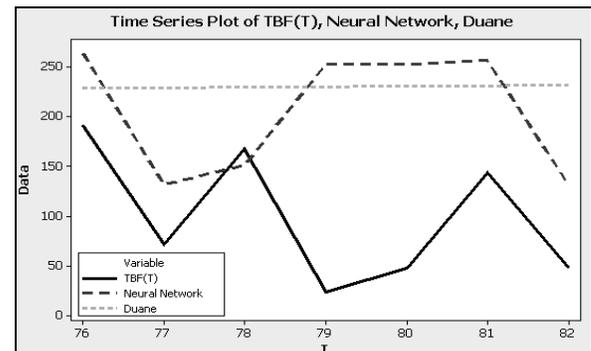
Untuk mempermudah pembacaan perbandingan antara target dengan *output*, maka akan ditampilkan dalam bentuk plot pada Gambar 10 berikut



Gambar 10. Plot perbandingan antara target ($MTBF_{(T)}$) dengan *output* ($\overline{MTBF}_{(T)}$)

Hasil dari Gambar 10 menunjukkan bahwa hasil prediksi model *Duane* memiliki pola naik secara monoton seiring dengan bertambahnya waktu yang digambarkan dengan garis putus-putus, sedangkan data riil digambarkan dengan garis lurus yang terhubung pada tiap satuan waktu.

Sebagai tahap evaluasi, maka setelah hasil *output* pada kedua model didapatkan berikut pada Gambar 11 akan ditampilkan hasil *output* perbandingan antara kedua model, di mana prediksi model *Neural Network* ditampilkan dengan garis putus-putus besar, sedangkan model *Duane* digambarkan dengan garis lurus putus-putus kecil, dan data riil ditampilkan dengan garis lurus.



Gambar 11. Hasil perbandingan *output* prediksi model *Neural Network* dan Model *Duane*

Hasil perbandingan nilai *RMSE* antara data *training* dan data testing pada kedua model yang akan dibandingkan ditampilkan pada Tabel 8 berikut.

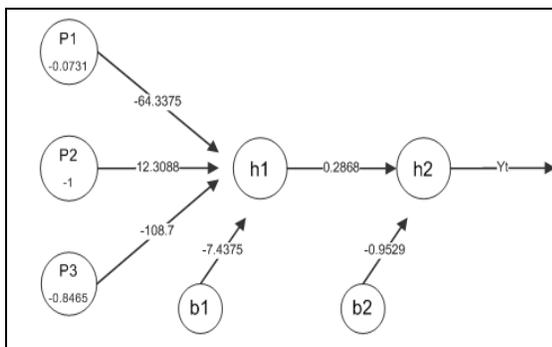
Tabel 8. Hasil perbandingan metode peramalan pada data kerusakan *equipment*

| Metode peramalan | RMSE | |
|------------------------|---------------|--------------|
| | Data training | Data testing |
| Neural Network (1,1,1) | 272,926 | 132,440 |
| Duane | 259,903 | 144,676 |

Berdasarkan hasil pada Tabel 8 dapat dijelaskan bahwa model *NN(1,1,1)* merupakan model yang paling baik dalam memprediksi suatu kerusakan pada kasus data kerusakan *equipment*. Hal ini ditunjukkan oleh nilai *RMSE* pada data testing model *Neural Network* yang lebih kecil, karena hasil pada data testing lebih diutamakan.

Hasil pada data kerusakan software

Dengan menggunakan strategi permodelan untuk model *Neural Network* yang sama seperti pada kasus kerusakan *equipment* sebelumnya, maka hasil penentuan variabel *lag input* yang optimal melalui uji *trial and error* adalah sebanyak 3 yaitu Y_{t-5} , Y_{t-8} , dan Y_{t-10} yang dilambangkan sebagai P1, P2, dan P3 dengan jumlah neuron di lapisan tersembunyi sebanyak 1 yang dilambangkan sebagai *h1*, maka model *Neural Network* dapat disimbolkan sebagai *NN(3,1,1)* di mana Gambar interpretasi *NN(3,1,1)* disajikan pada Gambar 12.



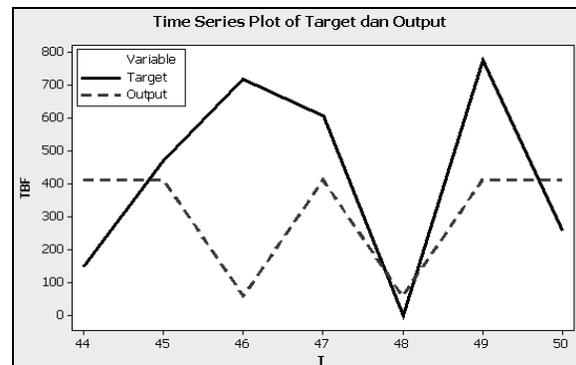
Gambar 12. Interpretasi *perceptron* dengan 3 input

Dengan cara perhitungan yang sama dengan kasus sebelumnya, maka hasil perhitungan *output* antara model *Neural Network* dengan data riil bisa didapatkan, di mana hasil perbandingan tersebut akan ditampilkan pada Tabel 9 berikut.

Tabel 9. Hasil Perbandingan nilai antara target dengan *output* jaringan kasus kerusakan *software*

| T | Target | Output | Error |
|------|---------|---------|------------|
| 44 | 149,000 | 411,065 | -262,065 |
| 45 | 469,000 | 411,065 | 57,935 |
| 46 | 716,000 | 57,994 | 658,006 |
| 47 | 604,000 | 411,065 | 192,935 |
| 48 | 0,000 | 57,994 | -57,994 |
| 49 | 774,000 | 411,065 | 362,935 |
| 50 | 256,000 | 411,065 | -155,065 |
| MSE | | | 100194,367 |
| RMSE | | | 316,535 |

Untuk mempermudah pembacaan perbandingan antara target dengan *output*, maka pada Gambar 13 berikut disajikan dalam bentuk plot.



Gambar 13. Perbandingan antara target dengan *output* kasus kerusakan *software*

Hasil plot prediksi model *Neural Network* pada tiap satuan waktu memiliki pola yang berfluktuasi naik, dan turun yang digambarkan dengan garis putus-putus, untuk data riil digambarkan dengan garis lurus.

Dengan cara yang sama pada kasus sebelumnya, maka persamaan model *Duane* yang sesuai untuk kasus kerusakan *Software* adalah

$$\hat{M}c_{(T)} = 26,049 \times T^{1,68} \quad (7)$$

Dengan persamaan baru tersebut, maka perhitungan untuk mendapatkan nilai prediksi model *Duane* akan di tampilkan pada Tabel 10 beserta dengan data riil atau yang dikenal sebagai target.

Tabel 10. Perbandingan antara data riil dengan *output* model *Duane*

| <i>T</i> | <i>Target</i> | <i>Prediksi</i> | <i>Error</i> |
|-------------|---------------|-----------------|--------------|
| 44 | 149,000 | 569,210 | -420,210 |
| 45 | 469,000 | 578,070 | -109,070 |
| 46 | 716,000 | 586,880 | 129,120 |
| 47 | 604,000 | 595,620 | 8,380 |
| 48 | 0,000 | 604,300 | -604,300 |
| 49 | 774,000 | 612,920 | 161,080 |
| 50 | 256,000 | 621,480 | -365,480 |
| <i>MSE</i> | | | 104273,685 |
| <i>RMSE</i> | | | 322,914 |

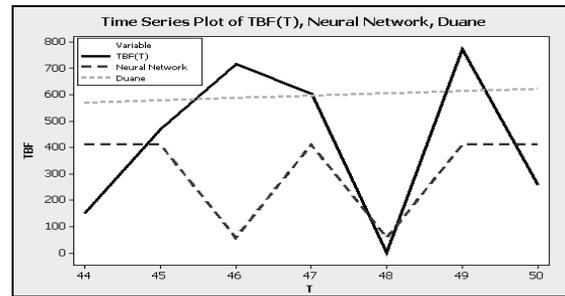
Untuk mempermudah pembacaan perbandingan antara target dengan *output* di atas, maka akan ditampilkan dalam bentuk plot pada Gambar 14 berikut.



Gambar 14. Plot perbandingan antara data riil dengan hasil *output* persamaan model *Duane*

Hasil prediksi *output* model *Duane* memiliki pola naik secara monoton seiring dengan bertambahnya waktu yang tidak mengikuti pola data riil yang berfluktuasi naik maupun turun. Pola model *Duane* digambarkan dengan garis lurus putus-putus, sedangkan pada data riil digambarkan dengan garis lurus.

Sebagai tahap evaluasi, maka hasil *output* model *Neural Network*, dan model *Duane* akan ditampilkan pada Gambar 15 berikut. Model *Neural Network* digambarkan dengan garis penghubung putus-putus besar, sedangkan model *Duane* digambarkan dengan garis putus-putus kecil, dan data riil digambarkan dengan garis lurus bersambung.



Gambar 15. Hasil perbandingan *output* prediksi model *Neural Network* dan Model *Duane*

Hasil perbandingan *RMSE* antara data *training* dan data *testing* untuk kedua model yang dibandingkan akan ditampilkan pada Tabel 11 sebagai berikut.

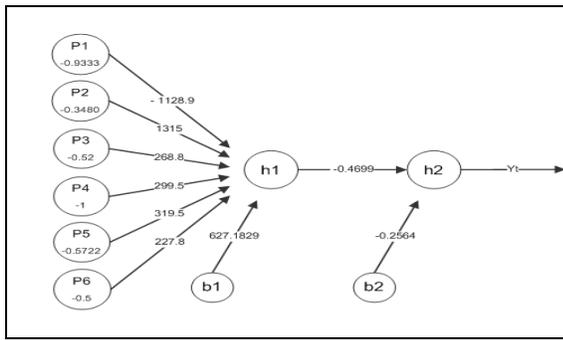
Tabel 11 Hasil perbandingan metode peramalan pada data kerusakan *software*

| Metode peramalan | <i>RMSE</i> | |
|-------------------------------|----------------------|---------------------|
| | Data <i>training</i> | Data <i>testing</i> |
| <i>Neural Network</i> (3,1,1) | 563,424 | 316,535 |
| <i>Duane</i> | 468,605 | 322,914 |

Hasil pada Tabel 11 menunjukkan bahwa model *NN(3,1,1)* merupakan model yang paling baik untuk peramalan data kerusakan *software*. Hal ini ditunjukkan oleh nilai *RMSE* yang paling kecil pada data *testing*, karena hasil pada data *testing* lebih diutamakan.

Hasil pada data kerusakan mesin fotokopi

Dengan cara, dan strategi yang sama pada kasus-kasus kerusakan sebelumnya dalam model *Neural Network*, maka hasil penentuan variabel *lag* yang optimal melalui uji *trial and error* didapatkan *input* yang optimal adalah sebanyak 6 yang dilambangkan sebagai P1, P2, P3, P4, P5 dan P6, sedangkan hasil dari uji *trial and error* penentuan jumlah neuron yang optimal di lapisan tersembunyi adalah sebanyak 1 yang dilambangkan sebagai *h1*, dan bobot bias pada lapisan pertama dilambangkan sebagai *b1*, sehingga model *Neural Network* dapat disimbolkan sebagai *NN(6,1,1)* contoh gambar interpretasi *perceptron* untuk kasus kerusakan mesin fotokopi ditampilkan pada Gambar 16.



Gambar 16. Interpretasi perceptron dengan 6 input

Dengan cara perhitungan model *Neural Network* yang sama pada kasus-kasus sebelumnya, maka hasil prediksi model *Neural Network* pada saat Y_{86} hingga Y_{92} bisa didapatkan yang akan ditampilkan pada Tabel 12 beserta dengan data riil atau yang dikenal sebagai target.

Tabel 12. Hasil Perbandingan nilai antara target dengan *output* jaringan kasus kerusakan mesin fotokopi

| T | Target | Output | Error |
|------|--------|--------|---------------|
| 86 | 36,000 | 9,203 | 26,797 |
| 87 | 20,000 | 9,167 | 10,833 |
| 88 | 64,000 | 31,999 | -31935,000 |
| 89 | 22,000 | 9,206 | 12,794 |
| 90 | 9,000 | 26,900 | -17,900 |
| 91 | 17,000 | 7,150 | 9,850 |
| 92 | 10,000 | 9,708 | 0,292 |
| MSE | | | 145692234,519 |
| RMSE | | | 12070,304 |

Guna mempermudah pembacaan perbandingan antara target dengan *output*, maka pada Gambar 17 berikut akan ditampilkan gambar plot perbandingan antara target dengan *output* dari Tabel 12 di atas.



Gambar 17. Perbandingan antara target dengan *output* prediksi model *Neural Network*

Hasil *output* prediksi model *Neural Network* memiliki pola yang berfluktuasi naik maupun turun yang digambarkan dengan garis penghubung putus-putus, sedangkan untuk data riil atau target digambarkan dengan garis lurus.

Dengan cara yang sama pada kasus kerusakan sebelumnya, maka persamaan model *Duane* yang terbaik untuk kasus kerusakan mesin fotokopi adalah

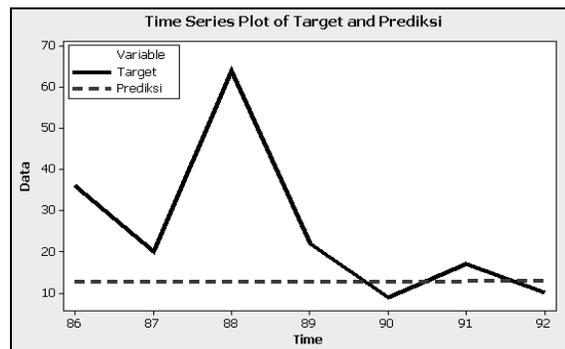
$$\hat{M}_{C(T)} = 8,248 \times T^{1,08} \quad (8)$$

Dengan persamaan baru tersebut, maka prediksi model *Duane* pada saat Y_{86} hingga Y_{92} dapat dilakukan di mana hasil perhitungan tersebut akan ditampilkan Tabel 13 beserta dengan data riil atau yang dikenal sebagai target.

Tabel 13. Perbandingan antara data riil dengan *output* model *Duane*

| T | Target | Prediksi | Error |
|------|--------|----------|---------|
| 86 | 36,000 | 12,715 | 23,285 |
| 87 | 20,000 | 12,727 | 7,273 |
| 88 | 64,000 | 12,739 | 51,261 |
| 89 | 22,000 | 12,751 | 9,250 |
| 90 | 9,000 | 12,762 | -3,762 |
| 91 | 17,000 | 12,773 | 4,227 |
| 92 | 10,000 | 12,785 | -2,785 |
| MSE | | | 478,297 |
| RMSE | | | 21,870 |

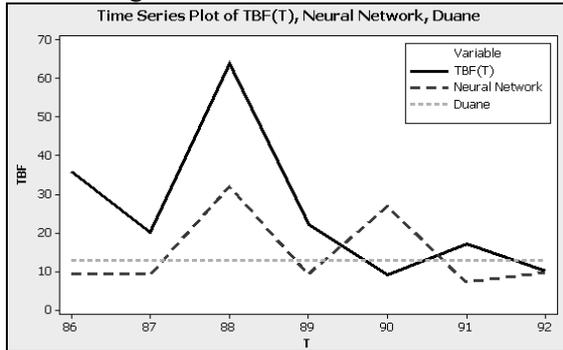
Untuk mempermudah pembacaan perbandingan antara target dengan prediksi model *Duane* diatas, maka pada Gambar 18 berikut akan ditampilkan dalam bentuk gambar plot perbandingan sebagai berikut.



Gambar 18. Perbandingan antara *output* data riil dengan model *Duane*

Hasil prediksi model *Duane* memiliki pola monoton naik secara perlahan yang digambarkan dengan garis lurus putus-putus, sedangkan data riil atau target digambarkan dengan garis yang berfluktuasi naik maupun turun pada tiap waktu.

Sebagai tahap evaluasi, maka hasil *output* antara model *Neural Network*, dan model *Duane* akan ditampilkan pada Gambar 19 berikut. Hasil prediksi model *Neural Network* digambarkan dengan garis penghubung putus-putus besar, sedangkan model *Duane* digambarkan dengan garis putus-putus kecil, dan data riil digambarkan dengan garis lurus bersambung.



Gambar 19. Hasil perbandingan *output* prediksi model *Neural Network* dan Model *Duane* untuk kerusakan mesin fotokopi

Hasil perbandingan *RMSE* antara data *training*, dan data testing untuk kedua model yang dibandingkan akan ditampilkan sebagai berikut.

Tabel 14. Hasil perbandingan metode peramalan pada data kerusakan mesin fotokopi

| Metode peramalan | <i>RMSE</i> | |
|-------------------------------|----------------------|--------------|
| | Data <i>training</i> | Data testing |
| <i>Neural Network</i> (6,1,1) | 12,064 | 18,673 |
| <i>Duane</i> | 13,527 | 21,870 |

Berdasarkan hasil pada Tabel 14 dapat dijelaskan bahwa model *NN(6,1,1)* merupakan model yang paling baik untuk peramalan data kerusakan mesin fotokopi. Hal ini ditunjukkan oleh nilai *RMSE* yang paling kecil baik pada data *training* maupun data testing, namun dalam hal ini data testing lebih diutamakan.

KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan maka dapat diambil kesimpulan sebagai berikut:

1. Model *Neural Network* terbaik yang diperoleh pada data kasus kerusakan *equipment* di PT. Petro Kimia Gresik adalah menggunakan jumlah *neuron* optimal sebanyak 1 di mana hasil prediksinya membentuk pola naik dan turun (nonlinier), sedangkan pada data kerusakan *software* dan kerusakan sistem mesin fotokopi dari *paper*

Xie dan Ho (1999) menunjukkan jumlah *neuron* optimal yang digunakan adalah sebanyak 1 di mana hasil prediksinya membentuk pola naik dan turun (nonlinier).

2. Model *Duane* terbaik yang diperoleh pada data kasus kerusakan *equipment* di PT. Petro Kimia Gresik dengan T adalah urutan terjadinya kerusakan yang memiliki nilai $\alpha > 1$ di mana hasil prediksinya membentuk pola yang bersifat naik secara monoton (linier), sedangkan pada data kerusakan *software*, dan kerusakan sistem mesin fotokopi dari *paper* Xie dan Ho (1999) dengan T adalah urutan terjadinya kerusakan yang memiliki nilai $\alpha < 1$ membentuk pola prediksi yang bersifat naik secara monoton (linier), dimana semakin besar nilai α , maka pola prediksi akan membentuk eksponensial naik secara monoton, sedangkan bila nilai $\alpha > 1$ akan membentuk eksponensial turun secara monoton.
3. Hasil perbandingan pada kasus data riil menunjukkan bahwa model *Neural Network* dapat memprediksi kerusakan lebih baik dari model *Duane* dikarenakan model *Neural Network* sangat tepat digunakan pada data kerusakan yang memiliki pola naik dan turun (nonlinier), sedangkan model *Duane* lebih tepat digunakan pada suatu data kerusakan yang memiliki pola *trend* naik maupun turun yang monoton (linier).

DAFTAR PUSTAKA

- [1] Xie, M. dan Ho, S. L., *Analysis of Repairable System Failure Data Using Time Series Models*. *Journal of Quality in Maintenance Engineering*, Vol. 5 No. 1, hlm. 50-61, 1999
- [2] Ho, S. L., Xie, M., dan Goh, T. N., A Comparative Study of Neural Network and Box-Jenkins ARIMA Modeling in Time Series Prediction, *Computers and Industrial Engineering*, Vol. 42, hlm. 371-375, 2002
- [3] Kusumadewi, S., *Membangun Jaringan Syaraf Tiruan (Menggunakan MATLAB dan Excel Link)*, Edisi Pertama, hlm. 49-215, Penerbit Graha Ilmu, Yogyakarta, 2004
- [4] Anonim, *Model Duane*, <http://www.weibull.com/RelGrowthWeb/reliabilitygrowth.htm>, diakses 17 Mei 2008