

PEMILIHAN ARSITEKTUR OPTIMAL MODEL NN DENGAN METODE KONTRIBUSI INCREMENT

Triastuti Wuryandari dan Budi Warsito
Jurusan Matematika FMIPA UNDIP
Jl. Prof. H. Soedarto, S.H, Semarang 50275

Abstract. *Neural Network* is an information processing system that has certain characteristic in common with biological neural network. In development *NN* has been many applied in several surface, one of them is for forecasting. For the best application of *NN*, architecture has determined. One of methode to get optimal architecture *NN* is incremental contribution methods. This methods will to determine the size of hidden and input cell in the network with excluding respectively. One of the unit cell with a low incremental contribution will be exclusion from network. The result shows that the incremental contribution methods is capable reducing the size of the network is propozed, so getting optimal architecture from network.

Keywords: Time series, *NN*, Optimal Architecture, Contribution Incremental

1. PENDAHULUAN

Artificial Neural Network (*ANN* atau *NN* saja) adalah jaringan saraf buatan yang merupakan model sistem komputasi yang bekerja seperti sistem jaringan saraf biologi [8]. Untuk pemilihan model peramalan terbaik pada *NN*, lebih dahulu ditentukan arsitekturnya. Dalam pemilihan arsitektur optimal, metode untuk menentukan ukuran jumlah unit input dan unit tersembunyi perlu untuk dipilih [1]. Salah satu metode yang dapat diterapkan untuk menentukan ukuran jaringan adalah metode kontribusi increment. Dengan metode ini akan diperoleh hasil yang ditunjukkan oleh jaringan dengan mengeluarkan satu unit tertentu dari jaringan secara bergantian. Unit tertentu yang mempunyai kontribusi kecil akan dikeluarkan dari jaringan, sehingga diperoleh jumlah unit input dan unit tersembunyi yang optimal [5].

Tulisan ini membahas pemilihan arsitektur optimal pada arsitektur multi-layer perceptron dengan satu lapisan tersembunyi dan satu unit output. Adapun konsep belajar yang digunakan adalah algoritma belajar perambatan balik (*back-propagation*) momentum yang merupakan variasi dari algoritma belajar *backpropagation* standar dan bobotnya diestimasi dengan metode optimasi *gradient descent*.

2. MODEL NEURAL NETWORK

NN merupakan sekumpulan elemen pemroses sederhana yang disebut neuron [4]. Neuron-neuron dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut lapisan neuron (*neuron layer*). Neuron-neuron dihubungkan saluran penghubung dan inputnya akan dirambatkan melalui lapisan tersembunyi (*hidden layer*). Setiap neuron berhubungan dengan neuron lainnya untuk mentransformasikan input yang diterimanya. Besarnya informasi yang digunakan disebut bobot koneksi [3]. *NN* menentukan bobot koneksi antar neuron dengan jalan melakukan pelatihan (*training*). Pelatihan dengan Pengarahan (*Supervised Training*) memberikan pola-pola output yang diharapkan dengan memasang setiap vektor atau pola input dengan vektor target. Setiap input akan dihitung nilai outputnya dan dibandingkan dengan vektor target. Perbedaan antara output yang dihasilkan dengan target disebut error yang digunakan untuk menentukan bobot koneksi yang baru. Hasil penjumlahannya dibandingkan dengan batas ambang tertentu melalui fungsi aktivasi. Fungsi aktivasi yang sering dipakai adalah fungsi *Threshold*, fungsi Sigmoid Logistik dan fungsi Identitas. Susunan dari neuron-neuron dalam lapisan dan pola keterhubungannya dalam dan antar lapisan

disebut sebagai arsitektur jaringan. Kadang-kadang dalam jaringan ditambahkan sebuah unit input yang nilainya sama dengan satu yang disebut bias. Bias berfungsi untuk mengubah nilai *threshold* menjadi nol. *Threshold* berperan sebagai penimbang dalam suatu hubungan dari sebuah unit tertentu.

Perceptron merupakan jaringan lapis tunggal yang hanya ada sebuah unit neuron pada lapisan outputnya. *Perceptron* terdiri dari beberapa unit input yang dihubungkan dengan sebuah unit output oleh suatu unit penghubung. Perubahan bobot dan bias yang menghubungkan unit input dan unit output dilakukan melalui pembelajaran perceptron, dimana setiap input yang masuk dalam jaringan output akan dibandingkan dengan target (error). Jika terdapat perbedaan maka bobot akan dimodifikasi. Pelatihan terus berlanjut sampai tidak terjadi error. Jika antara input dan lapisan tersembunyi dimasukkan bobot dan digunakan fungsi aktivasi non linier, maka model yang dihasilkan disebut model jaringan *multilayer perceptron* (MLP) [7]. Dalam MLP terdapat lapisan tersembunyi yang terletak diantara lapisan input dan lapisan output [8]. Dalam mencari bobot optimalnya, jaringan dengan fungsi aktivasi non linear pada unit tersembunyi mempunyai kemampuan lebih dibanding jaringan yang mengandung lapisan input dan output saja. Arsitektur jaringan MLP ditunjukkan pada Gambar 1.

Back Propagation (BP) merupakan algoritma pembelajaran yang terawasi dan digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan unit-unit yang ada pada lapisan tersembunyi. Algoritma *BP* menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Untuk pelatihan *BP* fungsi sigmoid adalah pilihan yang paling tepat [3], karena fungsi sigmoid memiliki nilai maksimum 1 (satu) dan untuk pola yang targetnya lebih besar

1 (satu), data harus ditransformasi sehingga semua polanya memiliki range yang sama dengan fungsi sigmoid yang dipakai.

BP melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola serta kemampuan memberikan respon yang benar terhadap pola input yang mirip dengan pola input yang dipakai selama pelatihan. Pelatihan *BP* menggunakan metode pencarian titik minimum untuk mencari bobot dengan error yang minimum. Algoritma pelatihan *BP* meliputi tiga tahap. Pertama adalah tahap maju, yaitu pola input dihitung maju mulai dari lapisan input hingga lapisan output dengan menggunakan fungsi aktivasi yang ditentukan.

Kedua adalah tahap balik, yaitu selisih antara output dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasi balik, dimulai dari garis yang berhubungan langsung dengan unit di lapisan output. Ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi. Selama propagasi maju, sinyal input (x_i) dipropagasikan ke lapisan tersembunyi menggunakan fungsi aktivasi. Output dari setiap unit tersembunyi (z_j) tersebut dipropagasikan maju lagi ke lapisan tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan output jaringan (y_k). Berikutnya, output jaringan (y_k) dibandingkan dengan target yang harus dicapai (t_k), $\delta_k = (t_k - y_k)$ adalah kesalahan yang terjadi. Satu siklus pelatihan yang melibatkan semua pola disebut *epoch* [8]. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka *epoch* dihentikan. Tetapi apabila kesalahan lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi. Pada Propagasi mundur, dihitung faktor δ_k , $k=1,2,\dots,m$ untuk mendistribusikan kesalahan di y_k ke semua lapisan tersembunyi yang terhubung langsung dengan y_k . Notasi δ_k juga dipakai untuk mengubah bobot garis yang

berhubungan langsung dengan unit output. Dengan cara yang sama, dihitung faktor δ_j di setiap unit tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari lapisan tersembunyi di lapisan dibawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit input dihitung. Pada tahap perubahan bobot, setelah semua faktor δ dihitung, bobot semua garis dimodifikasi secara bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di lapisan atasnya.

Pelatihan BP dilakukan untuk mengatur bobot, sehingga pada akhir pelatihan diperoleh bobot-bobot yang baik. Bobot-bobot dalam jaringan diestimasi dengan algoritma pelatihan optimisasi. Selama proses pelatihan optimisasi bobot-bobot diatur secara iteratif untuk meminimumkan fungsi MSEnya, dimana fungsi ini akan mengambil kuadrat error yang terjadi antara output dan target. Dalam tulisan ini digunakan penurunan gradien (*gradien descent*) yang merupakan metode yang paling sederhana untuk merubah bobot koneksi [2]. Pada dasarnya algoritma pelatihan BP akan mengubah bobot dan bias pada arah dimana fungsi kinerja (MSE) menurun paling cepat, yaitu dalam arah negatif gradiennya [8]. Dengan metode ini

kesalahan lokal tiap sel dapat dilihat sebagai bagian yang berkontribusi menghasilkan kesalahan total pada lapisan output. Apabila kesalahan pada lapisan output dapat dipropagasikan masuk ke lapisan tersembunyi, maka kesalahan lokal sel-sel pada lapisan tersebut dapat dihitung. Dengan metode gradien decent perubahan bobot dan bias dihitung sebagai berikut.

- Pada lapisan tersembunyi berlaku:

$$z_in_j = \sum_{i=1}^n x_i w_{jk} + b1_j . \tag{2.1}$$

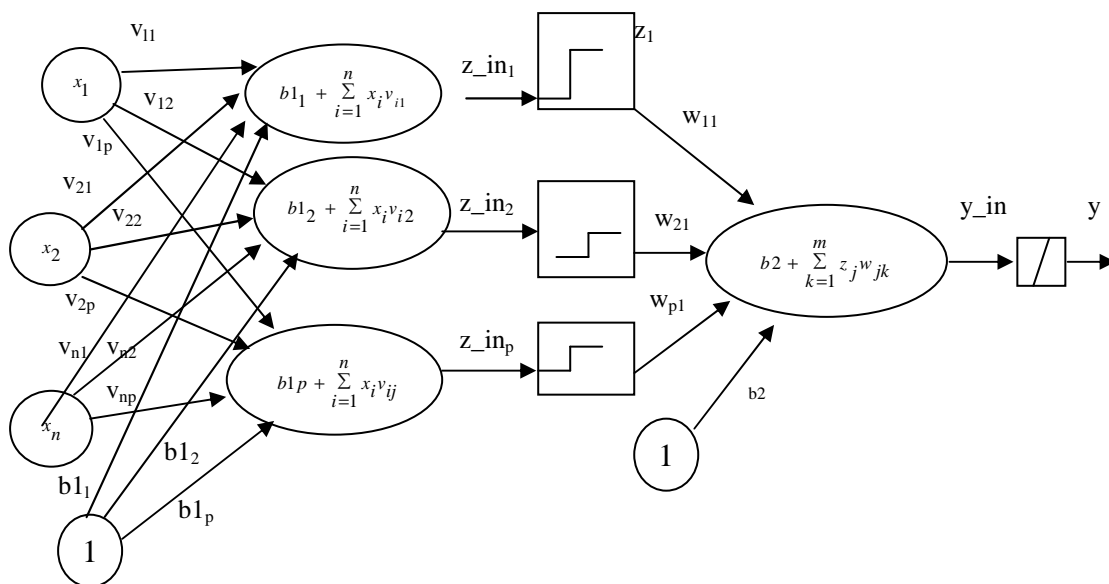
Sehingga dengan fungsi sigmoid ,

$$\begin{aligned} z_j &= f(z_in_j) \\ &= \frac{1}{1 + e^{-z_in_j}} \\ &= \frac{1}{1 + e^{-\sum_{k=1}^n x_i w_{jk} + b1_j}} \end{aligned} \tag{2.2}$$

Dengan z_j adalah output pada lapisan tersembunyi, w_{jk} bobot dari lapisan tersembunyi j ke lapisan output k, $b1_j$ bias pada lapisan tersembunyi.

- Pada lapisan output berlaku:

$$y_in = \sum_{j=1}^p w_j z_j + b2 . \tag{2.3}$$



Gambar 1. Arsitektur MLP dengan n unit input, p unit tersembunyi, dan 1 unit output.

Sehingga

$$y = f(y_{in}) = y_{in} = \sum_{j=1}^p w_j z_j + b_2 \quad (2.4)$$

Pada lapisan output, kesalahan pada unit output didefinisikan $\delta_k = (y_k - t_k)$ (2.5)

Dengan δ_k adalah kesalahan pada unit output ke-k, y_k keluaran yang diinginkan, t_k keluaran actual.

Agar persamaan perubahan bobot dapat dipecahkan, maka fungsi aktivasi yang digunakan harus bersifat diferensiable. Bobot-bobot koneksi dari sel-k ke sel-j (w_{jk}) diubah sebanding dengan negatif gradien fungsi kesalahan terhadap perubahan bobot:

$$\Delta \cdot w_j = \eta \cdot \left(-\frac{\partial E}{\partial w_j} \right),$$

dengan

$$E = \frac{1}{2} \sum_{k=1}^m (y - t)^2 \quad (2.6)$$

dan Δw_j adalah perubahan bobot sel-j ke sel-k, η adalah suatu konstanta pembelajaran ($0 \leq \eta \leq 1$).

3. PEMILIHAN ARSITEKTUR OPTIMAL MODEL NN DENGAN METODE KONTRIBUTSI INCREMENT

Dalam tulisan ini dibahas bentuk NN dengan satu lapisan tersembunyi dan satu unit output dengan persamaan umum prediksi untuk menghitung suatu ramalan dari x_t (output) dengan menggunakan pengamatan-pengamatan masa lalu yang terpilih x_{t-1}, \dots, x_{t-k} sebagai input dapat dituliskan dalam bentuk:

$$\hat{X}_t = \varphi_o \left\{ w_{co} + \sum_i w_{ho} \varphi_h \left(w_{ch} + \sum_i w_{ih} x_{t-k} \right) \right\}, \quad (3.1)$$

dengan w_{ch} adalah bobot hubungan antara input konstan dan lapisan tersembunyi w_{co} adalah bobot hubungan langsung antara input konstan dan output, w_{ih} dan w_{ho} masing-masing bobot hubungan antara input dan unit tersembunyi dan antara unit tersembunyi dengan output φ_h dan φ_o adalah

fungsi aktivasi pada lapisan tersembunyi dan output.

Notasi untuk model NN adalah NN (j_1, \dots, j_k, h) yang menyatakan NN dengan input lag j_1, \dots, j_k dan h unit dalam satu lapisan tersembunyi. Jumlah parameter (bobot) pada model NN dengan satu lapisan tersembunyi dirumuskan dengan $p = (n_i + 2)n_u + 1$ dengan n_i adalah jumlah input dan n_u menyatakan jumlah unit tersembunyi.

Nilai pada lag waktu t adalah nilai ramalan yang menggunakan nilai pada lag waktu 1 dan 12. Selanjutnya lag-lag tersebut disebut input dan nilai ramalannya disebut output. Selain lag-lag tersebut, ada suatu inputan konstan yang nilainya selalu sama dengan satu disebut sebagai bias yang dihubungkan secara langsung dengan unit tersembunyi dan unit output dengan bobot koneksi masing-masing b_{1j} ($j=1,2$) dan b_2 . Masing-masing unit input terhubung dengan setiap unit tersembunyi yang ada, dan setiap unit tersembunyi tersebut terhubung dengan output dengan bobot koneksi masing-masing v_{ij} dan w_{jk} . Bobot yang digunakan dalam model NN diestimasi dari data dengan metode optimisasi yaitu dengan meminimumkan jumlah kuadrat error peramalan dari sampel, biasanya ditulis $SSE = \sum_t (\hat{x}_t - x_t)^2$.

Nilai numerik masing-masing unit dihitung sebagai berikut. Misalkan fungsi linier dari input-input adalah $\sum v_{ij} x_i$ dimana v_{ij} menyatakan bobot dari hubungan antara input x_i dengan unit tersembunyi ke-j. Hasil penjumlahan linier dari seluruh unit input yang telah dikalikan dengan masing-masing bobot koneksinya adalah $v_j = \sum v_{ij} x_i$, kemudian v_j ditransformasi ke suatu fungsi aktivasi yang telah ditentukan. Fungsi aktivasi yang biasanya digunakan adalah fungsi sigmoid $z_j = 1 / \{ 1 + \exp(-v_j) \}$ yang memberikan nilai dalam range (0,1).

Bobot awal mempengaruhi apakah jaringan mencapai titik minimum lokal atau global terhadap nilai error, dan kecepatan konvergensinya. Bobot yang terlalu kecil menyebabkan perubahan bobotnya

menjadi sangat kecil, sehingga menyebabkan proses pelatihan berjalan sangat lambat. Bobot awal yang terlalu besar menyebabkan nilai turunan fungsi aktivasinya menjadi sangat kecil sehingga dalam BP, bobot dan bias diisi dengan bilangan acak kecil [8].

Secara umum suatu jaringan NN disusun atas 3 unit pemrosesan yaitu unit input, unit tersembunyi dan unit output. Unit-unit pemrosesan pada tiap-tiap lapisan dihubungkan melalui jaringan dimana masing-masing ditandai dengan suatu bobot untuk menggambarkan kekuatan jaringan tersebut. Dalam model NN, tiga hal yang harus ditekankan adalah pemilihan variabel input; pemilihan arsitektur dan fungsi aktivasi; serta kriteria seleksi model [3]. Salah satu metode penentuan jumlah unit input dan unit tersembunyi adalah metode Kontribusi Increment yang dikembangkan [5]. Metode ini termasuk dalam proses "General to Specific" yaitu prosedur yang dimulai dari model yang besar (komplek) dan kemudian menerapkan suatu algoritma untuk mereduksi jumlah unit dalam suatu jaringan.

Metode Kontribusi Increment digunakan untuk mendapatkan hitungan yang ditunjukkan oleh jaringan dengan salah satu unit dikeluarkan secara bergantian, sebuah unit dengan kontribusi increment yang kecil akan dikeluarkan dari jaringan, sehingga diperoleh jumlah unit yang optimal. Sebuah kandidat alami untuk menyatakan tampilan jaringan adalah dengan menggunakan kuadrat dari koefisien korelasi dari Y dan \hat{Y} sebagai berikut.

$$R^2 = \frac{(\hat{Y}'Y)^2}{(Y'Y)(\hat{Y}'\hat{Y})}, \quad (3.1)$$

dengan \hat{Y} adalah vektor dari titik output jaringan dan Y adalah nilai target

Tampilan (*performance*) suatu jaringan dengan menghilangkan salah satu unit dapat diukur dengan cara serupa. Untuk langkah pertama, jika kontribusi dari unit tersembunyi diambil nilai nol (satu unit pada lapisan tersembunyi dikeluarkan)

maka jaringan akan menghasilkan output \hat{Y}_{-h} dengan error

$$e_{-h} = Y - \hat{Y}_{-h}, \quad (3.2)$$

dan pengurangan jaringan dapat dinyatakan dengan kuadrat dari koefisien korelasi R^2_{-h} antara Y dan \hat{Y}_{-h} dengan

$$R^2_{-h} = \frac{(\hat{Y}_{-h}'Y)^2}{(Y'Y)(\hat{Y}_{-h}'\hat{Y}_{-h})}. \quad (3.3)$$

Kontribusi increment dari unit pada lapisan tersembunyi (h) dapat diberikan sebagai suatu selisih sebagai

$$R^2_{inc} = R^2 - R^2_{-h}. \quad (3.4)$$

Jika nilai R^2_{inc} kecil untuk beberapa h jika dibandingkan dengan nilai lainnya maka unit tersembunyi tersebut pantas untuk dikeluarkan, tetapi sebaliknya jika nilai R^2_{inc} besar maka unit tersembunyi tersebut tetap harus dipertahankan atau dimasukkan ke dalam jaringan. Prosedur yang sama dapat digunakan untuk mereduksi jumlah unit input dari suatu jaringan. Dalam kasus ini $\{\hat{Y}_{-i}(t)\}$ adalah output jaringan yang memberikan estimasi parameter jaringan tanpa unit input ke-i. Jika kontribusi dari unit input ke-i yang diambil nol, kemudian pengurangan jaringan dapat dinyatakan dengan kuadrat dari koefisien R^2_{-i} antara Y dan \hat{Y}_{-i} dengan

$$R^2_{-i} = \frac{(\hat{Y}_{-i}'Y)^2}{(Y'Y)(\hat{Y}_{-i}'\hat{Y}_{-i})}. \quad (3.5)$$

Kontribusi dari unit input ke-i diukur dengan

$$R^2_{inc} = R^2 - R^2_{-i}. \quad (3.6)$$

Ukuran relatif dari kontribusi increment dalam R^2_{inc} dapat digunakan dalam mengetahui apakah unit input dapat dikeluarkan dari jaringan atau tidak.

Dalam metode ini selain melihat nilai R^2_{inc} nya dapat juga dilihat dari hasil tampilan grafiknya yaitu dengan melakukan perbandingan grafik dari output jaringan dan data pengamatan dengan hanya satu unit dikeluarkan dan semua unit lainnya dimasukkan. Jika satu unit dikeluarkan dan menghasilkan tampilan yang bagus maka unit tersebut pantas dikeluarkan dari jari-

ngan, tetapi jika memperlihatkan tampilan yang kurang bagus, maka unit tersebut ada kemungkinan untuk tetap dipertahankan atau dimasukkan ke dalam jaringan.

Suatu cara untuk melihat tampilan jaringan adalah dengan membandingkan grafik data output asli $\{t, Y(t)\}$ dengan estimasi NN $\{t, \hat{Y}(t)\}$. Dianggap suatu jaringan dengan unit tersembunyi ke-h, jika unit tersembunyi ke-h ini dikeluarkan, maka output yang dihasilkan jaringan adalah $\hat{Y}_{-h}(t)$. Grafik $\{t, \hat{Y}_{-h}(t)\}$ dibandingkan dengan grafik dari $\{t, Y(t)\}$, hasil perbandingannya dapat memberikan bukti adanya kontribusi dari unit tersembunyi ke-h dalam menjelaskan variabel $Y(t)$.

Dengan cara yang sama untuk unit input dapat diujikan. Misalkan $\{\hat{Y}_{-i}(t)\}$ adalah output jaringan dengan memasukkan semua unit input kecuali unit input ke-i, kemudian dengan memeriksa hasil tampilan antara grafik $\{t, Y(t)\}$ dan $\{t, \{\hat{Y}_{-i}(t)\}\}$ dapat ditunjukkan dengan je- las suatu perkecualian dari unit input ke-i.

4 . APLIKASI NN PADA DATA TIME SERIES

Diketahui data bulanan Indeks Harga Saham Gabungan (IHSG) di Bursa Efek Surabaya (BES) selama 5 tahun terakhir (Januari 1998 sampai Juli 2005), akan dibuat model peramalan NN dengan arsitektur optimal untuk memperkirakan besarnya nilai IHSG yang terjadi pada bulan depan/tahun depan. Digunakan metode kontribusi increment untuk penentuan arsitektur optimal dalam NN.

Dalam kasus ini lebih dahulu dicari jumlah unit tersembunyi dan unit input yang paling optimal dengan menetapkan arsitektur awal tertentu. Dari arsitektur awal ini akan dilakukan pemilihan jumlah unit tersembunyi dan unit input yang optimal dengan metode kontribusi increment. Dari arsitektur awal ini akan dicari nilai prediksi masing-masing arsitektur dengan mengeluarkan satu unit tersembunyi tertentu secara bergantian. Selanjutnya dihitung nilai kontribusi masing-masing arsitekturnya.

Kemudian dihitung nilai R_{inc}^2 nya dan hasil optimalisasi dapat dilihat dalam Tabel 1 dibawah. Dalam Tabel 1 dibawah diperlihatkan hanya unit tersembunyi ke-12 yang akan dimasukkan ke dalam jaringan, karena mempunyai kontribusi paling besar dibandingkan unit tersembunyi lainnya. Dalam langkah ini diperoleh jaringan dengan 12 unit input dan 1 unit tersembunyi, dinotasikan NN(1,2,...,12 ; 1). Langkah selanjutnya dihitung nilai kontribusi incrementnya dengan mengeluarkan satu unit input tertentu secara bergantian dimulai dari unit input ke-1 sampai ke-12. Dengan cara yang sama, optimalisasi lebih lanjut dapat dijalankan. Setelah didapatkan nilai prediksinya, selanjutnya dihiitung nilai kontribusi masing-masing arsitektur Kemudian dihitung nilai R_{inc}^2 nya dan hasil optimalisasinya dapat dilihat dalam Tabel 2 .

Dari Tabel 1 diperlihatkan hanya unit input ke-1 yang akan dimasukkan ke dalam jaringan, karena unit input ini mempunyai kontribusi lebih besar dibandingkan unit input lainnya. Karena hanya ada 1 unit input dan 1 unit tersembunyi yang mempunyai kontribusi incremental lebih besar, maka berdasarkan nilai kontribusi increment diperoleh jaringan dengan 1 unit input dan 1 unit tersembunyi dan dinotasikan dengan NN (1;1).

Pemilihan unit input dan unit tersembunyi optimal ini juga dapat dilihat pada tampilan Gambar 2. Dari gambar tersebut terlihat unit tersembunyi ke 12 mempunyai pola kurang bagus jika dibandingkan dengan unit tersembunyi lainnya. Berarti kontribusi dari unit tersembunyi ke-12 lebih tinggi dibandingkan unit tersembunyi lainnya, sehingga unit tersembunyi ke-12 akan tetap dimasukkan dalam jaringan, sedangkan unit tersembunyi yang lainnya akan dikeluarkan dari jaringan. Dan untuk inputnya, unit input ke-1 (x_{t-1}) mempunyai kontribusi lebih tinggi bandingkan dengan unit input lainnya. Sehingga unit input ke-1 akan tetap dipertahankan dalam jaringan, sedangkan unit input lainnya akan dikeluarkan dari jaringan.

Berdasarkan tampilan grafik Gambar 2, maka jaringan terakhir yang diperoleh adalah jaringan dengan 1 unit input dan 1 unit tersembunyi. Hasil ini sama dengan hasil yang diperoleh pada pemilihan jumlah unit input dan unit tersembunyi dengan membandingkan nilai kontribusi masing-masing unit input dan unit tersembunyi. Dari dua hasil ini dapat disimpulkan bahwa arsitektur optimal yang didapatkan adalah jaringan dengan 1 unit input dan 1 unit tersembunyi dan dinotasikan $NN(1;1)$. Secara matematis persamaan umum model peramalan data time series IHSG di BES dengan arsitektur $NN(1;1)$ dapat dituliskan

$$y = \hat{X}_t = w_{co} + \frac{w_{ho}}{1 + e^{-w_{ch} - w_{ih} \cdot X_{t-1}}} \quad (3.7)$$

Karena arsitektur optimal telah diperoleh, maka selanjutnya dilakukan peramalan dan dibandingkan dengan data observasi aktualnya. Digunakan arsitektur $NN(1;1)$ sebagai model peramalan data time series IHSG. Dari arsitektur optimal yang diperoleh, diketahui bahwa input jaringan adalah nilai IHSG pada lag ke-1

(x_{t-1}) . Sebagai targetnya adalah nilai IHSG pada bulan pertama setelah periode terakhir yaitu lag ke-13 (x_t). Dengan demikian akan terdapat rentang data dari Januari 1998 sampai Juli 2005. Dari data tersebut akan digunakan sebanyak 66 data untuk pelatihan sedangkan sisanya sebanyak 13 data digunakan sebagai data cek/data pengujian.

NN yang akan digunakan dalam pemodelan data time series untuk peramalan data IHSG ini terdiri dari satu lapisan input, satu lapisan tersembunyi, dan satu lapisan output. Pada lapisan input terdiri atas 1 unit input, yang mewakili input data IHSG pada lag ke-1. Pada lapisan tersembunyi terdiri atas 1 unit tersembunyi dengan fungsi aktivasinya adalah fungsi sigmoid, sedangkan pada lapisan output terdiri atas satu unit output dengan fungsi aktivasinya adalah fungsi identitas. Untuk melakukan peramalan digunakan NN dengan metode *backpropagation* momentum sebagai variasi dari *backpropagation* standar, dengan metode optimisasi *gradient descent*.

Tabel 1. Nilai kontribusi increment dengan mengeluarkan satu unit tersembunyi tertentu secara bergantian.

$R^2 = 0,9989$

	H₁	H₂	H₃	H₄	H₅	H₆
R²_{-h}	0,9989	0,9988	0,9987	0,9989	0,9989	0,9989
R²_{inc}	-2,4290.10 ⁻⁶	1,1196.10 ⁻⁴	2,3934.10 ⁻⁴	7,0169.10 ⁻⁵	6,674.10 ⁻⁵	5,9990.10 ⁻⁵

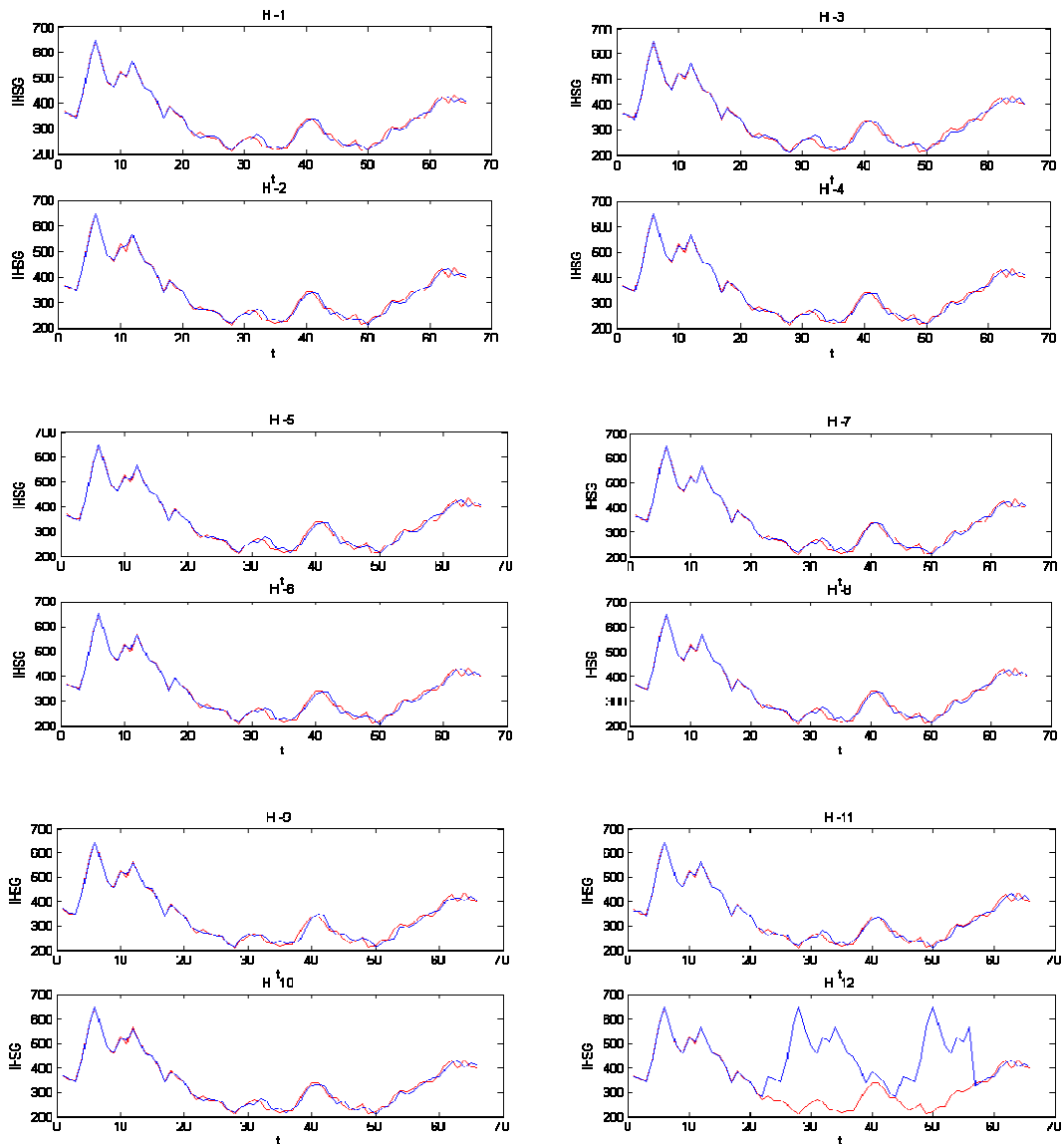
	H₇	H₈	H₉	H₁₀	H₁₁	H₁₂
R²_{-h}	0,9990	0,9989	0,9986	0,9988	0,9988	0,8095
R²_{inc}	-3,9883.10 ⁵	3,5993.10 ⁻⁵	3,0877.10 ⁻⁴	1,2014.10 ⁻⁴	1,7927.10 ⁻⁴	0,1085

Tabel 2. Nilai kontribusi increment dengan mengeluarkan satu unit input tertentu secara bergantian.

$R^2 = 0,9913$

	I₁	I₂	I₃	I₄	I₅	I₆
R²_{-i}	0,9810	0,9917	0,9912	0,9915	0,9918	0,9910
R²_{inc}	0,0102	-3,7364.10 ⁻⁴	1,0418.10 ⁻⁴	-2,4453.10 ⁻⁴	-4,9794.10 ⁻⁴	3,1084.10 ⁻⁴

	I₇	I₈	I₉	I₁₀	I₁₁	I₁₂
R²_{-i}	0,9905	0,9913	0,9919	0,9917	0,9912	0,9914
R²_{inc}	7,4241.10 ⁻⁴	1,0479.10 ⁻⁵	-6,5709.10 ⁻⁴	-4,7142.10 ⁻⁴	6,0580.10 ⁻⁵	-8,2730.10 ⁻⁵



Gambar 2. Grafik tampilan satu unit tersembunyi dikeluarkan dari jaringan secara bergantian pada $NN(1,2,\dots,12;12)$.

Keterangan: — Target / Nilai Aktual
— Output Jaringan

- Analisis Hasil Pelatihan dan Pengujian
 Pada data pelatihan dari bobot akhir yang didapat, dengan matlab diperoleh nilai MSE yang sudah mendekati nilai yang diharapkan sehingga dalam hal ini apabila diberi input yang mirip dengan pola yang dipakai selama pelatihan akan memberikan output yang masuk akal. Dari hasil pengujian diperoleh koefisien korelasi sebesar 0.9487 (mendekati 1), ini menunjukkan hasil yang baik untuk kecoco-

kan output jaringan dengan target. Adapun persamaan garis untuk kecocokan terbaik antara output jaringan dengan target pada data pengujian adalah $O=0.7068T + 123.1944$. Dari data pengujian diperoleh nilai MSE yang relatif kecil yaitu sebesar $1.7982 \cdot 10^{-003}$, sehingga pembentukan model peramalan data IHSIG di BES cukup berhasil.

5. PENUTUP

Berdasarkan hasil dari pembahasan dapat disimpulkan beberapa hal berikut ini.

1. Untuk mendapatkan arsitektur *NN* yang optimal sebagai model peramalan data time series dapat dilakukan dengan metode kontribusi increment.
2. Berdasarkan model akhir, ditunjukkan bahwa antara nilai kontribusi increment dan grafik saling memperkuat dalam menentukan arsitektur optimal suatu jaringan.
3. Dengan metode kontribusi increment didapatkan arsitektur *NN* yang paling optimal untuk peramalan data time series IHSG di BES dengan menggunakan fungsi aktivasi sigmoid logistik yaitu *NN* dengan satu unit input dan satu unit tersembunyi dan dinotasikan dengan *NN* (1;1). Sehingga dalam kasus ini dapat dituliskan persamaan umum model peramalan data IHSG sebagai berikut.

$$\hat{X}_t = w_{co} + \frac{w_{ho}}{1 + e^{-w_{ch} - w_{ih} \cdot X_{t-1}}}$$

4. Berdasarkan nilai kuadrat koefisien korelasi yang relatif besar dan perubahan MSE yang sangat kecil (10^{-5}), dapat dikatakan bahwa pembentukan model peramalan data IHSG di BES cukup berhasil.

6. DAFTAR PUSTAKA

- [1] Atok, R. M dan Suhartono (2000), *Perbandingan Antara Neural Network dan ARIMA Box-Jenkins Untuk*

Peramalan Data time Series Dalam Kerangka Pemodelan Statistik, 107-125. Laboratorium Ilmu Komputer dan Statistik ITS. F MIPA ITS.

- [2] Bishop, C.M. (1995), *Neural Network For Pattern Recognition*, Oxford University Press.
- [3] Faraway, J and Chatfield. C. (1998), *Time Series Forecasting With Neural Network a Comparative Study Using The Airline Data*, *Applied Statistics*, (47): 231-250.
- [4] Fausett, Laurence V. (1994), *Fundamentals of Neural Networks: Architecture, Algorithms, and Applications*, New Jersey, Prentice Hall.
- [5] Kaashoek, J.F and Van Dijk, H.K. (1991), *Neural Network Analysis Of Varying Trends In Real Exchange Rates*, *Journal Of Applied Econometrics*.
- [6] Kusumadewi, S. (2004), *Membangun Jaringan Saraf Tiruan menggunakan MATLAB & Excel Link*, Yogyakarta, Graha Ilmu.
- [7] Sarle, W.S. April. (1994), *Neural Network and Statistical Models*, *Proceeding of the Nineteenth, Annual SAS Users Group International Conference*, USA.
- [8] Siang, J. J. (2005), *Jaringan Saraf Tiruan & Pemrogramannya Menggunakan Matlab*, Yogyakarta, Andi Offset.