

APLIKASI ALGORITMA BACKPROPAGATION FEEDFORWARD NEURAL NETWORKS UNTUK PREDIKSI ALIRAN SUNGAI (STUDI KASUS DAS GUMBASA, PALU SULTENG)

Rais¹

¹Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Tadulako
email: rais76_untad@yahoo.co.id

Abstrak

Hydrological models are necessary in assessing water resources and valuable tool for water resources management. This paper describes applications of feedforward neural networks (FFNN) for Gumbasa watershed in Palu Sulteng, Indonesia. Back-Propagation was used in the learning rule of FFNN. A series of daily rainfall, evapotranspiration and discharge data for 2 years (2006-2007) from Gumbasa watershed was used. The accuracy is evaluated by statistical performance index, the shape of hydrographs and the flood peaks. The results show that FFNN is successful in predicting watershed discharge in Cidanau watershed. These hydrological models have been developed in form of application program Matlab 7.0.4 and applicable to use in other watershed.

Keywords : Artificial Neural Network, backpropagation, prediction, discharge

I. PENDAHULUAN

Model DAS sangat penting untuk kajian sumber daya air, pembangunan dan manajemen. Sebagai contoh digunakan untuk menganalisa kualitas dan kuantitas aliran air, operasi system reservoir, perlindungan dan pembangunan groundwater dan watersurface, sistem distribusi air, penggunaan air dan aktivitas manajemen sumberdaya air. Alokasi air juga membutuhkan integrasi model DAS dengan model lain seperti lingkungan fisik, populasi biologi dan kegiatan ekonomi. Pada masa datang model akan memerankan peranan penting dalam kehidupan.

Berbagai macam model DAS telah dikembangkan seiring dengan berkembangnya dunia digital (komputer). Baik itu model empiris (black box model), model konseptual (physical process based), lumped model, model distribusi, model single events dan model kontinyu (continoust events). Dan semuanya dibangun dengan persamaan matematika.

Pada tahun 1990an mulailah digunakannya model algoritma backpropagation feedforward neural networks (FFNN) untuk model hidrologi. Hal ini dikarenakan FFNN mempunyai kemampuan untuk belajar dari data dan tidak membutuhkan waktu yang lama dalam pembuatan model. Selain itu FFNN juga mempunyai sifat nonlinier. FFNN mampu mengidentifikasi struktur model dan FFNN juga efektif dan mampu menghubungkan input output simulasi dan model peramalan tanpa membutuhkan struktur internal DAS (1).

Pada tulisan ini akan dibahas aplikasi model algoritma backpropagation feedforward neural networks (FFNN) untuk mengevaluasi data curah hujan-data debit air sungai dan memprediksi aliran sungai pada daerah aliran sungai (DAS) palu Sulawesi Tengah, Indonesia.

II. TEORI FUNDAMENTAL

II.1. Backpropagation Neural Networks

Model pembelajaran FFNN yang digunakan adalah backpropagation dengan struktur multilayer. Terdiri 3 layer yaitu input layer, hidden layer dan output layer. Input layer mempunyai n node. Hidden layer mempunyai h node dan output layer mempunyai m node. Algoritma pembelajaran FFNN Backpropagation dijelaskan sebagai berikut:

II.1.1. Inisialisasi Pembobot

Pembobot awal pada FFNN diberi nilai secara acak. Nilai acak ini biasanya berkisar -1 sampai 1 atau 0 sampai 1.

II.1.2. Perhitungan Nilai Aktifasi

Perhitungan feedforward dimulai dengan menjumlahkan hasil perkalian input x_i dengan bobot-bobot pada layer tersembunyi di neuron ke-j v_{ij} . Dan menghasilkan N_j yang merupakan nilai input ke fungsi aktivasi pada layer tersembunyi, kemudian output A_j layer tersembunyi yang terproses di neural ke-j merupakan hasil fungsi aktivasi f dengan masukan N_j . hal ini dapat diformulasikan dalam persamaan

$$N_j = \sum_{i=1}^p v_{ji}x_i \quad , \quad \text{atau} \quad N_{j(k)} = \sum_{i=1}^p v_{ji}x_{i(k)} \quad \dots\dots\dots(1)$$

$$A_j = f_j(N_j), \quad \text{atau} \quad A_{j(k)} = f_j^h(N_{j(k)}) = f_j \left(\sum_{i=1}^p v_{ji}x_{i(k)} \right) \quad \dots\dots\dots(2)$$

Nilai output pada layer tersembunyi kemudian dikalikan dengan nilai bobot-bobot pada w_{ij} yang menghasilkan nilai I_k yang merupakan nilai input fungsi aktivasi pada layer output.

Nilai output Z_k pada output layer dihitung dengan menggunakan fungsi aktivasi f dengan masukan I_k . Hal ini diformulasikan dalam persamaan 3 dan 4 berikut:

$$I_k = \sum_{j=1}^q w_{kj}A_j \quad \dots\dots\dots(3)$$

$$z_k = f(I_k) \quad \dots\dots\dots(4)$$

Dengan fungsi aktivasi yang digunakan untuk layer tersembunyi dan layer output adalah fungsi yang sama yaitu logistik sigmoid, yaitu

$$f(n) = (1 + e^{-\alpha n})^{-1}. \quad \dots\dots\dots(5)$$

Dimana α adalah gain atau slope fungsi *logistic sigmoid* (konstanta).

II.1.3. Pelatihan (pengoreksian) Nilai Pembobot

Pelatihan nilai pembobot pada FFNN ini dilakukan dengan mengurangi atau menurunkan total error sistem untuk semua data melalui koreksi (ajusment) pembobotan dengan algoritma metode Gardien Descent yaitu

$$\Delta w = -\eta \frac{\partial E(w)}{\partial w} \quad \text{atau} \quad w^{(m+1)} = w^m - \eta \frac{\partial E(w)^{(m)}}{\partial w} \quad \dots\dots\dots (6)$$

Dimana η adalah laju pembelajaran (konstanta yang nilainya $0 < \eta < 1$).

Secara ringkas pengkoreksi pembobot antara output layer dan layer tersembunyi adalah sebagai berikut:

$$-\eta \frac{\partial E}{\partial w_{kj}} = \Delta w_{kj} = \eta (t_k - t_z) f'(I_k) y_j \dots\dots\dots(7)$$

Dan pengkoreksi pembobotan antara layer tersembunyi dan input layer adalah sebagai berikut:

$$-\eta \frac{\partial E}{\partial v_{ij}} = \Delta v_{kj} = \eta x_i f'(N_j) \sum_k \delta_k w_{kj} \dots\dots\dots(8)$$

Dimana $\delta = (t_k - t_z) f'(I_k)$

Untuk mempercepat konvergen, ditambahkan inersia atau momentum, yaitu dengan menambahkan pengkoreksi pembobot sebelumnya ke pengkoreksi pembobot sekarang. Sehingga pengkoreksi pembobot antara output layer dan hidden layer dan antara hidden layer dan input layer berturut-turut ditulis seperti berikut:

$$\Delta w_{kj}(t + 1) = -\eta \frac{\partial E}{\partial w_{kj}} + \alpha \Delta w_{kj}(t) \dots\dots\dots(9)$$

$$\Delta v_{kj}(t + 1) = -\eta \frac{\partial E}{\partial v_{kj}} + \alpha \Delta v_{kj}(t) \dots\dots\dots(10)$$

dimana α adalah momentum (konstanta $0 < \alpha < 1$)

Proses perhitungan pembobot atau updating bobot-bobot antara output layer dan layer tersembunyi dilakukan dengan persamaan berikut:

$$w_{kj}^{(m+1)} = w_{kj}^{(m)} + \Delta w_{kj}(t + 1) \dots\dots\dots(11)$$

dan pembobot atau updating bobot-bobot antara layer tersembunyi dan input layer dilakukan dengan persamaan berikut:

$$v_{kj}^{(m+1)} = v_{kj}^{(m)} + \Delta v_{kj}(t + 1) \dots\dots\dots(12)$$

Ilustrasi jalanya persamaan update untuk bobot-bobot di layer output $w_{kj}^{(m+1)}$ dapat dilihat pada gambar 1, sedangkan untuk di layer tersembunyi $v_{kj}^{(m+1)}$ dapat di lihat pada gambar 2.

II.1.4. Bias pada FFNN

Selain mengandung bobot-bobot network dalam layer tersembunyi atau pada layer output, sebagian besar FFNN juga mengandung bias atau konstanta pada masing-masing layer tersebut. Bias biasanya disimpulkan dengan b, dalam penelitian ini dinotasikan dengan b_j^h untuk bias di layer tersembunyi dan b_o^0 yang ada di layer output. Sehingga vektor di persamaan (1) dapat ditulis

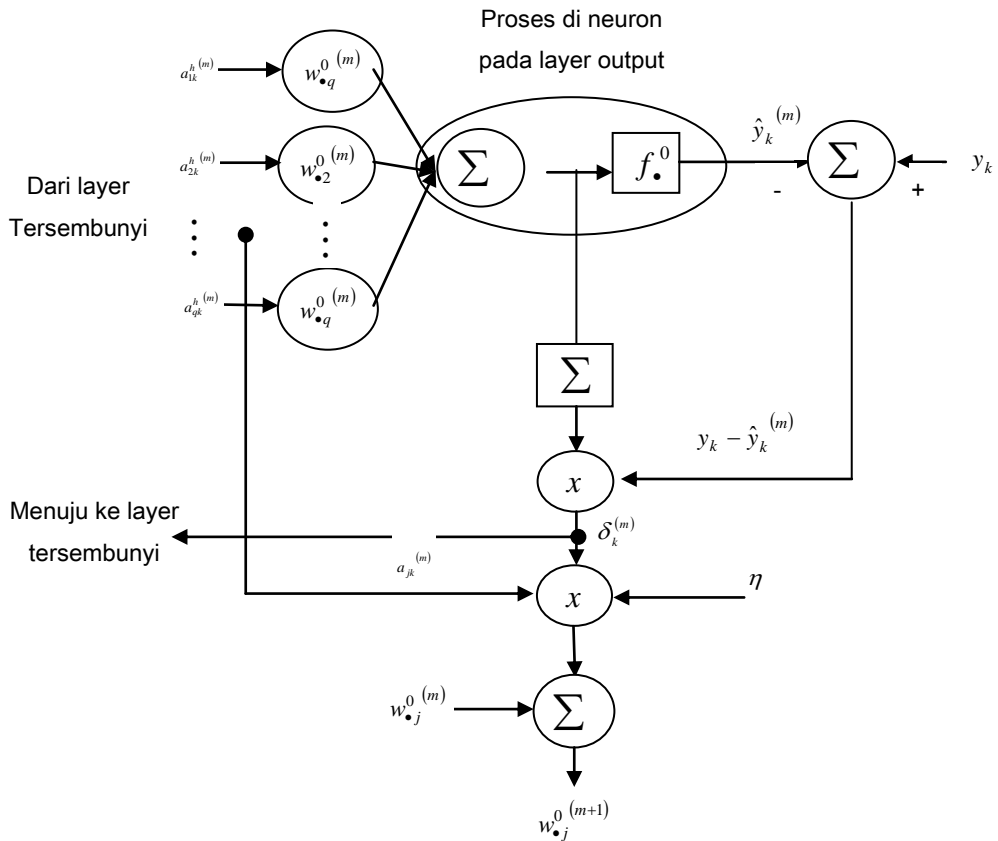
$$kembali menjadi $N_j = \sum_{i=1}^p v_{ji} x_i + b_j^h$, atau $N_{j(k)}^h = \sum_{i=1}^p v_{ji}^h x_{i(k)} + b_j^h$. \dots\dots\dots(13)$$

Pada layer output yang menyatakan penjumlahan input dan bobot-bobot pada neuron di layer ini adalah $n_{o(k)}^0 = \sum_{j=1}^q w_{oj}^0 a_{j(k)}^h + b_o^0$. \dots\dots\dots (14)

Pada tahap ini, formula dari algoritma gradien untuk mengupdate bias dari FFNN telah dapat diturunkan. Dua persamaan update untuk bias b_j^h dan b_o^0 akan diberikan secara terpisah, yaitu

Untuk updating bias pada layer output $b_{\bullet}^{0(m+1)} = b_{\bullet}^{0(m)} + \eta \sum_{k=1}^n \delta_{(k)}^{0(m)}$(15)

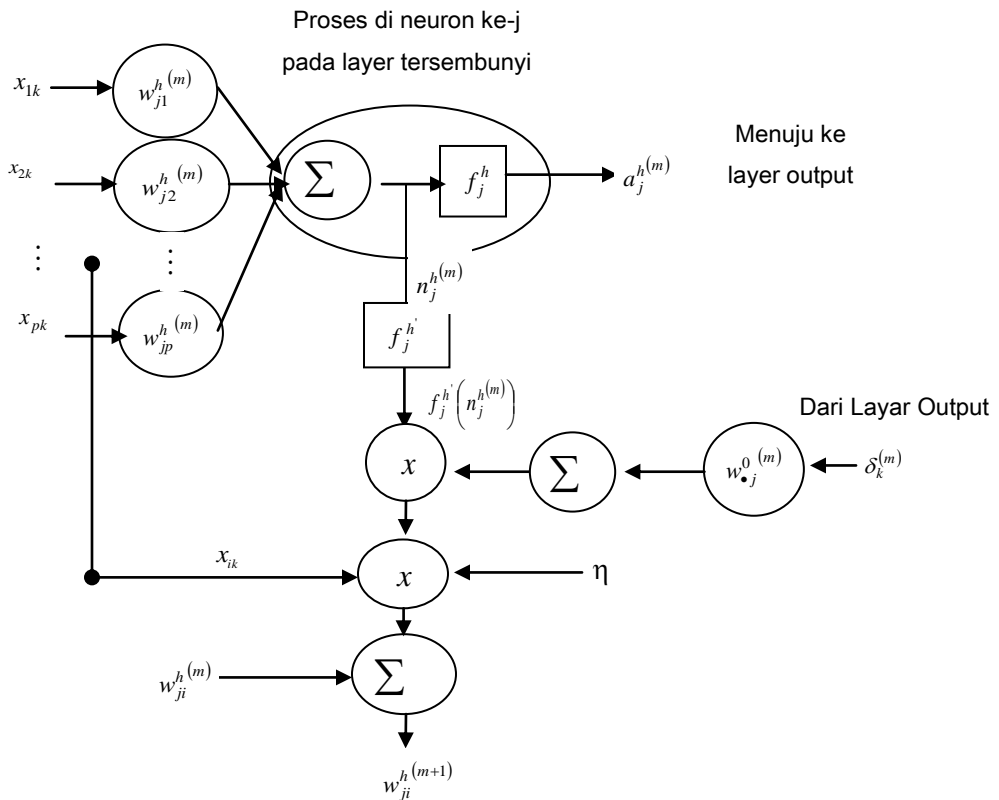
Untuk updating bias pada hidden layer $b_j^{h(m+1)} = b_j^{h(m)} + \eta \sum_{k=1}^n \left(\delta_{j(k)}^{h(m)} \right)$ (16)



Gambar: 1. Ilustrasi Persamaan update bobot pada output layer

II.1.5. Pengulangan

Keseluruhan proses ini dilakukan pada setiap contoh dan setiap iterasi. Proses pemberian contoh atau pasangan input-output, perhitungan nilai aktivasi dan pembelajaran dengan mengkoreksi pembobot dilakukan terus menerus sampai didapatkan nilai pembobot dengan nilai total error system mencapai minimum global.



Gambar. 2. Ilustrasi Persamaan update bobot pada hidden layer

III. METODE DAN BAHAN

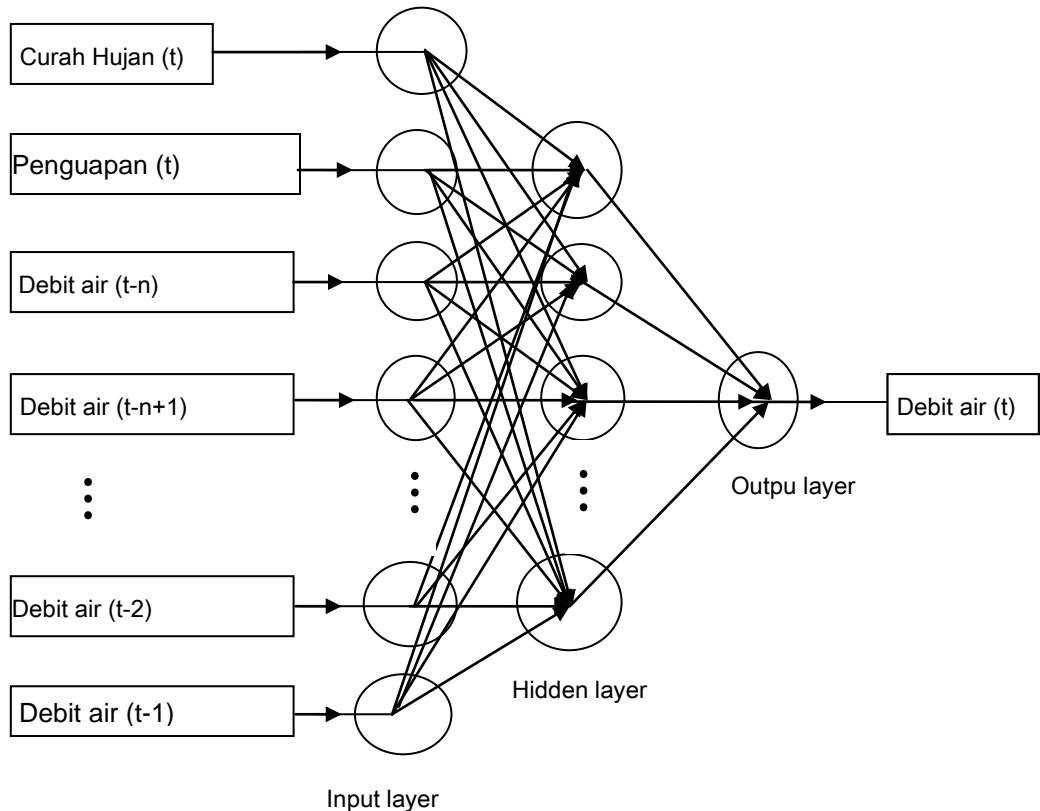
Model FFNN digunakan untuk menganalisis hubungan curah hujan dan debit air pada sungai palu. Data harian curah hujan, penguapan dan debit aliran sungai tercatat dengan baik selama tiga tahun yaitu 2006 – 2008, data tahun 2006-2007 digunakan sebagai pembelajaran dan data tahun 2008 digunakan sebagai verifikasi.

Struktur FFNN yang digunakan adalah multi-layer perceptrons. Tiga struktur FFNN multilayer digunakan untuk ujicoba (jumlah node mengindikasikan untuk masing-masing model dalam masing-masing layer). Berikut adalah ketiga model tersebut:

Model 1 (3 – 10 – 5 – 1): 3 node input yaitu data curah hujan, penguapan, data debit aliran sungai t-1; 10 node hidden layer pertama, 5 node hidden layer kedua dan 1 node output layer yaitu data debit aliran sungai t.

Model 2 (4 – 10 – 5 – 1): 4 node input yaitu data curah hujan, penguapan, data debit aliran sungai t-1, data debit aliran sungai t-2; 10 node hidden layer pertama, 5 node hidden layer dan 1 node output layer yaitu data debit aliran sungai t.

Model 3 (5 – 10 – 5 – 1): 5 node input yaitu data curah hujan, penguapan, data debit aliran sungai t-1, data debit aliran sungai t-2, data debit aliran sungai t-3 ; 10 node hidden layer pertama, 5 node hidden layer dan 1 node output layer yaitu data debit aliran sungai t. arsitektur jaringan model FFNN dapat dilihat pada gambar 3.



Gambar 3. Arsitektur Model FFNN untuk prediksi debit air

Laju pembelajaran, konstanta momentum dan gain fungsi sigmoid diberi nilai sama yaitu 0.9. Pemberhentian proses pembelajaran didasarkan pada jumlah iterasi. Pada penelitian ini pembelajaran berhenti jika iterasi mencapai 10.000 dan ditampilkan setiap iterasi ke-50. Software Neural Network (NN) yang digunakan adalah Matlab 7.0.4 yang ditulis dalam bahasa pemrograman komputer .

Kinerja model FFNN dilihat dari indikator keeratan linier debit aliran sungai observasi dan perhitungan yang berupa R (Coefficient of Correlation) dan indikator error yang berupa RMSE (Root Square Mean Error). Evaluasi model juga dapat dilihat secara kualitatif deviasi time of peak dan perbandingan kurva yang memberikan keseluruhan gambaran keandalan model.

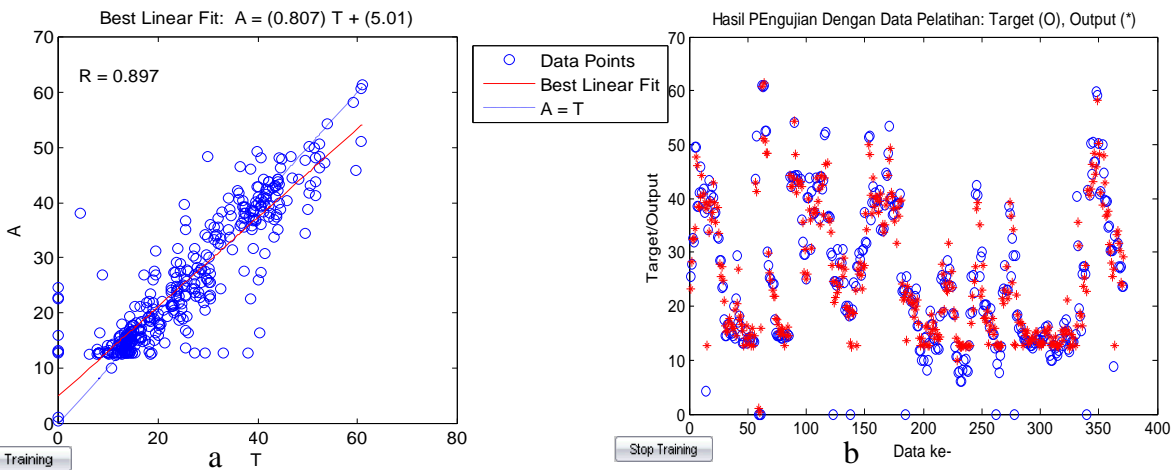
IV. HASIL ANALISIS DAN PEMBAHASAN

Hasil kinerja masing-masing model yang berupa korelasi (R) dan Root Means Square Error (RMSE) untuk prediksi Daerah Alira Sungai Palu Sulawesi tengah ditunjukkan Pada tabel 1, dimana

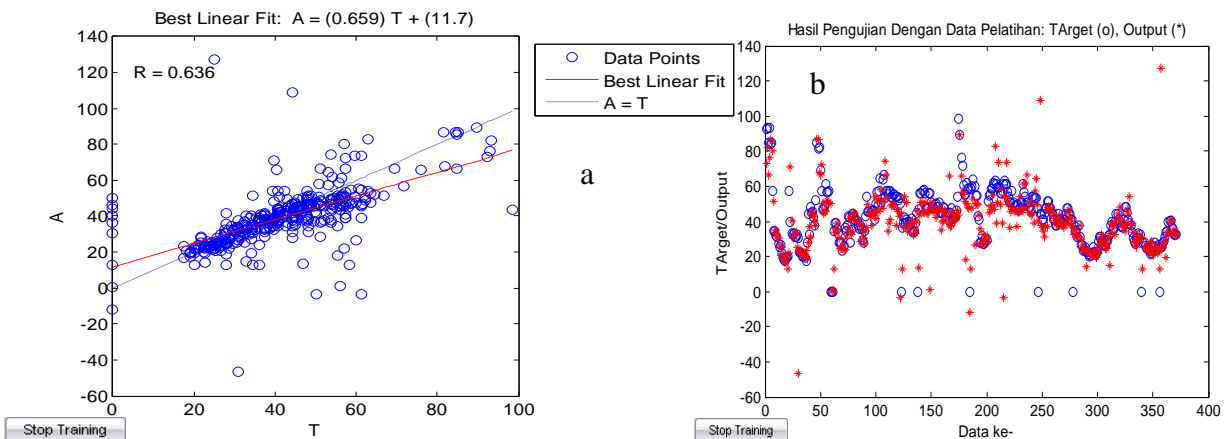
Korelasi dan RMSE hubungan runoff (discharge) hitung dengan observasi hasil training menunjukkan bahwa model 3 memberikan hasil terbaik dengan nilai R (0.935) mendekati satu seperti pada gambar 8, sedangkan untuk hasil verifikasi menunjukkan bahwa model 1 memberikan hasil yang terbaik dengan nilai R adalah 0.636. hubungan dan perbandingan antara target dengan output jaringan untuk data training dan data verifikasi dapat dilihat pada gambar 4-9. Hasil terbaik terjadi apabila posisi (O) dan (*) betul-betul berada pada posisi yang sama.

Tabel 1. Kinerja R dan RMSE untuk masing-masing model

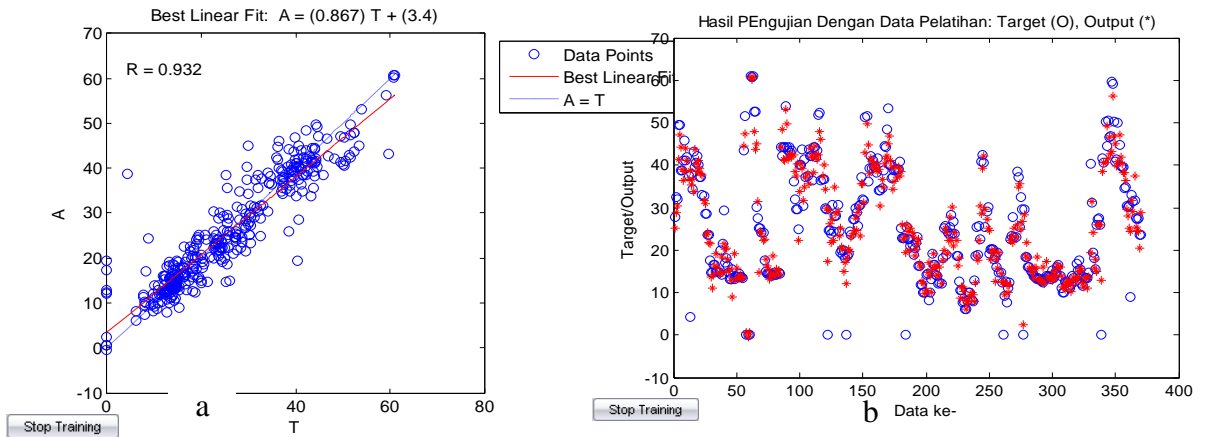
Model	RMSE	Data Training	Data Verifikasi
		R	R
Model 1	0.195147	0.897	0.636
Model 2	0.131439	0.932	0.622
Model 3	0.125044	0.935	0.461



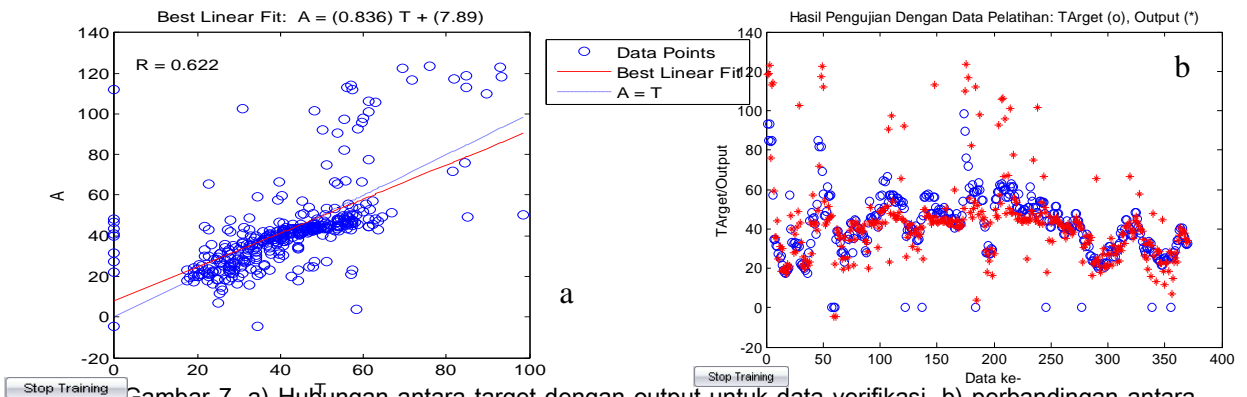
Gambar 4. a) Hubungan antara target dengan output untuk data training, b) perbandingan antara target dengan output untuk data training untuk model 1



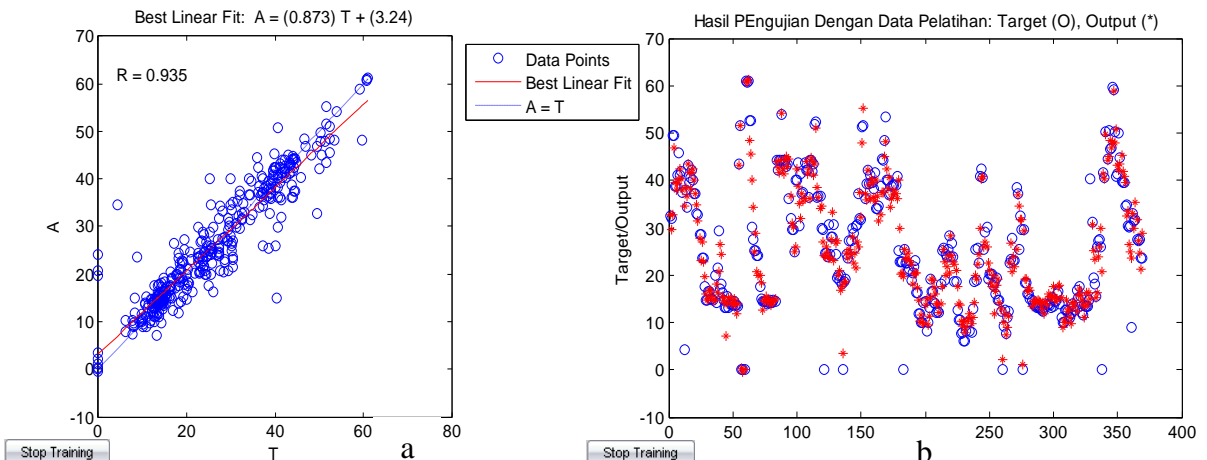
Gambar 5. a) Hubungan antara target dengan output untuk data verifikasi, b) perbandingan antara target dengan output untuk data verifikasi, untuk model 1



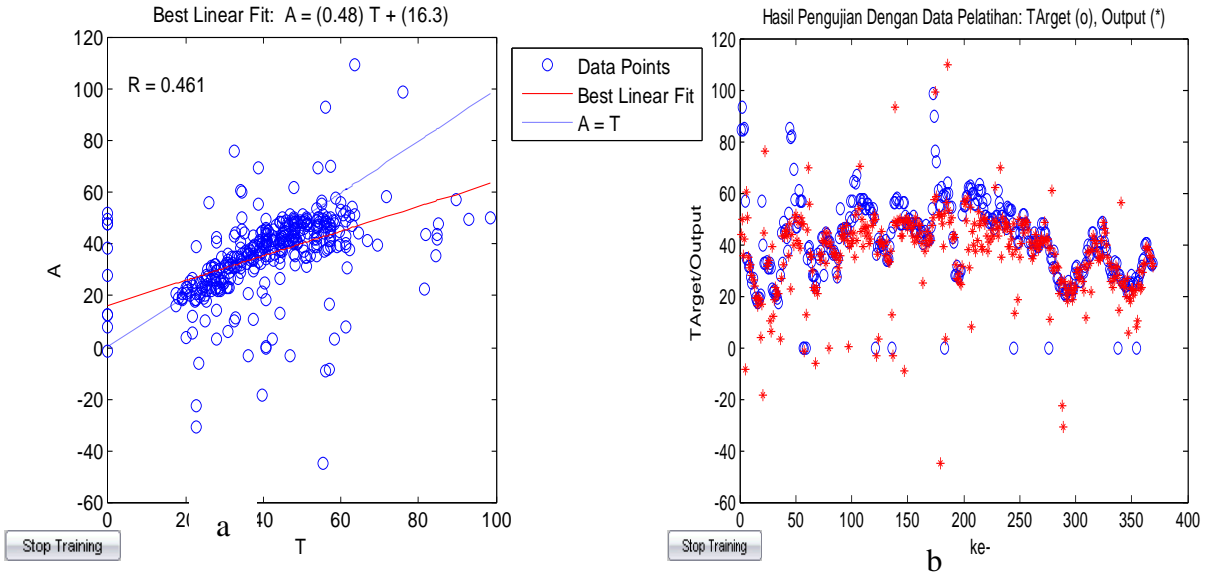
Gambar 6 a) Hubungan antara target dengan output untuk data training, b) perbandingan antara target dengan output untuk data training untuk model 2



Gambar 7. a) Hubungan antara target dengan output untuk data verifikasi, b) perbandingan antara target dengan output untuk data verifikasi untuk model 2



Gambar 8. a) Hubungan antara target dengan output untuk data training, b) perbandingan antara target dengan output untuk data training untuk model 3



Gambar 9. a) Hubungan antara target dengan output untuk data verifikasi, b) perbandingan antara target dengan output untuk data verifikasi untuk model 3

IV. KESIMPULAN

Ketiga model ANN yang dibuat telah mampu dengan baik menduga *runoff* DAS Palu, dari ketiga model yang dikembangkan model 3 (5 – 10 – 5 – 1) memberikan hasil yang terbaik dengan R adalah 0.935.

V. DAFTAR PUSTAKA

1. Budi, I.S. and Rudiyanto, 2004, *Aplikasi Neural Network Untuk Prediksi Aliran Sungai (Studi Kasus DAS Cidanau, Indonesia dan DAS Terauchi, Jepang*, Prosiding Semiloka Teknologi Simulasi dan Komputasi serta aplikasi 2004, 61-66
2. Fausett, Laurene, 1994, *Fundamentals of Neural Networks*, Prentice Hall Inc, New Jersey.
3. McCulloch, W.S. and Pitts, W., 1943, *A Logical Calculus of ideas immanent in nervous activity*, *Bulletin of Msthematical Biophysicss*, 15, 115-133.
4. Rais, 2005, *Kriteria Kesesuaian Model Untuk Penentuan Arsitektur Optimal Pada Neural Network Untuk Pemodelan Time Series*, Tesis Yogyakarta.
5. 2008, *Kriteria Kesesuaian Model Untuk Penentuan Arsitektur Optimal Model Neural Network Backpropagation Untuk Indeks Harga Saham Gabungan*, Jurnal Ilmiah Matematika dan Terapan Vol. 5 No. 1 hal 1 – 11.
6. Ripley, B.D., 1996, *Pattern recognition and Neural Network*, Cambridge University Press, Australia.