

PENERAPAN JARINGAN SARAF TIRUAN UNTUK PERAMALAN

Siana Halim

Dosen Fakultas Teknologi Industri, Jurusan Teknik Industri – Universitas Kristen Petra

Adrian Michael Wibisono

Alumnus Fakultas Teknologi Industri, Jurusan Teknik Industri – Universitas Kristen Petra

ABSTRAK

Ada banyak metode yang telah dikembangkan untuk mencapai hasil yang optimal dari suatu peramalan. Salah satu yang akan diulas pada makalah ini adalah penggunaan *Neural Network* atau jaringan saraf untuk mendapatkan hasil peramalan yang diharapkan dapat meningkatkan optimasi dan akurasi. Hasil dari metode ini akan dibandingkan dengan metode *GARCH(1,1)* dalam bentuk *Means Absolute Deviation (MAD)* dan *Means Square Error (MSE)*. Selain itu dilakukan pula pengamatan terhadap peredaman *jump* (perubahan mendadak). Data yang digunakan adalah nilai tukar mata uang dari empat negara di Asia yang diambil selama krisis moneter di Asia.

Kata Kunci: *Backpropagation, MAD, MSE, GARCH(1,1), jump.*

ABSTRACT

Many methods have been developed to get the optimal result in forecasting. One of them that will be used in this paper is using Neural Network for forecasting. The result will be compared with GARCH(1,1) in the terms of Means Absolute Deviation (MAD) and Means Square Error (MSE). Besides that the accuracy and the power to damp the jump will be observed. The data is currency rate from 4 countries in Asia taken during the Asian Monetary Crisis from 1997 up to 1999 since the jump was happened in that series.

Keywords: *Backpropagation, MAD, MSE, GARCH(1,1), jump.*

1. PENDAHULUAN

Beberapa studi tentang penerapan jaringan syaraf tiruan untuk peramalan telah dilakukan (Bambang, et. al., 1999), (Pratama, 1999) dan (Resmana dan Dwi Wiyanto, 1997). Namun demikian pada penelitian ini ingin diketahui kelebihan dan kekurangan dari metode ini bila dibandingkan dengan penelitian yang pernah dilakukan dengan Metode *Generalized Auto Regressive Contional Heteroscedastics (GARCH(1,1))*, (Siana Halim, et. al., 1999), terutama dalam permasalahan peredaman *jump* (perubahan yang terjadi secara mendadak pada fluktuasi mata uang).

Ide dasar untuk menggunakan jaringan saraf sebagai metode peramalan didasari pada adanya kesamaan yang ditemukan antara struktur jaringan saraf dengan pendekatan umum metode peramalan, khususnya antara *Adaptive Linear Filter* dengan proses Autoregresi, yaitu:

$$\tilde{z}_t = \mathbf{f}_1 \tilde{z}_{t-1} + \mathbf{f}_2 \tilde{z}_{t-2} + \dots + \mathbf{f}_p \tilde{z}_{t-p} + a_t \quad (1)$$

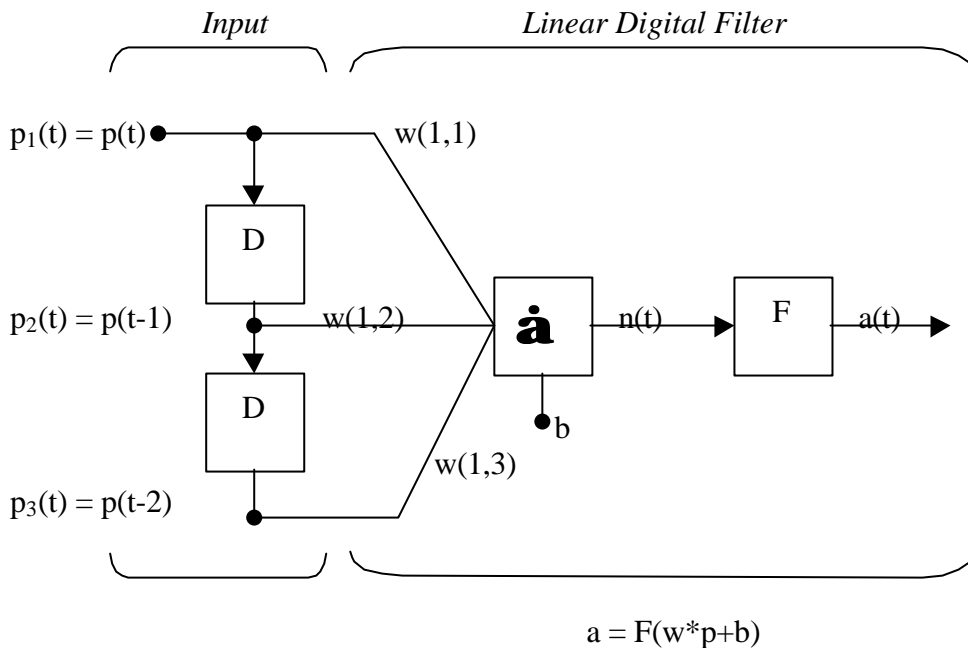
Dengan notasi sebagai berikut :

Z_{t-p} adalah data pada periode ke- $t-p$

\mathbf{f}_p adalah parameter yang merupakan pembobotan dari autoregresi

a_t , adalah galat.

Adaptive Filter (Contoh)



Gambar 1. Contoh Adaptive Filter

Dengan notasi sebagai berikut :

- $p_1(t)$ adalah *input* 1 pada periode t
- $w(1,1)$ adalah bobot untuk *input* 1 pada *neuron* 1
- b adalah bias / galat

Dapat diperhatikan bahwa dengan adanya *delay D*, menjadikan data pada periode sebelumnya juga masuk sebagai *input* (seperti pada proses autoregresi dengan parameter Z_{t-p}), dan juga adanya faktor bobot w (dianalogikan dengan parameter f_p) juga adanya faktor bias (b) – analogi dengan faktor a_i (galat)

Perkembangan jaringan saraf selanjutnya memberikan algoritma *Backpropagation* yang diperkenalkan oleh David Rumelhart dan James Mc Clelland (1986). Saat ini *Backpropagation* merupakan struktur jaringan saraf yang lebih umum digunakan dan juga akan digunakan pada makalah ini.

2. JARINGAN BACKPROPAGATION

Algoritma *backpropagation (BP)* merupakan pengembangan dari algoritma *least mean square* yang dapat digunakan untuk melatih jaringan dengan beberapa layer. *BP* menggunakan pendekatan algoritma *steepest descent*. Algoritma ini menggunakan *performance index*-nya adalah *mean square error*¹

¹ Hagan, Martin T., et al. *Neural Network Design*. Boston: PWS Publishing Company; 1996

Untuk melatih jaringan diperlukan seperangkat pasangan data seperti berikut :

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_n, t_n\} \quad (2)$$

dimana p_n adalah nilai *input* ke- n jaringan dan t_n adalah target, yaitu nilai *output* yang seharusnya dihasilkan. Untuk setiap *input* yang masuk dalam jaringan, *output* yang dihasilkan oleh jaringan akan dibandingkan dengan target. Algoritma ini akan mengatur atau menyesuaikan parameter-parameter jaringan untuk meminimalkan *mean square error*, yaitu :

$$F(x) = E(e^2) = E[(t-a)^2] \quad (3)$$

dimana x , e , t dan a merupakan vektor bobot dan bias, vektor *error*, vektor target dan vektor *output*. Jika jaringan mempunyai beberapa *output* maka persamaan di atas dapat dikembangkan menjadi:

$$F(x) = E[e^T e] = E[(t-a)^T (t-a)] \quad (4)$$

Mean square error didekati dengan

$$\hat{F}(x) = e^T(k) e(k) \quad (5)$$

Langkah-langkah dalam algoritma *BP* adalah sebagai berikut :

- a. *Forward propagation*
Menyalurkan *input* ke dalam jaringan dan tiap *layer* akan mengeluarkan *output*. *Output* dari satu *layer* akan menjadi *input* untuk *layer* berikutnya.
- b. *Back propagation*
Menghitung nilai sensitivitas untuk tiap *layer*. Dimana sensitivitas untuk *layer* m dihitung dari sensitivitas pada *layer* $m+1$ sehingga penghitungan sensitivitas ini berjalan mundur.
- c. *Weight Update*
Menyesuaikan nilai parameter bobot (W) dan bias (b) dengan menggunakan pendekatan *steepest descent*.

Backpropagation dengan *least mean square* seperti di atas memang menjamin penyelesaian dengan minimum *mean square error* selama *learning rate*-nya tidak terlalu besar. Kekurangannya adalah bila *learning rate*-nya kecil, maka pencapaian nilai konvergennya lambat, sedangkan bila *learning rate*-nya besar, pencapaian nilai konvergennya cepat namun ada bahaya osilasi yang dapat mengakibatkan nilai minimum global tidak tercapai. Untuk mengatasi hal ini maka digunakanlah variasi *backpropagation* sebagai berikut :

- a. *Momentum*
Metode ini bekerja dengan tujuan untuk menghaluskan osilasi yang terjadi. Filter momentum ini akan ditambahkan pada persamaan *weight matrix* dan bias.
- b. *Variable Learning Rate*
Metode ini bekerja dengan berusaha menaikkan *learning rate* bila menjumpai permukaan yang datar dan kemudian menurunkan *learning rate* bila terjadi peningkatan *slope*.

3. PENGOLAHAN DATA

Perencanaan sistem *forecasting* valuta asing menggunakan aplikasi jaringan saraf buatan, dengan metode *backpropagation* terdiri dari langkah-langkah berikut :

1. Menetapkan tujuan sistem yaitu mampu mengidentifikasi dan mempelajari pola dari *financial time series* dan selanjutnya menggunakannya untuk *forecasting*.
2. Menentukan jangka waktu prediksi melalui pengujian jaringan saraf yang sudah terlebih dahulu menjadi proses pelatihan sehingga dapat diketahui seberapa kemampuan jaringan tersebut dalam meramalkan nilai di masa mendatang dengan *error* minimum. Pada makalah ini jangka waktu prediksi akan dibatasi antara 4 sampai 8 periode ke depan.
3. Memperoleh data, yaitu dari internet [<http://www.xe.net/ict/>], dari nilai tukar Rupiah Indonesia, Yen Jepang, Dolar Singapura dan Baht Thailand terhadap Dolar Amerika. *Financial time series* dibentuk dari Januari 1997 sampai dengan Oktober 1999. Penentuan *input-output* pengolahan data adalah :
 - a. *X*, variabel waktu
 - b. *Moving Average* dan Momentum yang dikalkulasi secara keseluruhan sesuai dengan yang diinginkan, misalnya: bila *input* = 5, maka data akan diolah untuk menghitung *MA* 2 sampai *MA* 5 dan momentum 2 sampai momentum 5. Penentuan *input* £ 5 dengan dasar bahwa data finansial memiliki interval 5 hari kerja.
 - c. *Output* sistem data terdiri dari 4 macam valas yaitu: *Indonesian Rupiah (IDR)*, *Japanese Yen (JPY)*, *Singapore Dollar (SGD)*, dan *Thailand Baht (THB)*, masing-masing terhadap 1 US\$.Dengan demikian, sistem data akan memiliki 33 *input* dan 4 target *output*.
4. Menentukan *transfer function*. Dalam kasus ini bentuk khusus dari suatu fungsi transfer tidak akan memiliki pengaruh yang signifikan terhadap *performance* suatu jaringan². Namun dengan dasar karakteristik data yang berfluktuasi kuat dan mengingat tujuan dari sistem jaringan yaitu untuk mampu mengatasi perubahan mendadak, maka fungsi transfer hendaknya yang bersifat *smooth* dan terdiferensial. Atas dasar hal-hal di atas, maka dipilih fungsi transfer *log sigmoid*.
5. Normalisasi data *input* bertujuan untuk menyesuaikan nilai *range* data dengan *log-sigmoid threshold function* dalam sistem *backpropagation*. Ini berarti nilai kudrat *input* harus berada pada range 0 sampai 1. Sehingga *range input* yang memenuhi syarat adalah nilai data *input* dari 0 sampai 1 atau dari -1 sampai 1. Oleh karena itu *output* yang dihasilkan pun akan berada pada range 0 sampai 1, untuk mendapatkan nilai sebenarnya dari *output* perlu dilakukan proses denormalisasi.
6. Merancang struktur jaringan bertujuan untuk mendapatkan komposisi jumlah *neuron*, *hidden layer*, elemen *input* dan nilai parameter *training* yang optimal. *Input* untuk sistem jaringan ini terdiri atas 33 elemen yaitu satu variabel waktu, 16 elemen *MA* dan 16 elemen momentum, dari masing-masing valuta asing. *Output* yang dihasilkan sejumlah 4 buah yaitu prediksi nilai tukar *IDR*, *JPY*, *SGD* dan *THB*, masing-masing terhadap 1 US\$. Langkah-langkah membangun model jaringan data ini adalah :
 - a. Membetuk data *training* dan *testing*. Persentase data *training* adalah 90% dari data (sejumlah 638) dan 10 % sisanya (76 data) untuk *testing*.
 - b. Fungsi tranfer yang digunakan *logsig*. Penentuan jumlah *hidden layer* dan jumlah *neuron* ditentukan secara *trial and error* sampai dicapai *error* minimumnya
 - c. Penempatan perintah pada program pelatihan untuk menyimpan *rmse (root mean square) training*, *rmse testing* dan *epoch* apabila tercapai *rmse testing*

² Franke, Juergen. Kaiserslautern : Dept. of Mathematics, University of Kaiserslautern Germany, 1999.

yang lebih minimum dari sebelumnya. Modifikasi ini diharapkan untuk membantu mempercepat tercapainya minimum global dari *epoch* maksimum yang dijalankan. Metode training yang diterapkan pada jaringan saraf *backpropagation* antara lain, *trainbp*, *trainbpm*, *trainbpx* dan *trainnlm*. Masing-masing diterapkan pada sistem jaringan sehingga tujuan dapat tercapai.

4. HASIL PENGOLAHAN DAN ANALISA DATA

Topologi jaringan saraf dari data lanjutan yang berhasil mencapai *error* paling optimal adalah satu *hidden layer* dengan 80 *neuron* (33-80-4). Hasil tabulasi *rmse* dapat dilihat pada tabel berikut. Pada Tabel 1 di bawah terlihat adanya pola menurunnya *error training* seiring dengan bertambahnya jumlah *neuron*. Sebaliknya, *error testing* cenderung berbanding lurus terhadap pertambahan jumlah *neuron*. Oleh karena itu dipilihlah jumlah *neuron* 80 sebagai titik optimal.

Kolom 2 pada Tabel 1 memuat *epoch* terjadinya pasangan *error* (*training* dan *testing*). Hal ini ditambahkan pada program sedemikian karena *Neural Network Toolbox* MATLAB tidak menyimpan *error* terbaik, melainkan *error* terakhir. Dengan adanya tambahan tersebut, *error* terbaiklah yang didokumentasikan dari 10.000 *epoch*. Pola dokumentasi *epoch* bersifat acak, serupa dengan sistem pemberian bobot dan bias pada masing-masing *neuron* oleh struktur jaringan yang telah dibentuk. Maka dapat ditarik kesimpulan bahwa semakin banyak jumlah *epoch* yang dilatihkan pada jaringan akan memperbesar kemungkinan tercapainya *rmse* terbaik.

Tabel 1. RMSE Terbaik

<i>Neuron</i>	<i>Epoch</i>	<i>RMSE Training</i>	<i>RMSE Testing</i>	<i>Metode</i>	<i>Learning Rate</i>
75	1664	0.0312	0.0261	TrainBP	0.01
80	8446	0.0293	0.0272	Trainbp	0.01
81	6811	0.0264	0.0284	TrainBP	0.01
83	9999	0.0273	0.0291	TrainBP	0.01

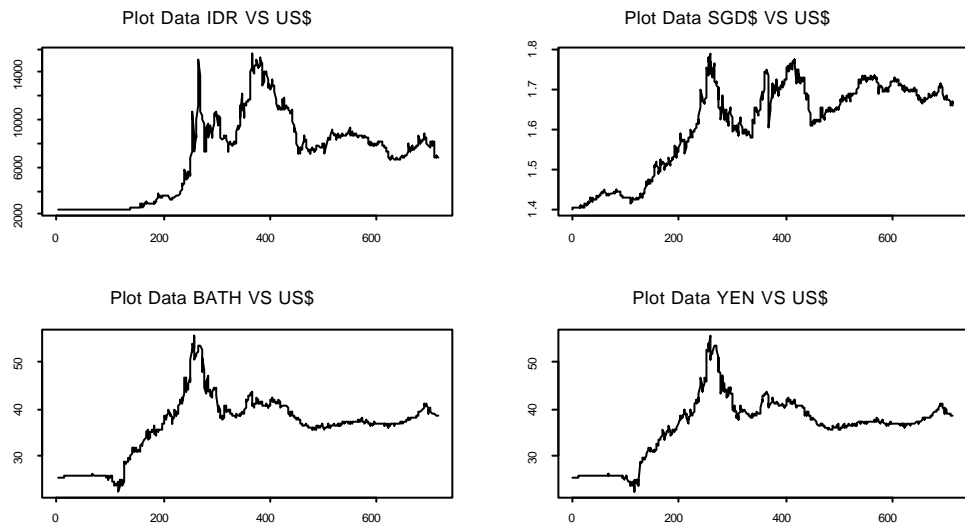
Untuk data ini, *MAD* dan *MSE* yang dibandingkan adalah antara hasil *training* jaringan saraf dan *fitting GARCH(1,1)*, Tabel 2 dan Tabel 3. *MAD* maupun *MSE* jaringan syaraf adakalanya tidak lebih baik dari *MAD* dan *MSE* dari *GARCH (1,1)*. Namun kemampuannya dalam meredam *jump* masih lebih baik. Hasil ini dapt dilihat lewat perbandingan *plot residual Vs time order*.

Tabel 2. Model GARCH(1,1)

Time series	Model	MAD	MSE
Indonesian Rupiah	$X_t^2 = 418858.3 + 1.04672 X_{t-1}^2 - 0.05368796 W_{t-1}^2 + W_t$	183.1265	122045.2
Japanese Yen	$X_t^2 = 0.01583169 + 0.9208159 X_{t-1}^2 + 0.07313421 W_{t-1}^2 + W_t$	0.8232848	1.341864
Singapore Dollar	$X_t^2 = 1.583154 + 0.9208116 X_{t-1}^2 + 0.07313853 W_{t-1}^2 + W_t$	0.9432898	1.624721
Thailand Baht	$X_t^2 = 8.9251 + 1.029563 X_{t-1}^2 - 0.03606654 W_{t-1}^2 + W_t$	0.5262436	0.544534

Tabel 3. Tabulasi MAD dan MSE Struktur Jaringan (33-80-4)

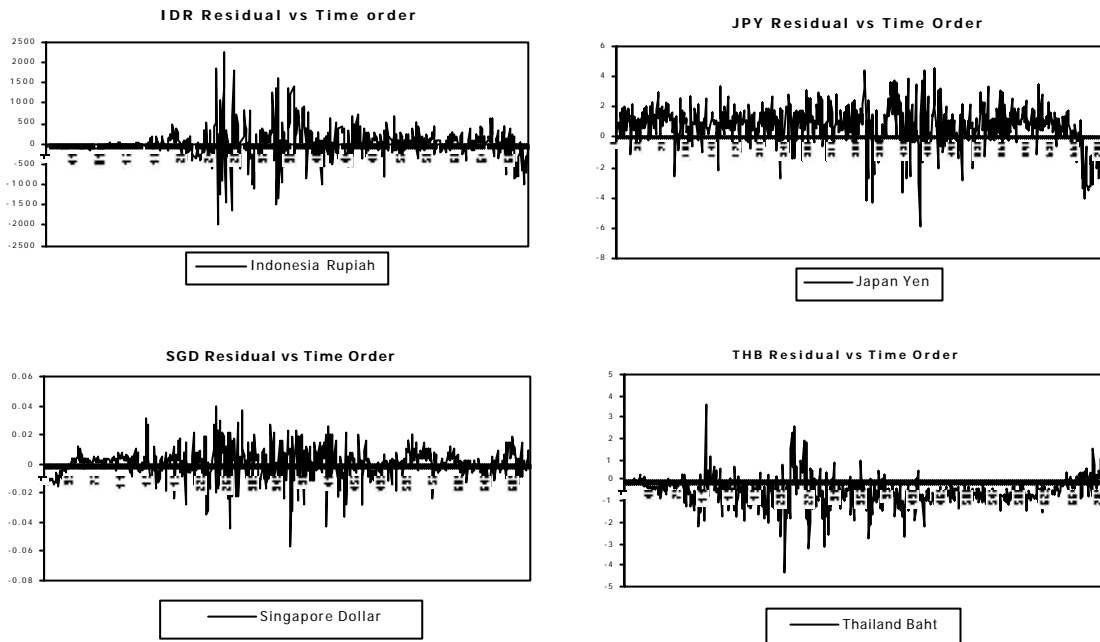
	Indonesian Rupiah		Japanese Yen	
	MAD	MSE	MAD	MSE
Training	218.62	132335.10	1.218089	2.230355
Testing	277.91	135365.46	1.36299	2.902746
	Singapore Dollar		Thailand Baht	
	MAD	MSE	MAD	MSE
Training	0.00756	0.0001144	0.737	0.8383
Testing	0.00641	6.3754E-05	0.3912	0.2409



Gambar 2. Plot Data IDR, JPY, SGD dan THB

Dalam menentukan *input* suatu struktur jaringan saraf tiruan, perlu dilakukan pengamatan serta analisa bobot dan bias yang dihasilkan pada tiap *input*. Hal ini akan membantu jaringan karena dapat mengetahui jenis *input* yang signifikan bagi tujuan proses sehingga kalkulasi menjadi lebih sederhana dan mempercepat proses *running*. Analisa bobot dan bias ini dapat dijalankan dengan melakukan pemetaan bobot dan bias dan/atau *arbitraty checking*.

Hasil analisa terhadap *input* variabel waktu menunjukkan hasil yang memperlihatkan pola bobot maupun bias yang *asymptotic* ke titik tertentu. *Abitraty checking* juga menunjukkan hasil yang serupa dengan pemetaan sebelumnya. Dari hasil ini dapat diambil kesimpulan bahwa variabel waktu harus dipertimbangkan kembali sebelum ditetapkan sebagai *input*, atau diolah/dimodifikasi sedemikian hingga menjadi *input* yang signifikan bagi proses dari jaringan yang dibangun.



Gambar 3. Plot Residual Vs Time Order dari IDR, JPY, SGD dan THB

5. KESIMPULAN

Penerapan jaringan saraf untuk peramalan membutuhkan waktu yang tidak sedikit karena perlu melakukan banyak percobaan dalam menetapkan jumlah *hidden layer*, menetapkan jumlah *neuron* dalam *hidden layer*, penentuan besarnya *learning rate* serta menerapkan teknik pembelajaran pada jaringan yang direncanakan. Kombinasi stuktur yang dihasilkan sekarang pun masih belum bisa dikatakan sebagai hasil terbaik dari *performance* maksimum dari jaringan saraf.

Jaringan saraf memiliki hasil yang lebih baik dalam meredam *error* yang terjadi akibat adanya perubahan mendadak pada data *non stasioner* dan *non homogen*, seperti terlihat pada perbandingan *plot residual vs order*, walaupun terkadang *MAD* maupun *MSE*-nya tidak lebih baik dari metode *GARCH(1,1)* yang mampu memberikan *fitting* yang cukup bagus untuk *heteroskedastik time series*.

Dalam mendapatkan hasil maksimal, perlu pengamatan dan analisa *weight* dan *bias* untuk tiap-tiap *input* yang hendak ditetapkan. Signifikansi *input* yang tinggi akan membantu jaringan dalam mempercepat proses kalkulasi sehingga dapat menghemat waktu.

DAFTAR PUSTAKA

- Bambang D.P., Budi, et.al., 1999. Teknik Jaringan Syaraf Tiruan Feedforward Untuk Prediksi Harga Saham pada Pasar Modal Indonesia. *Jurnal Informatika*, Vol.1 No.1 Mei 1999, hal. 33-37. Program Pascasarjana Peran Teknik Kendali, Institut Teknologi Bandung.
- Box, G.E.P and G.M Jenkins, 1976. *Time Series Analysis Forecasting and Control*, Revised Edition, Holden-Day.
- Demuth, Howard, and Mark Beale, 1996. *Neural Network Toolbox for Use with MATLAB*, Mass: The Math Works Inc.
- Hagan, Martin T. et. al., 1996. *Neural Network Design*. Boston: PWS Publishing Company.
- MATLAB*, 1996. *High-Performance Numeric Computation and Visualization Software, User's Guide*. Nattick, MA : The Math Works Inc.
- MATLAB*, 1996. *High-Performance Numeric Computation and Visualization Software, Reference Guide*. Nattick, MA : The Math Works Inc.
- Pratama, T. Iwan B.,1999. Metode Peramalan Memakai Jaringan Saraf Buatan dengan Cara *Backpropagation*, *Jurnal Teknologi Industri*, Vol. III. No.2, hal 109-116.
- Resmana, Dwi Wiyanto, 1997. *Prediksi Nilai Tukar Valuta Asing: Sebuah Studi Kasus Penggunaan Jaringan Syaraf Tiruan Untuk Peramalan*, Edisi Pertama. Surabaya: Lembaga Penelitian dan Pengabdian Kepada Masyarakat Universitas Kristen Petra.
- Siana Halim, Jani Rahardjo, Shirley Adelia, 1999. Model Matematik untuk Menentukan Nilai Tukar Mata Uang Rupiah terhadap Dollar Amerika. *Jurnal Teknik Industri: Jurnal Keilmuan dan Aplikasi Teknik Industri*, Vol. I No.1, hal.30-40.