

**ANALISIS JARINGAN SARAF TIRUAN BACKPROPAGATION
TERHADAP PERAMALAN NILAI TUKAR MATA UANG RUPIAH DAN DOLAR**
Analysis of Backpropagation Artificial Neural Network to forecast Rupiah and Dollar

DORTEUS LODEWYIK RAHAKBAUW

Jurusan Matematika Fakultas MIPA Universitas Pattimura

Jl. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon

E-mail: lodewyik@gmail.com

ABSTRAK

Kurs atau nilai tukar mata uang. Jenis kurs ada tiga macam, yaitu kurs jual, kurs beli, dan kurs tengah. Kurs dibutuhkan untuk menentukan sesuatu yang perlu dilakukan yang berkaitan dengan kurs itu misalnya keputusan investasi jangka pendek, keputusan penganggaran modal, keputusan pembiayaan jangka panjang, dan penilaian laba. Oleh karena itu, perlu dilakukan upaya untuk memprediksi besarnya kurs untuk beberapa waktu ke depan. Permasalahan yang dihadapi adalah cara untuk memprediksi besarnya kurs yang menghasilkan nilai prediksi dengan tingkat kesalahan yang minimal. Peramalan merupakan suatu proses untuk memprediksi kejadian ataupun perubahan di masa yang akan datang. Dalam suatu proses kegiatan, proses peramalan ini merupakan awal dari suatu rangkaian kegiatan, dan sebagai titik tolak kegiatan berikutnya. Pemodelan *time series* seringkali dikaitkan dengan proses peramalan (*forecasting*) suatu nilai karakteristik tertentu pada periode kedepan, melakukan pengendalian suatu proses ataupun untuk mengenali pola perilaku sistem. Dengan mendeteksi pola dan kecenderungan data, kemudian memformulasikannya dalam suatu model, maka dapat digunakan untuk memprediksi data yang akan datang. Model dengan akurasi yang tinggi akan menyebabkan nilai prediksi cukup valid untuk digunakan sebagai pendukung dalam proses pengambilan keputusan.

Salah satu metode peramalan yang berkembang saat ini adalah menggunakan *Artificial Neural Network* (ANN), dimana ANN telah menjadi objek penelitian yang menarik dan banyak digunakan untuk menyelesaikan masalah pada beberapa bidang kehidupan, salah satu diantaranya adalah untuk analisis data *time series* pada masalah *Forecasting* (Loh, 2003). Salah satu jaringan yang sering digunakan untuk prediksi data *time series* adalah *Backpropagation neuron network*. Dalam penelitian ini akan dibahas mengenai penggunaan jaringan saraf tiruan backpropagation untuk memprediksi kurs jual Rupiah (Rp) per 1 dolar amerika (USD). Dalam penelitian ini akan dibagi sebanyak 70% dari data yang ada sebagai pelatihan dan 30% dari data sebagai data pengujian. Dan dalam penelitian ini digunakan data kurs bulan Oktober 2013-Januari 2014, yang diambil dari situs Bank Indonesia. Dalam proses penelitian diperoleh Learning rate yang digunakan untuk data harian adalah 0.5, proses epoch berhenti pada iterasi ke-27088 untuk data harian, dengan pencapaian gradient sebesar 0,0081822 dan nilai R untuk pelatihan data sebesar 0,99494 yang berarti sangat baik. Selanjutnya data di uji dan memperoleh R sebesar 0,48638 yang berarti masih dikatakan baik untuk memprediksi data uji. Beberapa hal yang mempengaruhi hasil penelitian juga seperti data histories yang digunakan untuk variable masukkan JST kurang banyak, data yang digunakan untuk memprediksi kurs tidak bisa mewakili sebagai faktor utama yang mempengaruhi nilai kurs, dan batas nilai kesalahan yang kurang kecil serta kesesuaian bobot dalam arsitektur jaringan.

Kata Kunci: *Backpropagation*, Jaringan Saraf Tiruan, Kurs

PENDAHULUAN

Kurs yang juga dikenal sebagai nilai tukar adalah rasio pertukaran antara dua mata uang yang berbeda negara. Atau dengan kata lain kurs dapat diartikan sebagai harga satu unit mata uang asing dinyatakan dalam mata uang domestik. Bank Sentral masing-masing negara dapat memilih sistem nilai tukar yang berbeda. Kurs tetap ditentukan oleh Bank Sentral (sebuah institusi yang memutuskan harga mata uang tersebut). Ada berbagai macam faktor yang mempengaruhi nilai tukar, seperti suku bunga, inflasi, dan keadaan politik dan ekonomi di setiap negara. Biasanya dalam pasar forex, atau dalam perdagangan mata uang pasar internasional, terdiri dari jaringan agen yang saling berhubungan. Serta melibatkan banyak agen, seperti investor individu, investor institusi, bank sentral, bank komersial, dll.

TINJAUAN PUSTAKA

Jaringan syaraf tiruan telah dikembangkan sejak tahun 1940. Pada tahun 1943 McCulloch dan W.H.Pitts memperkenalkan pemodelan matematis neuron. Tahun 1949, Hebb Mencoba mengkaji proses belajar yang dilakukan oleh neuron. Teori ini dikenal sebagai Hebbian Law. Tahun 1958, Rosenblatt memperkenalkan konsep perseptron suatu jaringan yang terdiri dari beberapa lapisan yang saling berhubungan melalui umpan maju (*feed forward*). Konsep ini dimaksudkan untuk memberikan ilustrasi tentang dasar-dasar intelegensi secara umum. Hasil kerja Rosenblatt yang sangat penting adalah *perceptron convergence theorem* (tahun 1962) yang membuktikan bahwa bila setiap perseptron dapat memilah-milah dua buah pola yang berbeda maka siklus pelatihannya dapat dilakukan dalam jumlah yang terbatas. Pada tahun 1960 Widrow dan Hoff menemukan ADALINE (*Adaptive Linear Neuron*). Alat ini dapat beradaptasi dan beroperasi secara linier. Penemuan ini telah memperlebar aplikasi jaringan syaraf tiruan tidak hanya untuk pemilihan pola, tetapi juga untuk pengiriman sinyal khususnya dalam bidang *adaptive filtering*. Tahun 1969, Minsky dan Papert melontarkan suatu kritikan tentang kelemahan perseptronnya Rosenblatt di dalam memilah-milah pola yang tidak linier. Sejak saat itu penelitian di bidang jaringan syaraf tiruan telah mengalami masa vakum untuk kurang lebih satu dasawarsa. Tahun 1982, Hopfield telah memperluas aplikasi JST untuk memecahkan masalah-masalah optimasi. Hopfield telah berhasil memperhitungkan fungsi energi ke dalam jaringan syaraf yaitu agar jaringan memiliki kemampuan untuk mengingat atau memperhitungkan suatu obyek dengan obyek yang pernah dikenal atau diingat sebelumnya (*associative memory*). Konfigurasi jaringan yang demikian dikenal sebagai *recurrent network*. Salah satu aplikasinya adalah *Travelling Salesman Problem* (TSP).

Pada tahun 1986 Rumelhart, Hinton dan William menciptakan suatu algoritma belajar yang dikenal sebagai propagasi balik (*backpropagation*). Bila algoritma ini diterapkan pada perseptron yang memiliki lapisan banyak (*multi layer perceptron*), maka dapat dibuktikan bahwa pemilahan pola-pola yang tidak linier dapat diselesaikan

sehingga dapat mengatasi kritikan yang dilontarkan oleh Minsky dan Papert.

Definisi Jaringan Saraf Tiruan

Jaringan syaraf tiruan merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi [Siang, 2005 dalam Maru'ao, 2010]. Menurut [Sekarwati, 2005 dalam Maru'ao, 2010], jaringan syaraf tiruan merupakan sistem komputasi yang didasarkan atas pemodelan sistem syaraf biologis (*neurons*) melalui pendekatan dari sifat-sifat komputasi biologis (*biological computation*). Menurut [Subiyanto, 2002 dalam Maru'ao, 2010], jaringan syaraf tiruan adalah membuat model sistem komputasi yang dapat menirukan cara kerja jaringan syaraf biologi. Selain itu [menurut Siang, 2005 dalam Maru'ao, 2010], jaringan syaraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi dengan asumsi sebagai berikut.

- Pemrosesan informasi terjadi pada banyak elemen sederhana (*neurons*).
- Sinyal dikirimkan diantara *neuron-neuron* melalui penghubung - penghubung.
- Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal
- Untuk menentukan keluaran (*output*), setiap *neuron* menggunakan fungsi aktivasi yang dikenakan pada penjumlahan masukan (*input*) yang diterima. Besarnya keluaran (*output*) ini selanjutnya dibandingkan dengan suatu batas ambang.

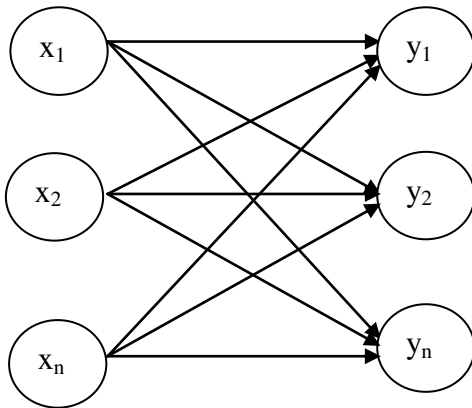
Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian jaringan syaraf tiruan (Siang 2005:23). *Neuron* terdiri dari 3 elemen pembentuk sebagai berikut.

- Himpunan unit-unit yang dihubungkan dengan jalur koneksi.
- Suatu unit penjumlah yang akan menjumlahkan masukan-masukan sinyal yang sudah dikalikan dengan bobotnya.
- Fungsi aktivasi yang akan menentukan apakah sinyal dari *input neuron* akan diteruskan ke *neuron* lain atau tidak

Arsitektur Jaringan (Konfigurasi Jaringan)

Berdasarkan arsitekturnya, model jaringan syaraf tiruan digolongkan menjadi:

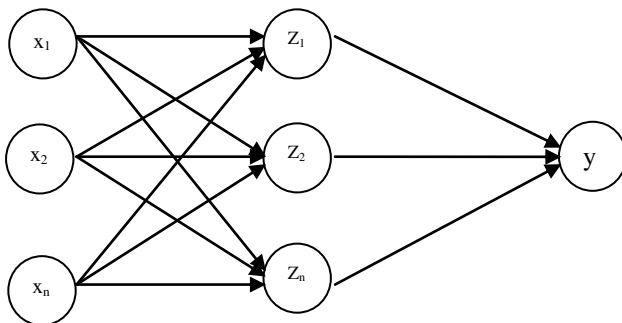
- Jaringan Layar Tunggal (*Single Layer Network*)
Pada jaringan ini, sekumpulan masukan neuron dihubungkan langsung dengan sekumpulan keluarannya. Sinyal mengalir searah dari layar (lapisan) masukan sampai layar (lapisan) keluaran. Setiap simpul dihubungkan dengan simpul lainnya yang berada di atasnya dan dibawahnya, tetapi tidak dengan simpul yang berada pada lapisan yang sama. Model yang masuk kategori ini antara lain : ADALINE, Hopfield, Perceptron, LVQ, dan lain-lain. Pada Gambar 1 diperlihatkan arsitektur jaringan layar tunggal
Dengan n buah masukan (x_1, x_2, \dots, x_n) dan m buah keluaran (y_1, y_2, \dots, y_m)



Gambar 1. Jaringan Layer Tunggal

b. Jaringan Layer Jamak (*Multiple Layer Network*)

Jaringan ini merupakan perluasan dari jaringan layer tunggal. Dalam jaringan ini, selain unit masukan dan keluaran, ada unit-unit lain (sering disebut layer tersembunyi). Dimungkinkan pula ada beberapa layer tersembunyi. Model yang termasuk kategori ini antara lain : MADALINE, *backpropagation*. Pada Gambar 2 diperlihatkan jaringan dengan n buah unit masukan (x_1, x_2, \dots, x_n), sebuah layer tersembunyi yang terdiri dari m buah unit (z_1, z_2, \dots, z_m) dan 1 buah unit keluaran.



Gambar 2. Jaringan Layer Jamak

c. Jaringan Reccurent

Model jaringan *reccurent* (*reccurent network*) mirip dengan jaringan layer tunggal ataupun jamak. Hanya saja, ada simpul keluaran yang memberikan sinyal pada unit masukan (sering disebut *feedback loop*). Dengan kata lain sinyal mengalir dua arah, yaitu maju dan mundur.
Contoh : *Hopfield network, Jordan network, Elmal network*.

Jaringan Saraf Tiruan Backpropagation

Backpropagation merupakan model jaringan syaraf tiruan dengan layer jamak. Seperti halnya model jaringan syaraf tiruan lainnya, *backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

a. Fungsi Aktivasi pada *Backpropagation*

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat sebagai berikut.

1. Kontinu.
2. Terdiferensial dengan mudah.
3. Merupakan fungsi yang tidak turun.

Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi sigmoid biner yang memiliki *range* (0,1). Fungsi sigmoid biner didefinisikan sebagai berikut.

$$f_1(x) = \frac{1}{1 + \exp(-x)}$$

dengan

$$f_1'(x) = f_1(x)[1 - f_1(x)]$$

Fungsi lain yang sering dipakai adalah fungsi sigmoid bipolar dengan *range* (-1,1) yang didefinisikan sebagai berikut.

$$f_2(x) = \frac{2}{1 + \exp(-x)} - 1$$

dengan

$$f_2'(x) = \frac{1}{2} [1 + f_2(x)] [1 - f_2(x)]$$

Fungsi sigmoid memiliki nilai maksimum 1. Untuk pola yang targetnya lebih dari 1, pola masukan dan keluaran harus terlebih dahulu ditransformasi sehingga semua polanya memiliki *range* yang sama seperti fungsi sigmoid yang dipakai.

Alternatif lain adalah menggunakan fungsi aktivasi sigmoid hanya pada layer yang bukan layer keluaran. Pada layer keluaran, fungsi aktivasi yang dipakai adalah fungsi identitas $f(x) = x$.

b. Pelatihan *Backpropagation*

Pelatihan *backpropagation* meliputi 3 fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layer masukan hingga layer keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagation mundur, dimulai dari garis yang berhubungan langsung dengan unit – unit di layer keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

Pelatihan *backpropagation* meliputi 3 fase sebagai berikut. Fase 1, yaitu propagasi maju.

Selama propagasi maju, sinyal masukan ($= x_i$) dipropagasikan ke layer tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layer tersembunyi ($= z_j$) tersebut selanjutnya dipropagasikan maju lagi ke layer tersembunyi di atasnya menggunakan fungsi aktivitas yang ditentukan. Demikian seterusnya hingga menghasilkan keluaranjaringan ($= y_k$). Berikutnya, keluaran jaringan ($= y_k$) dibandingkan dengan target yang harus dicapai ($= t_k$). Selisih $t_k -$

y_k adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

2. Fase 2, yaitu propagasi mundur.

Berdasarkan kesalahan $t_k - y_k$, dihitung faktor δ_k ($k = 1, 2, \dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit y_k ke semua unit tersembunyi yang terhubung langsung dengan y_k . δ_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor δ_j di setiap unit layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar dibawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

3. Fase 3, yaitu perubahan bobot.

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamman. Perubahan bobot suatu garis didasarkan atas faktor δ neuron dilayar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas δ_k yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

Algoritma pelatihan untuk jaringan *backpropagation* dengan satu layar tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut.

Step 0 : Inisialisasi nilai bobot dengan nilai acak yang kecil.

Step 1 : Selama kondisi berhenti masih tidak terpenuhi, laksanakan *step 2 - 9*.

Step 2 : Untuk tiap pasangan pelatihan, kerjakan *step 3-8*.

Feedforward :

Step 3 : Untuk tiap unit *input* ($X_i, i=1, \dots, n$) menerima sinyal *input* x_i dan menyebarkan sinyal itu keseluruh unit pada *layer* atasnya (*hidden layer*)

Step 4 : Untuk tiap unit dalam ($Z_j, j=1, \dots, p$) dihitung nilai *input* dengan menggunakan nilai bobotnya

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

Kemudian dihitung nilai *output* dengan menggunakan fungsi aktivasi yang dipilih

$$z_j = f(z_in_j)$$

Hasil fungsi tersebut dikirim ke semua unit pada *layer* di atasnya (unit *output*).

Step 5 : Untuk tiap unit *output* ($Y_k, k=1, \dots, m$) dihitung nilai *input* dengan menggunakan nilai bobotnya

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

Kemudian dihitung nilai *output* dengan menggunakan fungsi aktivasi

$$y_k = f(y_in_k)$$

Backpropagation Of Error

Step 6 : Untuk tiap unit *output* ($Y_k, k=1, \dots, m$) menerima pola target yang bersesuaian dengan pola *input* dan kemudian dihitung informasi kesalahan

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

Kemudian dihitung koreksi nilai bobot yang kemudian akan digunakan untuk memperbaharui nilai bobot w_{jk} :

$$\Delta w_{jk} = \alpha \delta_k z_j$$

Hitung koreksi nilai bias yang kemudian akan digunakan untuk memperbaharui nilai w_{0k} :

$$\Delta w_{0k} = \alpha \delta_k$$

dan kemudian nilai δ_k dikirim ke unit pada *layer* sebelumnya.

Step 7 : Untuk tiap unit dalam ($Z_j, j=1, \dots, p$) dihitung delta *input* yang berasal dari unit pada *layer* di atasnya :

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk}$$

Kemudian nilai tersebut dikalikan dengan nilai turunan dari fungsi aktivasi untuk menghitung informasi kesalahan :

$$\delta_j = \delta_in_j f'(z_in_j)$$

Hitung koreksi nilai bobot yang kemudian digunakan untuk memperbaharui nilai v_{ij}

$$\Delta v_{ij} = \alpha \delta_j x_i$$

dan hitung nilai koreksi bias yang kemudian digunakan untuk memperbaharui v_{0j}

$$\Delta v_{0j} = \alpha \delta_j$$

Update nilai bobot dan bias :

Step 8 : Tiap unit *output* ($Y_k, k=1, \dots, m$) mengupdate bias dan bobotnya ($j=0, \dots, p$)

$$w_{jk}(\text{new}) = w_{jk}(\text{old}) + \Delta w_{jk}$$

Tiap unit *hidden* ($Z_j, j=1, \dots, p$) mengupdate bias dan bobotnya ($i=0, \dots, n$)

$$v_{ij}(\text{new}) = v_{ij}(\text{old}) + \Delta v_{ij}$$

Step 9 : Menguji apakah kondisi berhenti sudah terpenuhi.

Parameter α merupakan laju pemahaman yang menentukan kecepatan iterasi. Nilai α terletak antara 0 dan 1 ($0 \leq \alpha \leq 1$). Semakin besar harga α , semakin sedikit iterasi yang dipakai. Akan tetapi jika harga α terlalu besar,

maka akan merusak pola yang sudah benar sehingga pemahaman menjadi lambat. Satu siklus pelatihan yang melibatkan semua pola disebut *epoch*.

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf tiruan dalam mencapai minimum global (atau mungkin lokal saja) terhadap nilai *error* (kesalahan) dan cepat tidaknya proses pelatihan menuju kekonvergenan.

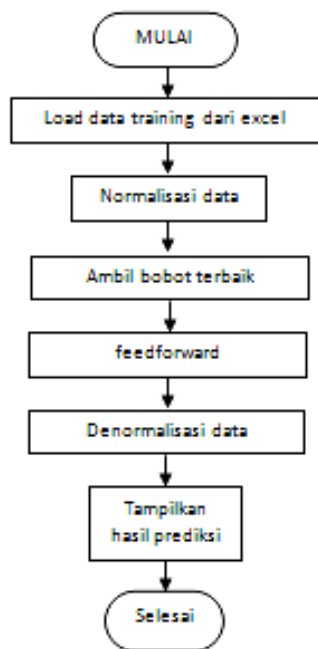
Apabila bobot awal terlalu besar maka *input* (masukan) ke setiap lapisan tersembunyi atau lapisan *output* (keluaran) akan jatuh pada daerah dimana turunan fungsi sigmoidnya akan sangat kecil. Apabila bobot awal terlalu kecil, maka *input* (masukan) ke setiap lapisan tersembunyi atau lapisan *output* (keluaran) akan sangat kecil. Hal ini akan menyebabkan proses pelatihan berjalan sangat lambat. Biasanya bobot awal diinisialisasi secara random dengan nilai antara -0.5 sampai 0.5 (atau -1 sampai 1 atau interval yang lainnya).

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola.

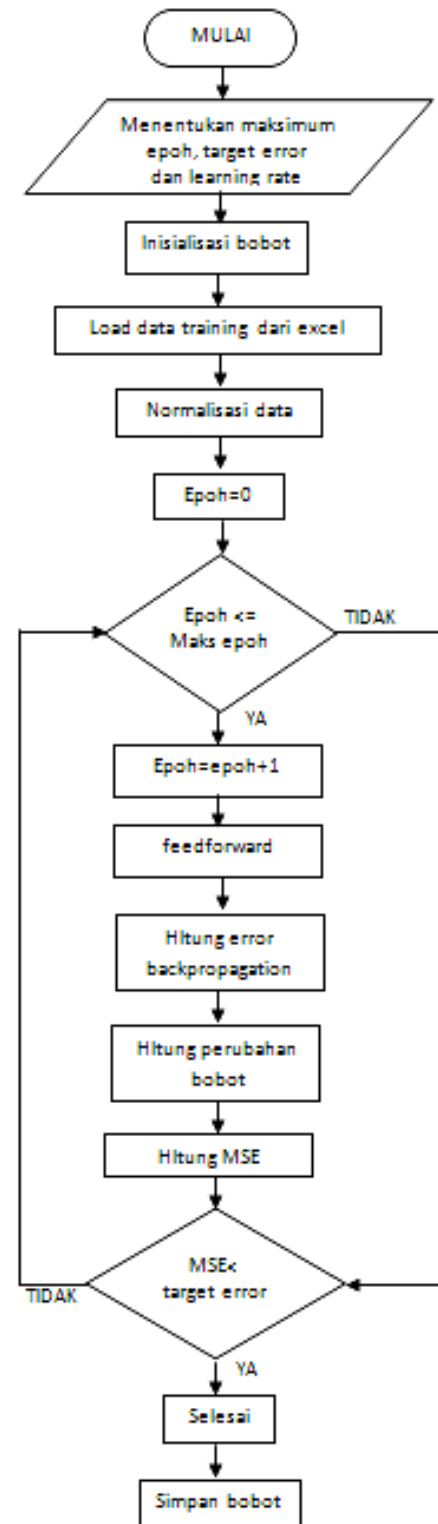
Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

Diagram Alir Sistem (Flowchart)

Diagram alir sistem digunakan untuk menggambarkan langkah kerja sistem yang akan dibuat dan juga akan digunakan oleh penulis untuk menentukan langkah-langkah berikutnya.. Data yang digunakan adalah data sekunder dari harga kurs USD/AUD. Input data kurs ini berdasarkan pada variabel-variabel data harga pembukaan, harga penutupan, harga tertinggi, harga terendah, volume. Gambar 4 menunjukkan flowchart pelatihan jaringan syaraf tiruan untuk 1 arsitektur jaringan, untuk arsitektur yang lain caranya juga sama. Pelatihan dilakukan dengan data yang banyak yang bisa mewakili keadaan kurs yang terjadi.



Gambar 3. Alur kerja forecasting



Gambar 4. Flowchart algoritma Backpropagation

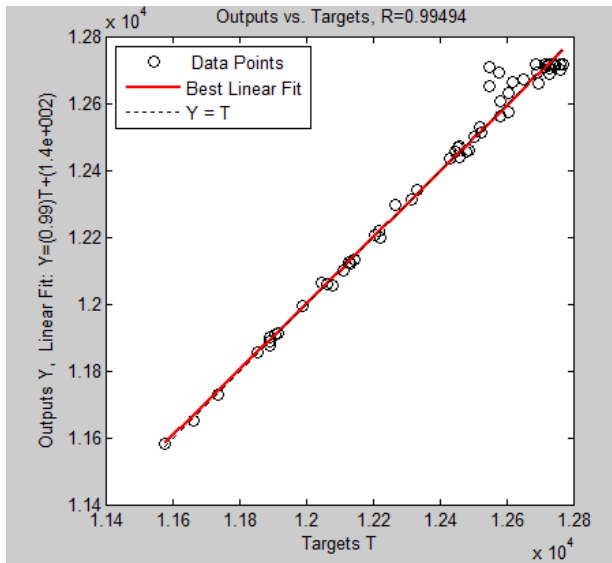
HASIL DAN PEMBAHASAN

Perancangan Arsitektur Jaringan

Arsitektur jaringan adalah jumlah layer dan neuron pada input, hidden dan output. Dalam permasalahan ini untuk memprediksi harga kurs USD terhadap Rupiah, menggunakan jaringan syaraf dengan momentum. Arsitektur jaringan saraf ini terdiri dari lapisan input, 2

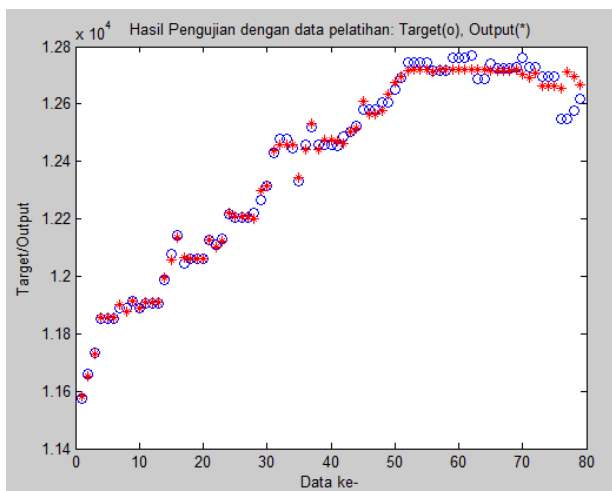
lapisan tersembunyi dan lapisan output. Lapisan tersembunyi terdiri atas 10 neuron dengan fungsi aktivasi tansig, lapisan tersembunyi yang kedua terdiri dari 5 neuron dengan fungsi aktivasi logsig, sedangkan lapisan output terdiri dari 1 neuron dengan fungsi aktivasi purelin.

Analisis Hasil Pelatihan

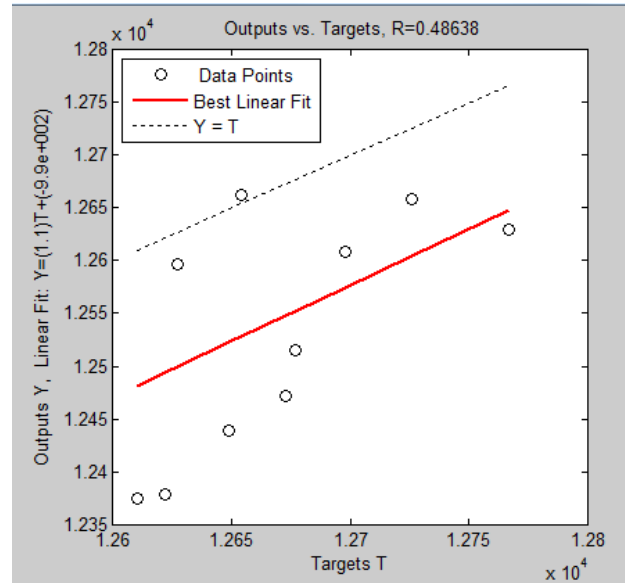


Gambar 5. Hasil Training data harian

Dalam proses training penelitian diperoleh Learning rate yang digunakan untuk data harian adalah 0.5, proses epoch berhenti pada iterasi ke-27088 untuk data harian, dengan pencapaian gradient sebesar 0,0081822 dan nilai R untuk pelatihan data sebesar 0,99494 (Gambar 5) yang berarti sangat baik karena mendekati nilai 1.



Gambar 6.



Gambar 7.

Selanjutnya data di uji dan memperoleh R sebesar 0,48638 yang berarti masih dikatakan baik untuk memprediksi data uji.

KESIMPULAN

Algoritma backpropagation dapat melakukan proses prediksi, akan tetapi baik atau tidaknya nilai yang dihasilkan sangat dipengaruhi oleh penentuan parameter seperti besarnya learning rate dan jumlah neuron pada hidden layer. Untuk menghasilkan konfigurasi parameter yang baik diperlukan waktu cukup lama dalam melakukan eksperimen mencari parameter yang terbaik yang nantinya parameter yang baru tersebut dapat dipakai untuk proses prediksi.

Dalam proses training penelitian diperoleh Learning rate yang digunakan untuk data harian adalah 0.5, proses epoch berhenti pada iterasi ke-27088 untuk data harian, dengan pencapaian gradient sebesar 0,0081822 dan nilai R untuk pelatihan data sebesar 0,99494 yang berarti sangat baik karena mendekati nilai 1. Selanjutnya data di uji dan memperoleh R sebesar 0,48638 yang berarti masih dikatakan baik untuk memprediksi data uji.

DAFTAR PUSTAKA

Fausett, Laurent, “*Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*”, Prentice-Hall, Inc., New Jersey, 1994.

Kusumadewi, S, “*Membangun Jaringan Syaraf Tiruan menggunakan Matlab & Excel link*”, Graha Ilmu, 2004

Maru'ao, “*Neural Network Implementation in Foreign Exchange Kurs Prediction*”, Gunadarma University, 2010.