

Peramalan Nilai Tukar Mata Uang Dollar AS terhadap Rupiah Menggunakan Neural Network Ensemble Bagging

Anang Ashari Romdhoni

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No 36 A, Surakarta
anang005@gmail.com

Wiharto

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No 36 A, Surakarta
wi_harto@yahoo.com

Esti Suryani

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No 36 A, Surakarta
suryapalapa@yahoo.com

ABSTRAK

Artikel ini membahas tentang penggunaan Neural Network Ensembles dengan menggunakan metode Levenberg-Marquardt pada pelatihan masing-masing subsample dan metode Bagging pada penggabungannya untuk meningkatkan hasil dari prediksi model time series. Neural Network Ensembles diimplementasikan untuk memprediksi nilai tukar mata uang dollar AS terhadap rupiah. Pada NNE ini menggunakan metode partisi sistematis. Metode mean imputation digunakan untuk menghilangkan adanya noise data yang berguna untuk memaksimalkan hasil MSE. Pada proses pengumpulan data, data dipartisi menjadi dua bagian, yaitu data pelatihan dan data pengujian dengan komposisi 60:40. Hasil menunjukkan bahwa model NNE dengan jumlah neuron input 2, neuron tersembunyi 5 dan neuron output 1 menghasilkan MSE paling kecil.

Kata kunci: Bagging, Levenberg-Marquardt, Neural Network Ensemble, , time series

1. PENDAHULUAN

Nilai tukar mata uang suatu negara merupakan salah satu indikator penting dalam suatu perekonomian. Nilai dari harga sebuah mata uang dari suatu negara yang diukur atau dinyatakan dalam mata uang lainnya disebut nilai tukar mata uang [1]. Nilai tukar juga mempunyai implikasi yang luas, baik dalam konteks ekonomi domestik maupun internasional, mengingat hampir semua negara di dunia melakukan transaksi internasional.

Semua negara tidak dapat mencukupi semua kebutuhan konsumsinya dari hasil produksi sendiri, meskipun ada pula beberapa komoditi yang hasilnya melebihi kebutuhan dalam negeri sehingga dapat diekspor. Oleh karena itu suatu bangsa pasti memerlukan mata uang asing dalam transaksi internasionalnya.

Dollar AS merupakan mata uang yang dominan (*hard currency*) yaitu mata uang yang bisa diterima secara luas sebagai bukti pembayaran internasional [2], sehingga banyak negara menggunakan mata uang dollar AS.

Mata uang memungkinkan mengalami penurunan dan kenaikan setiap saat. Penurunan dan kenaikan mata uang tersebut dipengaruhi oleh beberapa faktor, seperti jumlah permintaan barang dan jasa, tingkat inflasi, tingkat bunga, pengharapan pasar, dan intervensi Bank Sentral berdampak signifikan terhadap fluktuasi nilai tukar [3]. Fluktuasi-fluktuasi pada nilai tukar mata uang tersebut menjadi sebuah resiko bagi investor, perusahaan dan kalangan perbankan dalam transaksi internasional. Oleh karena itu, untuk mengurangi resiko tersebut diperlukan peramalan nilai tukar mata uang Dollar AS terhadap Rupiah.

Peramalan merupakan suatu usaha untuk meramalkan keadaan dimasa yang akan datang melalui pengujian keadaan dimasa lalu [4]. Selama ini banyak peramalan dilakukan secara intuitif menggunakan metode-metode statistika seperti metode *smoothing*, Box-Jenkins, ekonometri, regresi, jaringan syaraf tiruan dan sebagainya. Pemilihan metode tersebut tergantung pada berbagai aspek, yaitu aspek waktu, pola data, tipe model sistem yang diamati, tingkat keakuratan ramalan yang diinginkan dan sebagainya. Penelitian sebelumnya menyatakan bahwa peramalan data keuangan *time series* menggunakan jaringan syaraf tiruan lebih baik daripada statistika klasik dan metode lainnya [5].

Jaringan syaraf tiruan sederhana pertama kali diperkenalkan oleh McCulloch dan Pitts di tahun 1943. McCulloch dan Pitts menyimpulkan bahwa kombinasi beberapa neuron sederhana menjadi sistem neural akan meningkatkan kemampuan komputasinya [6]. Peneliti terdahulu hanya menggunakan jaringan dengan *layer tunggal (single layer)*. Penelitian sebelumnya mengembangkan *perceptron* menjadi *Backpropagation*, yang memungkinkan jaringan diproses melalui beberapa *layer*[7]. Algoritma pembelajaran syaraf tiruan yang menggunakan beberapa *layer* tersebut dapat menyelesaikan permasalahan model *time series*[8]. Akan tetapi, *overfitting* dapat terjadi pada jaringan syaraf tiruan jika dalam pengambilan sampel pada jaringan syaraf tiruan terdapat kesalahan. *Overfitting* yaitu kondisi dimana model yang dibuat hanya menghasilkan output yang baik untuk data yang dilatih saja dan tidak untuk data yang divalidasi [9].

Kondisi *overfitting* pada jaringan syaraf tiruan dapat diatasi dengan menggabungkan beberapa jaringan syaraf tiruan menjadi *Neural Network Ensembles (NNE)*[10]. NNE merupakan penggabungan dari beberapa jaringan syaraf tiruan dimana pembentukan NNE adalah dengan mempartisi data pelatihan menjadi beberapa *subsample* berdasarkan *n-input*. Penelitian sebelumnya tentang NNE telah dilakukan oleh Dewi dan Sun. Data pelatihan dipartisi menjadi beberapa *subsample* kemudian melatih masing-masing *subsample* dengan metode Levenberg-Marquardt lalu melakukan penggabungan dengan metode *backpropagation* [11]. Selain itu, penelitian ini juga melatih masing-masing *subsample* dengan metode Levenberg-Marquardt lalu melakukan penggabungan dengan metode *simple average*[11]. Akan tetapi, penggunaan metode *backpropagation* tersebut hasil MSE yang didapat belum maksimal. Pada metode *simple average*, hasil MSE lebih baik dibanding dengan *backpropagation*, tetapi penggunaan metode *simple average* diyakini bahwa hasil MSE yang didapat belum maksimal.

Selain menggunakan *backpropagation* dan *simple*

average, Bagging juga dapat digunakan sebagai metode penggabungan. Dalam penelitian lain, Bagging mempunyai kemampuan yang baik pada percobaan yang [12]. Untuk memperbaiki hasil MSE yang sebelumnya menggunakan metode penggabungan *simple average*, dalam hal ini peneliti menelaah dan mengkaji ulang tentang NNE dengan metode Levenberg-Marquardt untuk melatih *subsample* dan menggunakan metode Bagging pada penggabungannya. Metode *mean imputation* digunakan untuk menghilangkan adanya *noisedata* yang berguna untuk memaksimalkan hasil MSE.

2. DASAR TEORI

2.1. Jaringan Syaraf Tiruan

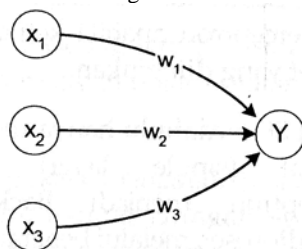
Jaringan syaraf tiruan merupakan salah satu sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi [6]. Jaringan syaraf tiruan mampu melakukan pengenalan kegiatan berbasis data masa lalu. Data masa lalu akan dipelajari oleh jaringan syaraf tiruan sehingga mempunyai kemampuan untuk memberikan keputusan terhadap data yang belum pernah dipelajari.

Jaringan syaraf tiruan ditentukan oleh 3 hal sebagai berikut [6]:

- a. Pola hubungan antar neuron (disebut arsitektur jaringan)
- b. Metode untuk menentukan bobot penghubung (disebut algoritma pelatihan/belajar)
- c. Fungsi aktivasi

Di dalam jaringan syaraf tiruan, istilah simpul (*node*) sering digunakan untuk menggantikan *neuron*. Setiap simpul pada jaringan menerima atau mengirim sinyal dari atau ke simpul-simpul lainnya. Pengiriman sinyal disampaikan melalui penghubung. Kekuatan hubungan yang terjadi antara setiap simpul yang saling terhubung dikenal dengan nama bobot.

Model-model jaringan syaraf tiruan ditentukan oleh arsitektur jaringan serta algoritma pelatihan. Arsitektur biasanya menjelaskan arah perjalanan sinyal atau data di dalam jaringan. Sedangkan algoritma belajar menjelaskan bagaimana bobot koneksi harus diubah agar pasangan masukan-keluaran yang diinginkan dapat tercapai. Perubahan harga bobot koneksi dapat dilakukan dengan berbagai cara, tergantung pada jenis algoritma pelatihan yang digunakan. Dengan mengatur besarnya nilai bobot ini diharapkan bahwa kinerja jaringan dalam mempelajari berbagai macam pola yang dinyatakan oleh setiap pasangan masukan-keluaran akan meningkat.



Gambar 1. Model sederhana Jaringan Syaraf Tiruan [6]

Simpul Y menerima masukan dari *neuron* x1, x2 dan x3 dengan bobot hubungan masing-masing adalah w1, w2 dan w3. Argumen fungsi aktivasi adalah net masukan (kombinasi linear masukan dan bobotnya). Ketiga sinyal simpul yang ada dijumlahkan:

$$net = x1w1 + x2w2 + x3w3 \quad (1)$$

Besarnya sinyal yang diterima oleh Y mengikuti fungsi

aktivasi $y = f(net)$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot.

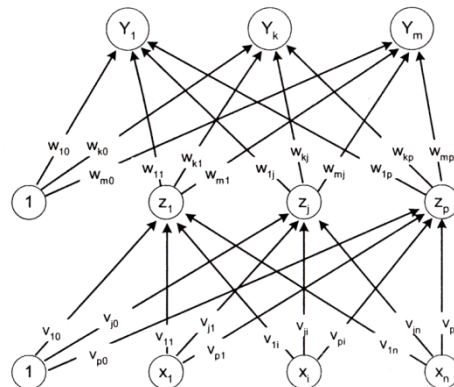
2.2. Backpropagation

Algoritma pelatihan Backpropagation atau bisa juga disebut propagasi balik, dikembangkan oleh Rumelhart 1986 [6]. Algoritma ini termasuk metode pelatihan supervised dan didesain untuk operasi pada jaringan feedforward multi *layer*.

2.3. Arsitektur Backpropagation

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih *layer* tersembunyi. Gambar 5 adalah arsitektur *backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah *layer* tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran.

v_{ji} merupakan bobot garis dari unit masukan x_i ke unit layer tersembunyi z_j (v_{j0} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layer tersembunyi z_j). w_{kj} merupakan bobot dari unit layer tersembunyi z_j ke unit keluaran y_k (w_{k0} merupakan bobot dari bias di layer tersembunyi ke unit keluaran y_k).



Gambar 2. Arsitektur Backpropagation[6]

2.4. Pelatihan Standar Backpropagation

Pelatihan *Backpropagation* meliputi tiga fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar 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 dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

- a. Fase I : Propagasi maju

Selama propagasi maju, sinyal masukan ($=x_i$) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi ($=z_j$) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ($=y_k$).

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.

b. Fase II : 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 di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

c. Fase III : Perubahan bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di layar 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.

2.5. Levenberg-Marquardt

Algoritma Levenberg-Marquadt dirancang dengan menggunakan pendekatan turunan kedua tanpa harus menghitung matriks Hessian. Apabila jaringan syaraf *feedforward* menggunakan fungsi kinerja *sum of square*, maka matriks Hessian dapat didekati sebagai

$$H = J^*J \tag{2}$$

dan gradien dapat dihitung sebagai

$$gW = J^*e \tag{3}$$

Dengan J adalah matriks Jacobian yang berisi turunan pertama dari error jaringan terhadap bobot, dan e adalah suatu vektor yang berisi *error* jaringan. Matriks Jacobian dapat dihitung dengan teknik *backpropagation* standart, yang tentu saja lebih sederhana dibanding dengan menghitung matriks Hessian [13].

2.6. Bagging

Bagging merupakan salah satu metode dari algoritma *machinelearning* yang merupakan metode yang berdasar pada *ensemblemethode*, yaitu metode yang menggunakan kombinasi dari beberapa model. Bagging prediktor adalah metode yang digunakan untuk membangkitkan *multiple versions* dari prediktor dan menggunakannya untuk mendapatkan kumpulan prediktor. *Multiple versions* dibentuk dengan replikasi *bootstrap* dari sebuah data percobaan [14]. Algoritma pada bagging merujuk pada algoritma *Bootstrap* yaitu pengambilan sampel dengan pengembalian pada data training sehingga terbentuk variasi data baru. Berikut ini merupakan algoritma Bagging.

B adalah *number data*

L adalah *base learner* (Tree)

Bagging ($examples, B, L$)

1. for i 1 to B

2. $examples_i$ (sampel *bootstrap* dari $examples$)

3. h_i (menerapkan L to $examples_i$)

4. return $h_1, h_2, h_3, \dots, h_B$

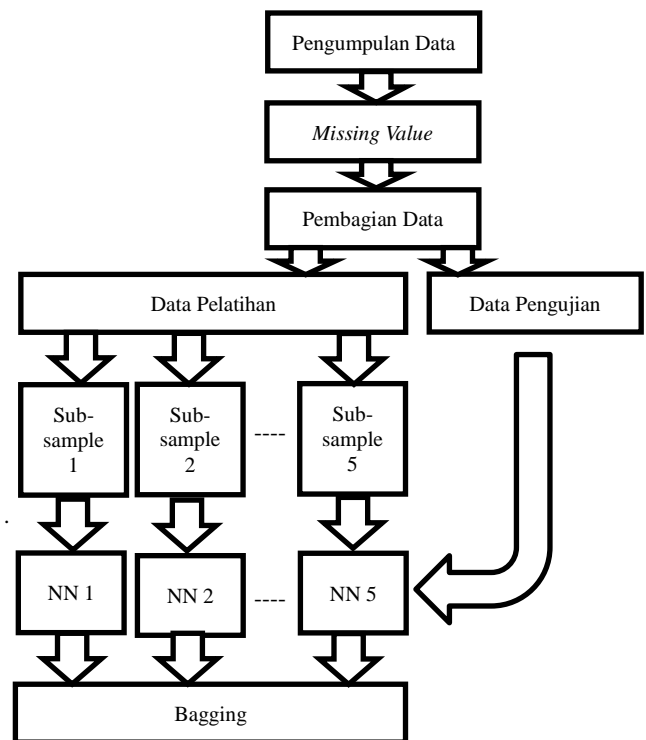
Langkah pertama yang dilakukan adalah membentuk sampel *bootstrap* dengan pengambilan sampel dengan penggantian. *Error* pada *learning* berasal dari *noisedata*, bias dan varians. Pengambilan sampel *bootstrap* mensimulasikan varians sampling. Setelah dilakukan pembentukan sampel *Bootstrap*, tiap sampel dilakukan *training* dengan *base learner*. *Averaging* pada sampel *bootstrap* dilakukan setelah *training* dengan *base learner*. dapat mengurangi kesalahan dari varians, terutama ketika perubahan kecil dalam training dapat menghasilkan perubahan besar pada parameter model [15].

2.7. Missing Value

Terdapat berbagai metode untuk menangani permasalahan *missingvalue* dalam analisis statistik. Salah satunya adalah menggunakan metode *mean imputation*. *Mean imputation* termasuk dalam kategori prosedur berbasis imputasi yaitu suatu alternatif metode yang umum dan fleksibel [16]. Dalam prosedur ini, *missing value* diisi baik dengan menduga langsung atau menggunakan penduga berbasis korelasi. Metode ini menangani *missing value* dengan cara yaitu dimana nilai yang hilang diganti oleh rata-rata (*mean*) dari kelompok sampel unit terkait. Dengan kata lain *missingvalue* dapat diganti dengan nilai rata-rata dari variabel yang bersangkutan [17].

3. METODOLOGI PENELITIAN

Bagan metodologi penelitian ini adalah seperti gambar 3 di bawah ini



Gambar 3. Bagan Metodologi Penelitian

Adapun rincian langkah-langkah pada bagan gambar 3 adalah

1. melakukan pengumpulan data yaitu data nilai tukar mata uang dollar AS terhadap Rupiah mulai

bulan 5 Januari 2009 – 27 Desember 2013 dari website. Data yang digunakan adalah kurs tengah. <http://www.bi.go.id/id/moneter/informasi-kurs/transaksi-bi/Default.aspx>

2. melakukan *preprocessing* data (*missing value*) dengan metode *mean imputation* yaitu mengisi nilai kosong dengan nilai rata-rata nilai sebelumnya dan nilai sesudahnya,
3. melakukan partisi data menjadi data pelatihan dan data pengujian dengan perbandingan 60% : 40%,
4. melakukan partisi data secara sistematis seperti pada tabel 1 [18]. Data pelatihan dipartisi menjadi beberapa *subsample* yang bergantung pada *lag-input* yang digunakan dalam arsitektur jaringan,

Tabel 1. Partisi Sistematis

No	Subsample1		Subsample2	
	Input	Target	Input	Target
1	x_1, x_2	x_3	x_2, x_3	x_4
2	x_3, x_4	x_5	x_4, x_5	x_6
3	x_5, x_6	x_7	x_7, x_8	x_9
:	:	:	:	:

5. melakukan transformasi data dengan interval [0.1,0.9] menggunakan rumus

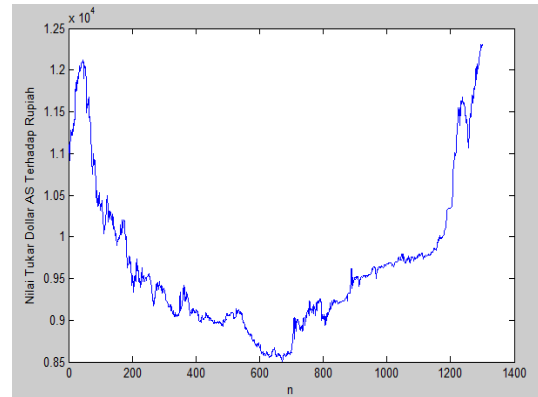
$$y'_i = \left[\frac{y_i - y_{min}}{y_{maks} - y_{min}} (a - b) \right] + b$$

6. melakukan pelatihan masing-masing sub sampel dengan algoritma pembelajaran *backpropagation* tipe *Levenberg-Marquardt* (LM). Pada pelatihan data *subsample*, target eror yang diharapkan sebesar 0,0001 dan maksimum *epoch* sebanyak 1000,
7. menggabungkan hasil pelatihan dari masing-masing *subsample* dengan metode Bagging. Pada proses metode Bagging, ada beberapa prosedur yaitu
 - i. hasil pelatihan dibuat beberapa sampel *Bootstrap*.
 - ii. tiap sampel dilakukan *training* dengan *base learner*.
 - iii. *output* yang dihasilkan dilakukan *averaging* (rata-rata).
8. melakukan percobaan arsitektur jaringan dengan menggunakan jumlah *neuron* pada lapisan input adalah $i = 2, \dots, 5$, jumlah *neuron* pada lapisan tersembunyi adalah $j = 2, \dots, 5$ dan jumlah *neuron output* 1,
9. mengetahui arsitektur jaringan yang optimum berdasarkan nilai terkecil dari MSE.

4. HASIL DAN PEMBAHASAN

Data nilai tukar mata uang Dollar AS terhadap Rupiah yang digunakan pada penelitian ini adalah data dari tanggal 5 Januari 2009 sampai dengan 27 Desember 2013 yang secara keseluruhan terdapat 1300 data. Pada gambar 4 merupakan data nilai tukar mata uang Dollar AS terhadap Rupiah yang digunakan pada penelitian ini adalah data dari tanggal 5 Januari 2009 sampai dengan 27 Desember 2013.

Gambar 4 merupakan grafik antara data nilai tukar mata uang Dollar AS terhadap Rupiah pada tanggal 5 Januari 2009 sampai dengan 27 Desember 2013 dengan n yaitu banyaknya data.



Gambar 4. Plot *time series* data nilai tukar mata uang Dollar AS terhadap Rupiah pada tanggal 5 Januari 2009 sampai dengan 27 Desember 2013

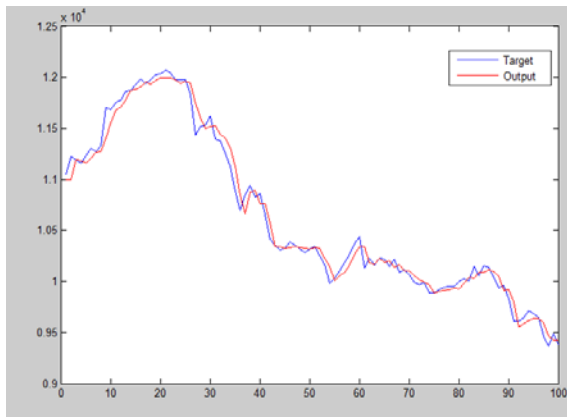
Data dibagi menjadi 2 bagian yaitu data untuk pelatihan dan data untuk pengujian. Data dari tanggal 5 Januari 2009 sampai tanggal 30 Desember 2011 yang terdapat 780 data digunakan untuk pelatihan. Sedangkan data dari tanggal 2 Januari 2012 sampai tanggal 27 Desember 2013 yang terdapat 520 data digunakan untuk pengujian.

Tabel 2 berikut ini merupakan perbandingan hasil dari pelatihan dan pengujian yang masing-masing menggunakan arsitektur jaringan yang sebelumnya sudah dituliskan.

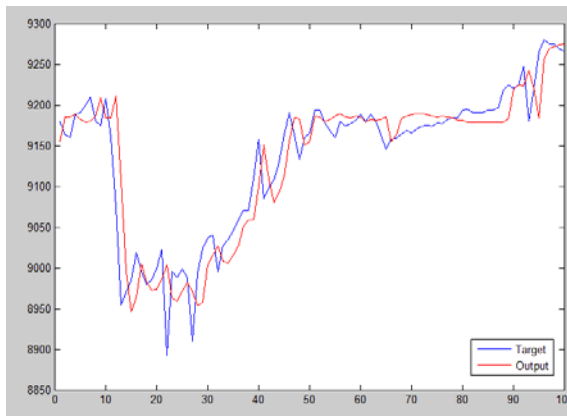
Tabel 2. Perbandingan hasil MSE pelatihan dan pengujian terhadap masing-masing arsitektur jaringan

Arsitektur Jaringan	MSE Pelatihan	MSE Pengujian
2-2-1	$1,70 \times 10^{-4}$	$1,70 \times 10^{-4}$
2-3-1	$1,69 \times 10^{-4}$	$1,75 \times 10^{-4}$
2-4-1	$1,64 \times 10^{-4}$	$1,67 \times 10^{-4}$
2-5-1	$1,63 \times 10^{-4}$	$1,60 \times 10^{-4}$
3-2-1	$1,86 \times 10^{-4}$	$1,91 \times 10^{-4}$
3-3-1	$1,77 \times 10^{-4}$	$1,93 \times 10^{-4}$
3-4-1	$1,73 \times 10^{-4}$	$1,87 \times 10^{-4}$
3-5-1	$1,73 \times 10^{-4}$	$1,85 \times 10^{-4}$
4-2-1	$2,53 \times 10^{-4}$	$3,31 \times 10^{-4}$
4-3-1	$2,51 \times 10^{-4}$	$3,24 \times 10^{-4}$
4-4-1	$2,54 \times 10^{-4}$	$3,34 \times 10^{-4}$
4-5-1	$2,50 \times 10^{-4}$	$3,25 \times 10^{-4}$
5-2-1	$2,9 \times 10^{-4}$	$4,5 \times 10^{-4}$
5-3-1	$3,15 \times 10^{-4}$	$4,90 \times 10^{-4}$
5-4-1	$2,95 \times 10^{-4}$	$4,93 \times 10^{-4}$
5-5-1	$2,93 \times 10^{-4}$	$4,78 \times 10^{-4}$

Dari tabel di atas dapat dilihat bahwa hasil dengan MSE terkecil pada pelatihan dan pengujian adalah dengan menggunakan arsitektur jaringan dengan unit *input*2, unit tersembunyi 5, dan unit *output*1. Gambar 5. berikut merupakan grafik pelatihan antara target dan output dan gambar 6. merupakan grafik pengujian target dan output.



Gambar 5. Grafik pelatihan



Gambar 6. Grafik pengujian

Tabel 3 berikut ini merupakan hasil peramalan nilai tukar mata uang Dollar AS terhadap Rupiah yang di dapat dari data pengujian.

Tabel 3. Hasil Peramalan Nilai Tukar Mata Uang Dollar AS Terhadap Rupiah daritanggal 2 Januari 2012 sampaitanggal 27 Desember 2013

Tanggal	Target	Output
4-Jan-12	9180	9152
5-Jan-12	9163	9190
6-Jan-12	9160	9190
9-Jan-12	9188	9193
10-Jan-12	9190	9186
11-Jan-12	9200	9181
12-Jan-12	9210	9181
13-Jan-12	9180	9192
14-Jan-12	9175	9206
:	:	:
:	:	:
24-Des-13	12215	12193
25-Des-13	12238	12193
26-Des-13	12238	12193
27-Des-13	12260	12193

Tabel 4. Perbandingan *framework ensembles* kasus peramalan

No.	Framework	Jumlah Data	Jumlah Subsample	MSE
1.	<i>Ensembles of Keep The Best model</i>	978	2	$1,44 \times 10^{-3}$ [18]
2.	<i>Neural Network Ensemble - Backpropagation</i>	840	2	$2,26 \times 10^{-3}$ [11]
3.	<i>Neural Network Ensemble - Simple average</i>	840	3	$2,11 \times 10^{-3}$ [11]
4.	<i>a. Neural Network Ensemble - Bagging</i>	1300	2	$1,60 \times 10^{-4}$
	<i>b. Neural Network Ensemble - Bagging</i>	978	2	$8,90 \times 10^{-4}$
	<i>c. Neural Network Ensemble - Bagging</i>	840	2	$1,69 \times 10^{-3}$

Tabel 4.10 merupakan gambaran umum *framework ensemble* yang digunakan pada kasus peramalan *time series*. Masing-masing terdapat *MSE* yang didapat dari masing-masing jurnal.

Penelitian yang pertama menggunakan *framework Ensembles of Keep The Best model* diimplementasikan pada kasus nilai tukar mata uang Dollar AS terhadap Pound sterling. Pada *framework* ini menggunakan partisi serial dan partisi sistematis menggunakan 978 data dengan data pelatihan berjumlah 782, data validasi berjumlah 104 dan data tes berjumlah 92 (Zhang dan Berardi, 2001).

Penelitian yang kedua menggunakan *framework Neural Network Ensemble-Backpropagation* dan *Neural Network Ensemble-Simple average* diimplementasikan pada kasus nilai tukar mata uang Dollar AS terhadap Rupiah. Pada *framework* ini menggunakan partisi sistematis menggunakan 840 data dengan komposisi data pelatihan, data tes, dan data validasi 60:20:20. Hasilnya dengan menggunakan *framework Neural Network Ensemble-Simple average* *MSE* yang dihasilkan lebih kecil dibandingkan menggunakan *framework Neural Network Ensemble-Backpropagation* (Dewi, 2012).

Terdapat beberapa variabel yang diuraikan pada masing-masing *framework* yaitu jumlah data dan jumlah *subsample*. Jumlah data merupakan keseluruhan data yang digunakan pada proses pelatihan, tes dan validasi.

5. KESIMPULAN DAN SARAN

Berdasarkan pembahasan diperoleh kesimpulan bahwa *NNE LM-Bagging* dengan *mean imputation* sebagai metode untuk mengatasi *missing value* didapatkan arsitektur jaringan yang optimum berdasarkan *MSE* yang paling kecil yaitu dengan menggunakan arsitektur jaringan dengan jumlah unit input 2, jumlah unit tersembunyi 5, dan jumlah unit output 1.

Pada penelitian ini data pelatihan di partisi menjadi beberapa *subsample* bergantung pada *lag-lag input* yang digunakan dalam arsitektur jaringan. Sehingga tidak bisa mencoba dengan menggunakan banyak *subsample*. Perlu dilakukan penambahan fitur sehingga jumlah *subsample* tidak bergantung pada *lag-lag input* yang digunakan.

7. DAFTAR PUSTAKA

- [1] Krugman, P.R. & Maurice, O. 1994. *Ekonomi Internasional: Teori dan Kebijakan*. Erlangga : Jakarta.
- [2] Kuncoro, M. 1996. *Manajemen Keuangan Internasional: pengantar ekonomi dan bisnis global*. BPFE : Yogyakarta.
- [3] Sartono, A. 2001. *Manajemen Keuangan*, BPFE : Yogyakarta.
- [4] Prasetya, H dan Lukiasuti, F . 2009. *Manajemen Operasi*. Media Pressindo : Yogyakarta.
- [5] Garliauskas, A. 1999. *Neural Network Chaos and Computational Algorithms of Forecast in Finance*. Proceedings of the IEEE SMC Conference on Systems, Man, and Cybernetics 2, 638-643. 12-15 October
- [6] Siang, J.J. 2005. *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan MATLAB*. Andi : Yogyakarta.
- [7] Fausett, L.,1994 *Fundamental Of Neural Networks*, Prantice Halt,
- [8] Moraga, C.,H. Allende and R. Salas, 1999. *Artificial Neural Networks in Time Series Forecasting: A Comparative Analysis*,
- [9] Wijayanti, D.T. 2013. *RBF and ARIMA Combined for Time Series Forecasting*. Seminar Nasional Aplikasi Teknologi Informasi. Yogyakarta.
- [10] Hansen, L.K. & Salamon, P. 1990. *Neural Network Ensembles*. IEEE.10, 993 – 1001.
- [11] Dewi, N, S. 2012. *Neural Network Ensembles Untuk Peramalan Nilai Tukar Dollar Terhadap Rupiah*. Jurusan Matematika FMIPA UNS.
- [12] Sun J, Liao B, dan Li H. 2013. *AdaBoost and Bagging Ensemble Approaches with Neural Network as Base Learner for Financial Distress Prediction of Chinese Construction and Real Estate Companies*. School of Economics and Management, Zhejiang Normal University.
- [13] Kusumadewi, S. 2004. *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*. Graha Ilmu : Yogyakarta.
- [14] Breiman, L. 1996. “*Bagging Predictors*,” *Machine Learning*. Vol. 24. 23-140.
- [15] Barutcuoglu dan Alpaydin. 2003. *A Comparison of Model Aggregation Methods for Regression*. Berlin : Springer-Verlag.
- [16] Budi. 2009. *Analisis Statistika Untuk Missing Data*. <http://statistikakomputasi.wordpress.com/2009/12/03/analisis-statistik-untuk-missing-data-introduction/>. Di akses pada tanggal 21 Januari 2014.
- [17] Santosa, S. 2007. *Buku Latihan SPSS Statistik Parametrik*. PT Alex Media Komputindo : Jakarta
- [18] Zhang, G.P dan Berardi, V.L. 2001. *Time Series Forecasting with Neural Network Ensembles: An Application for Exchange Rate Prediction*. Jstor.