

# PERAMALAN NILAI TUKAR RUPIAH TERHADAP DOLAR AMERIKA MENGUNAKAN JARINGAN SYARAF TIRUAN ( $p,d,q$ )

Gege Safet Yanto Raharjo<sup>1</sup>, Rian Febrian Umbara<sup>2</sup>, Danang Triantoro<sup>3</sup>

<sup>1,2,3</sup>Prodi Ilmu Komputasi-Telkom University, Bandung

<sup>1</sup>gegesafetyanto@students.telkomuniversity.ac.id, <sup>2</sup>rianfebrianumbara@telkomuniversity.ac.id,

<sup>3</sup>danangtriantoro@telkomuniversity.ac.id

---

## Abstrak

Nilai tukar merupakan tingkat dimana mata uang satu negara dapat ditukar dengan mata uang negara lain, memainkan peran penting dalam mengendalikan dinamika pasar valuta. Oleh karena itu, peramalan kurs menjadi hal yang menarik sebagai penentuan kebijakan. Model gabungan ARIMA dan JST disebut model Jaringan Syaraf Tiruan ( $p,d,q$ ). Hasil error dan differencing yang diperoleh dari model ARIMA akan digunakan pada model Jaringan Syaraf Tiruan arsitektur *Multi Layer Perceptron* algoritma Backpropagation. ARIMA memodelkan sifat linier data dan JST memodelkan sifat non-linier data. Data histori yang digunakan dalam sistem adalah data kurs Rupiah terhadap Dolar Amerika dalam rentang waktu Januari 2010 sampai Juni 2015. Setelah dilakukan prediksi, didapatkan hasil pengukuran kinerjanya, RMSE 60.43385, MAE 44.40632, dan MAPE 0.362984936 %. Sehingga dapat disimpulkan bahwa metode Jaringan Syaraf Tiruan ( $p,d,q$ ) bisa diterapkan dalam sistem peramalan kurs Rupiah terhadap Dolar Amerika, namun tidak signifikan lebih baik dibanding ARIMA atau JST saja.

**Kata kunci :** Kurs, ARIMA, JST, Jaringan Syaraf Tiruan ( $p,d,q$ ).

---

## Abstract

The exchange rate is the rate at which one country's currency can be exchanged for other currencies, play an important role in controlling the dynamics of the exchange market. Therefore, forecasting the exchange rate becomes interesting as a policy determination. The combined models ARIMA and ANN called Artificial Neural Network model ( $p,d,q$ ). Results error and differencing obtained from ARIMA models will be used in the model Neural Network architecture Multi Layer Perceptron Backpropagation algorithm. ARIMA modeling linear nature of the data and the ANN model the non-linear nature of the data. Historical data used in the system is the data of Rupiah exchange rate against the US dollar within the period January 2010 to June 2015. After the prediction, the result of measurement performance, RMSE 60.43385, 44.40632 MAE and MAPE 0.362984936 %. It can be concluded that the method of Artificial Neural Networks ( $p,d,q$ ) can be applied in the forecasting system Rupiahs exchange rate against the US Dollar, but not significant better than ARIMA or ANN only.

**Keywords:** Exchange rate, ARIMA, ANN, Artificial Neural Network ( $p,d,q$ ).

---

## 1. Pendahuluan

Pasar valuta asing telah mengalami perkembangan yang tak terduga selama beberapa dekade terakhir, dunia bergerak ke konsep "desa global" dan telah menjadi salah satu pasar keuangan terkemuka di dunia. Nilai tukar, yang merupakan tingkat dimana mata uang satu negara dapat ditukar dengan mata uang negara lain, memainkan peran penting dalam mengendalikan dinamika pasar valuta. Oleh karena itu, prediksi yang tepat dari nilai tukar merupakan faktor penting untuk mencapai sukses dalam banyak bisnis, investasi, dll di seluruh dunia. Saat ini terdapat bukti yang signifikan bahwa nilai tukar yang diprediksi sampai batas tertentu. Sebagian besar penelitian yang berkaitan dengan keuangan, prediksi pasar di era ini menunjukkan bahwa nilai tukar asing diprediksi dengan ketepatan tinggi [17].

Kesulitan dalam memprediksi nilai tukar telah menjadi masalah lama di bidang keuangan

internasional sebagai metode ekonometrik standar tidak dapat menghasilkan perkiraan signifikan lebih baik. Dari berbagai metode tukar peramalan, Model ARIMA telah menerima banyak perhatian dalam literatur dan cenderung menghasilkan sampel perkiraan terbaik. Literatur tentang model ARIMA cukup luas. Khususnya di daerah peramalan nilai tukar, [9][13][15][16]. Studi tambahan menggunakan teknik *time series* dengan berbagai macam data [10][11] dan beberapa model ARIMA yang di *hybrid* dengan metode lain [13][16]. Pembaca yang tertarik juga dapat merujuk pada buku yang ditulis oleh Chris Brooks [1].

Dalam beberapa dekade terakhir, dengan majunya teknologi komputer dan semakin populernya kecerdasan buatan, peneliti dan praktisi telah menjadi lebih cenderung untuk mengadopsi Jaringan Saraf Tiruan (JST) sebagai metode alternatif dalam peramalan keuangan. Meskipun sebagian besar model

JST berbagi tujuan bersama melakukan pemetaan fungsional, arsitektur jaringan yang berbeda sangat bervariasi dalam kemampuan mereka untuk menangani berbagai jenis masalah [17]. Literatur tentang Jaringan Syaraf Tiruan (JST) sebagai metode yang digunakan untuk memprediksi data *time series* [8], khususnya nilai tukar, [6][7][14][17].

Dalam jurnal [12] *An Artificial Neural Network (p,d,q) model for time series forecasting*, dikembangkan sebuah metode yaitu Jaringan Syaraf Tiruan (p,d,q). Metode tersebut, dikembangkan oleh Mehdi Khasei dan Mehdi Bijari pada tahun 2010. Pada penelitian tugas akhir ini dilakukan peramalan nilai tukar rupiah (IDR) terhadap dolar (USD) menggunakan model *Artificial Neural Network (p,d,q)* atau Jaringan Syaraf Tiruan (p,d,q). Dengan menggunakan model yang dikembangkan ini, maka penelitian ini dapat diterapkan untuk peramalan kurs untuk menghasilkan prediksi nilai tukar Rupiah terhadap Dolar Amerika yang lebih akurat.

## 2. Tinjauan Pustaka

### ARIMA

Metode yang dikembangkan oleh ilmuwan Inggris George Box dan Gwilym Jenkins (1976) ini memanfaatkan sepenuhnya data masa lalu dan data sekarang untuk menghasilkan peramalan jangka pendek yang akurat [1]. ARIMA yang secara penuh mengabaikan variabel bebas (*independent variable*) dalam membuat peramalan, juga merupakan salah satu metode peramalan yang baik sebagai metode peramalan model linier. ARIMA menggunakan data masa lalu dan sekarang sebagai variabel terikat (*dependent variable*). Oleh karena itu, ARIMA cocok jika observasi dari runtun waktu secara statistik berhubungan satu sama lain (*dependent*). Syarat utama menggunakan ARIMA sebagai model peramalan adalah, data harus bersifat stasioner, karena ARIMA merupakan peramalan yang baik untuk model linier [2]. Ada beberapa tahapan untuk metode peramalan ARIMA, yaitu [2]:



**Gambar 2.1** Tahapan ARIMA

- Tahap Identifikasi model, menggunakan grafik, statistik, dan alat lainnya untuk mengenali suatu pola dan komponen model.
- Tahap penaksiran dan pengujian model, menentukan koefisien atau parameter dengan menggunakan salah satu metode penentuan parameter model, lalu setiap model akan melalui uji diagnostik untuk memperoleh model terbaik.

- Tahap penerapan model, setelah didapatkan parameter dan model lolos uji diagnostik, maka model dapat digunakan dalam proses peramalan.

Dalam memilih model, kita akan mencoba untuk mematuhi prinsip parsimoni Prinsip parsimoni adalah, model yang digunakan harus memerlukan jumlah terkecil dari parameter yang akan memadai mewakili *time series*. Albert Einstein dikutip dalam Parzen (1982, hal. 68) sebagai berkomentar bahwa "segala sesuatu harus dibuat sesederhana mungkin tapi tidak sederhana." [2].

### Autoregressive (AR)

Bentuk umum model *autoregressive* dengan orde  $p$  (AR(p)) atau model ARIMA (p,0,0) dinyatakan sebagai berikut [2]:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-3} + \dots + \beta_p Y_{t-p} + e_t \quad (2.1)$$

dimana,

- $\beta_q$  = parameter *autoregressive* ke- $p$ .
- $\beta_0$  = konstanta AR.
- $Y_{t-p}$  = data pada saat  $t-p$ .
- $e_t$  = *error* prediksi AR pada saat  $t$ .

### Moving Average (MA)

Bentuk umum model MA dengan orde  $q$  (MA(q)) atau model ARIMA (0,0,q) dinyatakan sebagai berikut [2]:

$$Y_t = \theta_0 - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} + E_t \quad (2.2)$$

dimana,

- $\theta_q$  = parameter *moving average* ke- $q$ .
- $\theta_0$  = konstanta.
- $e_{t-q}$  = nilai *error* prediksi MA pada saat  $t-q$ .
- $E_t$  = *error* prediksi MA pada saat  $t$ .

### ARMA dan ARIMA Box-Jenkins

Model campuran ARMA atau ARIMA tanpa proses *differencing* atau jika data sudah stasioner di tuliskan sebagai berikut:

$$Y_t = \mu + \beta_1 Y_{t-1} + \dots + \beta_p Y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} + E_t \quad (2.3)$$

Jika data belum stasioner, maka model ARMA perlu ditambah proses *differencing* untuk data aslinya, dan untuk *error*-nya tidak perlu, menjadi model ARIMA (p,d,q) dengan notasi:

$$\Delta Y_t = \mu + \beta_1 \Delta Y_{t-1} + \dots + \beta_p \Delta Y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} + E_t \quad (2.4)$$

$$(Y_t - Y_{t-1}) = \mu + \beta_1 (Y_{t-1} - Y_{t-2}) + \dots + \beta_p \Delta Y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} + E_t \quad (2.5)$$

$$Y_t = \mu + (1 - \beta_1) Y_{t-1} + \dots + (\beta_{p-1} - \beta_p) \Delta Y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} + E_t \quad (2.6)$$

dimana,

- $\beta_p$  = parameter *autoregressive* ke- $q$ .
- $\mu$  = konstanta ARMA.
- $Y_{t-p}$  = data pada saat  $t-p$ .
- $e_t$  = *error* prediksi ARMA pada saat  $t$ .

- $\theta_q$  = parameter *moving average* ke- $q$ .
- $e_{t-q}$  = nilai *error* prediksi ARMA pada saat  $t-q$ .
- $\Delta Y_t$  = proses *differencing* (data ke- $t$  dikurang data ke-  $t-1$ )
- $E_t$  = *error* prediksi ARMA pada saat  $t$ .

### Jaringan Syaraf Tiruan (JST)

Jaringan Syaraf Tiruan (*Artificial Neural Network*) atau disingkat JST, adalah suatu arsitektur jaringan untuk memodelkan cara kerja sistem syaraf manusia (otak) dalam melaksanakan tugas tertentu [5]. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi pendekatan non-linier, klasifikasi data cluster dan regresi non-parametrik atau sebuah simulasi dari koleksi model syaraf biologi. Model syaraf ditunjukkan dengan kemampuannya dalam emulasi, analisis, prediksi, dan asosiasi.

Ada 4 elemen penting dalam arsitektur Jaringan Syaraf Tiruan (JST) [17]:

1. Banyaknya lapisan (*layers*)
2. Banyaknya *neuron* disetiap lapisan
3. Fungsi aktivasi disetiap lapisan tersembunyi (*hidden layer*) dan lapisan keluaran (*output layer*)

Algoritma pembelajaran (untuk pencarian nilai bobot dan bias akhir).

Tidak ada metode khusus untuk menentukan arsitektur terbaik suatu jaringan [13]. Untuk menentukan banyaknya *neuron* pada *hidden layer* digunakan beberapa cara yaitu melalui skenario dan bisa juga dengan persamaan [3]:

$$N_{hid} = \sqrt{N_{in} * N_{out}} \quad (2.7)$$

dimana  $N_{hid}$  adalah banyak *neuron hidden*,  $N_{in}$  adalah banyak *neuron input*, dan  $N_{out}$  adalah banyaknya *neuron output*.

- Fungsi Aktivasi

Salah satu fungsi aktivasi yang umum digunakan adalah fungsi aktivasi *logistic* atau *sigmoid*, dengan rumus [3]:

$$Sig(v) = \frac{1}{1 + \exp(-v)} \quad (2.8)$$

- Algoritma Backpropagation

Algoritma pelatihan jaringan syaraf perambatan galat mundur terdiri atas dua tahap perhitungan, yaitu: perhitungan maju untuk menghitung galat (*error*) antara keluaran actual dengan target; dan perhitungan mundur yang mempropagasikan balik galat tersebut untuk memperbaiki bobot-bobot sinaptik pada semua *neuron* yang ada.

### Jaringan Syaraf Tiruan ( $p, d, q$ )

Model gabungan pada penelitian ini, dari Jaringan Syaraf Tiruan yang memodelkan bagian non-

linier, bertujuan untuk menghasilkan hasil yang lebih akurat dengan menambahkan proses ARIMA untuk memodelkan bagian liniernya. Dalam model gabungan ini, runtun waktu dianggap sebagai fungsi non-linier dari beberapa pengamatan sebelumnya dan *random error*, sebagai berikut [12]:

$$y_t = f[(z_{t-1}, z_{t-2}, \dots, z_{t-m}), (e_{t-1}, e_{t-2}, \dots, e_{t-n})] \quad (2.9)$$

dimana,

- $f$  = fungsi determinan non-linier dari JST
- $z_t$  = data *differencing* asli
- $e_t$  = *error* hasil peramalan ARIMA pada waktu ke-  $t$
- $m$  dan  $n$  = bilangan integer

Pada tahap awal, model ARIMA digunakan untuk mendapatkan nilai *error*. Tahap selanjutnya, JST digunakan untuk menyelesaikan permasalahan non-linier dan hubungan linier yang ada di *error* dan data aslinya. Dengan demikian didapatkan persamaan [12],

$$z_t = w_0 + \sum_{j=1}^Q w_j \cdot g(w_{0j} + \sum_{i=1}^p w_{ij} \cdot z_{t-i} + \sum_{i=p+1}^{p+q} w_{ij} \cdot e_{t+p-i}) + \varepsilon_t \quad (2.10)$$

dimana,

$w_{ij} (i = 0, 1, 2, \dots, p + q, j = 0, 1, 2, \dots, Q)$  = bobot perhitungan

$w_j (j = 0, 1, 2, \dots, Q)$  = bobot *output*

$p, q, Q$  = bilangan integer, yang ditentukan dalam proses JST akhir

Sebagai catatan bahwa setiap set variabel  $\{e_i (i = t - 1, \dots, t - n)\}$  atau  $\{z_t (i = t - 1, \dots, t - n)\}$  bisa saja dihapus dalam proses pembentukan JST akhir. Hal ini mungkin terkait dengan dasar proses yang menghasilkan linier dan non-linier dalam data [12]. Sehingga, masukan system bisa saja *error* semua, kombinasi antara *error* dan *differencing*, bisa juga data *differencing* semua. Dengan data asli atau data kurs waktu ke- $t$  ( $Y_t$ ). Setelah mendapatkan hasil prediksi  $\hat{z}_t$  dari system JST, selanjutnya hasil prediksi  $\hat{z}_t$  tersebut digunakan untuk mencari data kurs prediksi ( $\hat{Y}_t$ ) dengan persamaan [12]:

$$\hat{Y}_t = \hat{z}_t + Y_{t-1} \quad (2.11)$$

dimana,

$\hat{Y}_t$  = Hasil prediksi

$\hat{z}_t$  = Hasil prediksi data *differencing*

$Y_{t-1}$  = Data asli pada saat-  $t-1$  (hari sebelumnya).

Model gabungan ini memiliki sistem kerja Jaringan Syaraf Tiruan dengan algoritma pembelajaran bobot menggunakan *backpropagation* (propagasi balik). Sistem tersebut memiliki masukan (*input*) berupa data asli hasil proses *differencing* ( $z_t$ ) dan data *error* ( $e_t$ ) hasil prediksi dari sistem ARIMA ( $p, d, q$ ). Model gabungan (*hybrid*) ini menghasilkan prediksi data *differencing* ( $\hat{z}_t$ ), kemudian hasil prediksi ( $\hat{z}_t$ ) dikembalikan menjadi data asli hasil

prediksi model gabungan ( $\hat{Y}_t$ ) [12]. Dengan memanfaatkan orde ARIMA sebagai banyaknya *input* pada sistem JST sesuai persamaan (2.40), maka jumlah *neuron input* ditentukan dengan menjumlahkan orde  $p$  dan orde  $q$  dari ARIMA

**Pengukuran Kinerja**

Sebagai alat ukur kinerja peramalan dalam penelitian ini, digunakan beberapa alat ukur, untuk menghitung *error*. Berikut berbagai alat ukur kinerja yang biasa digunakan :

$$MAE = \frac{\sum_{t=1}^n |Y_t - \hat{Y}_t|}{n} \tag{2.12}$$

dimana,

$Y_t$  = nilai aktual periode  $t$

$\hat{Y}_t$  = nilai ramal periode  $t$

$n$  = jumlah periode ramal

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (Y_t - \hat{Y}_t)^2}{n}} \tag{2.13}$$

dimana,

$Y_t$  = data aktual periode  $t$

$\hat{Y}_t$  = nilai hasil peramalan

$n$  = jumlah periode ramal

$$MAPE = \frac{\sum_{t=1}^n \frac{|Y_t - \hat{Y}_t|}{Y_t} \times 100}{n} \tag{2.14}$$

dimana,

$Y_t$  = nilai aktual periode  $t$

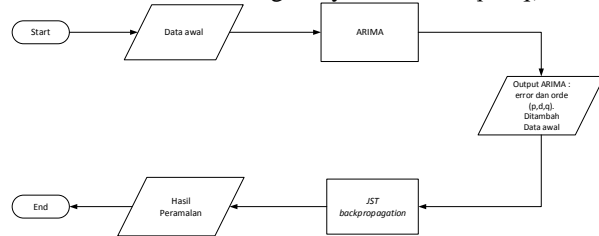
$\hat{Y}_t$  = nilai ramal periode  $t$

$n$  = jumlah periode ramal

**3. Perancangan Sistem**

**Deskripsi Sistem**

Pada penelitian ini akan dibangun sebuah sistem untuk melakukan peramalan nilai tukar Rupiah terhadap Dolar Amerika. Sistem ini menggunakan gabungan metode ARIMA ( $p,d,q$ ) dan Jaringan Syaraf Tiruan, sehingga membentuk sistem baru yang dinamakan metode Jaringan Syaraf Tiruan ( $p,d,q$ ).



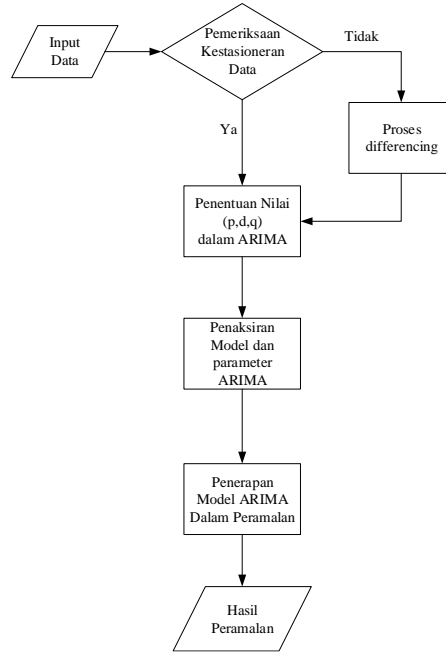
**Gambar 3.1** Gambaran Umum Sistem

Berdasarkan gambar 3.1, langkah proses secara singkat dapat dijelaskan sebagai berikut:

**Proses ARIMA**

Sistem ARIMA yang akan dibangun pada penelitian ini, terdapat 3 komponen utama, yaitu : *Autoregressive* (AR) dilambangkan dengan  $p$ ,

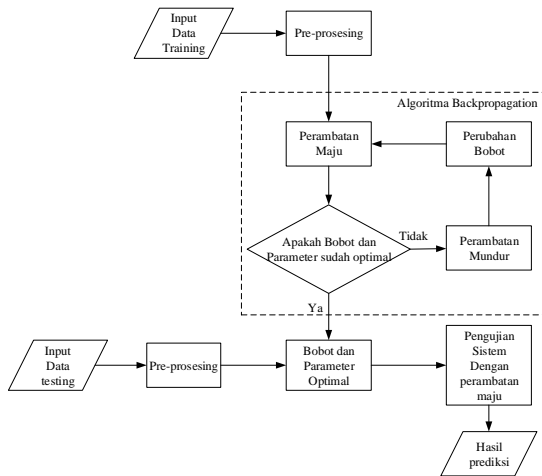
*integrated* (I) dilambangkan dengan  $d$ , dan *Moving Average* (MA) dilambangkan dengan  $q$ . Sehingga model ARIMA yang terbangun ditulis dengan ARIMA ( $p,d,q$ ). Model ARIMA yang digunakan dalam sistem adalah ARIMA Non-seasonal atau model ARIMA yang tidak dipengaruhi faktor musim dalam data. Skema proses perancangan sistem ini adalah sebagai berikut:



**Gambar 3.2** Proses ARIMA

**Proses Jaringan Syaraf Tiruan ( $p,d,q$ )**

Sistem Jaringan Syaraf Tiruan ( $p,d,q$ ) pada penelitian ini menggunakan arsitektur *Multi Layer Perceptron* (MLP) dengan satu *hidden layer*. Untuk proses pelatihan dan pembaharuan bobot, maka dipilih Algoritma *Backpropagation*. Terdapat dua input untuk arsitektur Jaringan Syaraf Tiruan ( $p,d,q$ ) ini, yaitu *error* yang didapat dari proses peramalan ARIMA dan *differencing* data asli pada waktu sebelumnya. Selanjutnya, akan diproses melalui 2 tahap utama dalam JST yaitu proses pelatihan (*training*) untuk mendapatkan bobot optimal menggunakan data *training*, yang digunakan pada tahap kedua yaitu tahap pengujian (*testing*) menggunakan data *testing*.



**Gambar 3.3** Skema Proses JST backpro

#### 4. Hasil dan Analisis

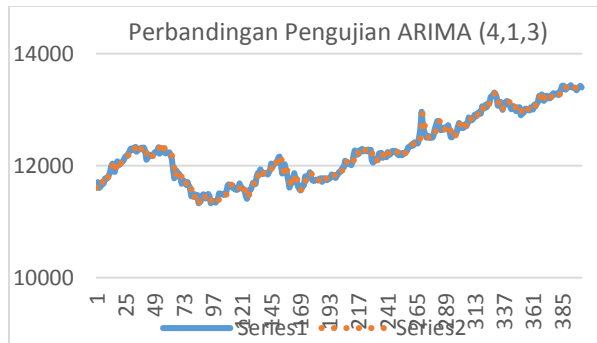
Data dibagi menjadi data *training* 70% dan data *testing* 30%, data *training* digunakan untuk memperoleh parameter-parameter pada proses ARIMA dan bobot optimal juga arsitektur pada proses JST dan Jaringan Syaraf Tiruan ( $p,d,q$ ). Setelah didapatkan parameter atau bobot dari data *training* lalu digunakan untuk data *training* digunakan untuk data validasi.

#### Analisis Hasil dan Pengujian Sistem ARIMA

Pada tahap identifikasi didapat 4 *lag cut-off* pada ACF dan 4 *lag cut-off* pada PACF. Setelah itu dilakukan pengujian parameter pada setiap kemungkinan model, sehingga didapatkan model ARIMA (4,1,3) tanpa konstanta sebagai model terbaik yang lolos uji dan layak untuk digunakan untuk peramalan. Dengan persamaan :

$$Y_t = Y_{t-1} + 1.010(Y_{t-1} - Y_{t-2}) + (-0.752)(Y_{t-2} - Y_{t-3}) + 0.529(Y_{t-3} - Y_{t-4}) + 0.136(Y_{t-4} - Y_{t-5}) - 1.118e_{t-1} - (-0.836)e_{t-2} - 0.598e_{t-3} + \varepsilon_t \quad (4.1)$$

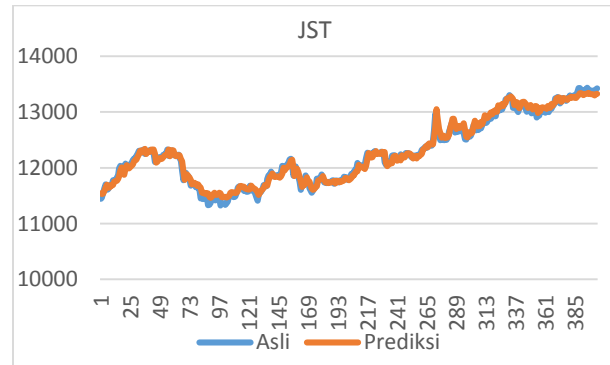
Dari model ARIMA tersebut didapatkan hasil pengukuran kinerja RMSE 38.77, MAE 22.244, dan MAPE 0.235%.



**Gambar 4.1** Grafik Hasil Prediksi Model ARIMA

#### Analisis Hasil dan Pengujian Sistem JST

Proses pencarian arsitektur JST terbaik dengan menggunakan skenario yang ditentukan untuk data *training*, lalu didapat arsitektur terbaik dengan 2 *input neuron* 20 *hidden neuron* 1 *output neuron* dengan *learning rate* 0.5 dan 5000 epoch. Dari arsitektur tersebut kemudian digunakan untuk prediksi dengan pengukuran kinerja RMSE 73.533, MAE 57.193, dan MAPE 0.4675%.



**Gambar 4.2** Grafik Hasil Prediksi JST

#### Analisis Hasil dan Pengujian Sistem Jaringan Syaraf Tiruan ( $p,d,q$ )

Proses pencarian arsitektur Jaringan Syaraf Tiruan ( $p,d,q$ ) terbaik dengan kombinasi *input* antara *error (et)* dan *differencing (Zt)* terbaik menggunakan skenario yang telah ditentukan untuk data *training*. Pada penelitian ini, dilakukan observasi dengan menggunakan jumlah *neuron input* dari orde ARIMA  $p$  dan  $q$  dan jumlah *neuron hidden* dengan persamaan (2.7), didapatkan arsitektur terbaik yaitu 7 *neuron input* dengan kombinasi 4 ( $Z_t$ ) dan 3 ( $e_t$ ), 3 *neuron hidden*, 1 *neuron output*, *learning rate* 0.5 dan 5000 epoch dengan pengukuran kinerja RMSE 60.433, MAE 44.406, dan MAPE 0.3629%.

#### Perbandingan Pengukuran Kinerja Sistem

1. Pengukuran kinerja prediksi data historis kurs Rupiah terhadap Dolar Amerika 4 Januari 2010 sampai 1 Nopember 2014.

**Tabel Error!** No text of specified style in document..1 Perbandingan *error* data histori

Training	RMSE	MAE	MAPE
ARIMA	38.11093181	22.22019651	0.234737485 %
JST	38.48910445	23.05229888	0.243946877 %
JST ( $p,d,q$ )	38.43295272	22.61929347	0.23934537 %

2. Pengukuran kinerja prediksi data pengujian kurs Rupiah terhadap Dolar Amerika 4 Nopember 2014 sampai 30 Juni 2015.

**Tabel Error! No text of specified style in document..2** Perbandingan *error* data pengujian

Testing	RMSE	MAE	MAPE
ARIMA	61.31831641	44.58160189	0.36462813 %
JST	73.53341942	57.19354839	0.467518837 %
JST ( <i>p,d,q</i> )	60.43385107	44.40632223	0.362984936 %

3. Pengukuran kinerja prediksi data validasi kurs Rupiah terhadap Dolar Amerika Juli 2015 sampai Agustus 2015.

**Tabel Error! No text of specified style in document..3** Perbandingan *error* data validasi

Validasi	RMSE	MAE	MAPE
ARIMA	23.73700156	17.70818402	0.131609027 %
JST	23.51736911	18.40570232	0.136759721 %
JST ( <i>p,d,q</i> )	22.50169553	16.1267404	0.119919404 %

## 5. Kesimpulan dan Saran

Model *hybrid* ARIMA dan JST yaitu Jaringan Syaraf Tiruan (*p,d,q*) sedikit lebih baik dibandingkan dengan ARIMA saja atau JST saja, namun tidak signifikan lebih baik dan harus dibuktikan melalui pengujian statistik untuk data kurs Rupiah terhadap Dolar Amerika. Dapat dilihat dari perbedaan yang kecil *error* yang dihasilkan dari ketiga model tersebut. Untuk bisa lebih meningkatkan akurasi prediksi, dapat digunakan metode optimasi arsitektur JST dengan menggunakan metode seperti GA, Prunning, , regresi dan lain-lain.

## 6. Daftar Pustaka

[1] Chris, Brooks. (2008). Introductory Econometrics for Finance. Second Edition. New York : Cambridge University Press

[2] Cyer, Jonathan D. dan Chan, Kung-Sik. (2008). Time Series Analysis With Applications in R. Second Edition. New York : Springer Science+Bussines Media, LLC.

[3] Suyanto, ST, MSc. (2009). Soft Computing. Bandung: Informatika.

[4] Suyanto, ST, MSc. (2011). Artificial Intelligence. Bandung: Informatika.

[5] Hermawan, Arief. 2006. Jaringan Saraf Tiruan. Yogyakarta: ANDI.

[6] Gioqinang Zhang and Michael Y. Hu, "Neural Network Forecasting of the British Pound/US Dollar Exchange Rate", Omega, Int. J. Mgmt Sci., Vol. 26, No. 4, pp. 495-506, 1998.

[7] Chakradhara Panda and V. Narasimhan, "Forecasting Exchange Rate Better with Artificial Neural Network", Journal of Policy Modeling, No. 29, pp. 227-236, 2007.

[8] G. Peter Zhang and Min Qi, "Neural Network Forecasting for Seasonal and Trend Time Series", European Journal of Operational Research 160. 501-514. 2005.

[9] Abul Kalam Azad and Md. Mahsin, "Forecasting Exchange Rates of Bangladesh using ANN and ARIMA Models: A Comparative Study", International Journal of Advanced Engineering Sciences and Technologies (IJAEST), ISSN:2230-7818, Vol. No. 10, Issue No. 1, pp. 31-36, 2011.

[10] Javier Contreas, Rosario Espinola, and Antonio J. Conejo, "ARIMA Models to Predict Next-Day Electricity Prices", IEEE TRANSACTION ON POWER SYSTEMS, VOL. 18, NO.3, August 2003.

[11] Ching-Wu Chu and Guoqiang Peter Zhang, "A comparative study of linear and nonlinear models for aggregate retail sales forecasting", Int. J. Production Economics 86, 217-231, 2003.

[12] Mehdi Khasei dan Mehdi Bijari, "An Artificial Neural Network (*p,d,q*) model for timeseries Forecasting", Expert Systems with Applications, 37, 479-489, 2010.

[13] Zhang, G. P, "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model", Neurocomputing, 50, 159-175, 2003.

[14] Adewole Adetunji Philip et. al. , "Artificial Neural Network Model for Forecasting Foreign Exchange Rate", World of Computer Science and Information Technology Journal (WCSIT), ISSN: 2221-0741, Vol. 1, No. 3, pp. 110-118, 2011.

[15] Joarder Kamruzzaman and Ruhul A. Sarker, "Comparing ANN Based Models with ARIMA for Prediction of Forex Rates", ASOR BULLETIN, Vol. 22, No. 2, pp. 2-11, 2003.

[16] Mehdi Khasei, Mehdi Bijari and Seyed Reza Hejazi, "Combining seasonal ARIMA models with computational intelligence techniques for time series forecasting", Springer-Verlag, ISBN: 978-1-61804-039-8, pp. 41-47, 2012.

[17] N.V.CHANDRASEKARA AND C.D.TILAKARATNE , "Forecasting Exchange Rates using Artificial Neural Networks", Research Gate, 2009.

[18] <http://www.bi.go.id/id/moneter/informasi-kurs/transaksi-bi/Default.aspx> , diakses pada tanggal 9 Oktober 2014 pukul 15.41 WIB dan tanggal 7 Agustus 2015 pukul 19.42 WIB.

[19] <http://panduanvalas.com/pengertian-kurs-valuta-asing-sistem-dan-jenisnya/198/> , diakses pada tanggal 29 Juli 2015, jam 10.30 WIB

[20]

<http://www.konsultanstatistik.com/2011/07/ipe-data.html> , diakses pada tanggal 29 Juli 2015 , jam 11.25 WIB

[21]

<http://www.rumusstatistik.com/2013/07/varian-dan-standar-deviasi-simpangan.html>, diakses pada tanggal 29 Juli 2015 , jam 11.43 WIB

[22] Makridakis, Spyros, Wheelwright, Steven C., McGee, Victor E. (1988). Metode dan Aplikasi Peramalan Jilid 1 Edisi Kedua, Jakarta: Penerbit Erlangga.

[23][ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/22.0/en/client/Manuals/IBM\\_SPSS\\_Forecasting.pdf](ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/22.0/en/client/Manuals/IBM_SPSS_Forecasting.pdf) , diakses pada tanggal 9 Agustus 2015, Pukul 11.43 WIB