

**PERAMALAN DATA PENUMPANG KERETA API
JANUARI 2013-NOVEMBER 2018 DENGAN MENGGUNAKAN
MAXIMAL OVERLAP DISCRETE WAVELET TRANSFORM-
RECURRENT NEURAL NETWORK (MODWT-RNN)**

Mira Andriyani, Subanar

Departemen Matematika, FMIPA Universitas Gadjah Mada

e-mail: miraandriyani1993@gmail.com

DOI: 10.14710/medstat.12.2.164-174

Article Info:

Received: 26 June 2019

Accepted: 21 December 2019

Available Online: 30 December
2019

Keywords:

*MODWT, RNN, Train
Passengers*

Abstract: The train is one of the public transportation that is very popular because it is affordable and free of congestion. There is often a buildup of train passengers at the station so that it sometimes causes an accumulation of passengers at the station and makes the situation at the station to be not conducive. In order to avoid a buildup of passengers, forecasting the number of passengers can be done. Forecasting is determined based on data in previous times. Data of train passengers in Java (excluding Jabodetabek) forms a non-stationary and contains nonlinear relationships between the lags. One of the nonlinear models that can be used is Recurrent Neural Network (RNN). Before RNN modeling, Maximal Overlap Wavelet Transform (MODWT) was used to make data more stationary. Forecasting model of train passengers in Java excluding Jabodetabek, Indonesia using MODWT-RNN results forecasting with RMSE is 252.85, while RMSE of SARIMA and RNN are 434.97 and 320.48. These results indicate that the MODWT-RNN model gives a more accurate result than SARIMA and RNN.

1. PENDAHULUAN

Alat transportasi merupakan salah satu kebutuhan penting manusia. Dari beberapa alat transportasi, kereta api merupakan salah satu angkutan umum yang sangat diminati masyarakat karena terjangkau dan bebas dari kemacetan. Tingginya animo masyarakat untuk menggunakan kereta api menimbulkan permasalahan baru. Terdapat dua permasalahan pokok yang muncul yaitu, kapasitas kereta dan frekuensi penyediaan layanan. Pada waktu tertentu seperti saat mendekati hari besar atau hari libur, sering terjadi penumpukan penumpang di stasiun. Penumpukan penumpang tersebut menyebabkan situasi di stasiun menjadi tidak kondusif, kotor, dan dimungkinkan terjadinya berbagai tindak kriminal. Pengaturan yang tepat tentang jadwal keberangkatan dan kedatangan kereta serta penyediaan banyak armada kereta api yang cukup dapat menjadi cara untuk

mengatasi penumpukan penumpang. Sebagai contoh jadwal yang tepat misalnya tidak ada kedatangan kereta dengan jumlah penumpang banyak ketika di stasiun ada banyak penumpang yang sedang menunggu keberangkatan. Penyediaan banyaknya kereta api juga harus tepat agar tidak merugikan penyedia layanan. Jadwal dan banyak kereta api dapat ditentukan dengan tepat salah satunya dengan memprediksi banyaknya penumpang di waktu mendatang menggunakan metode peramalan. Peramalan ditentukan berdasar data-data di waktu-waktu sebelumnya.

Data banyaknya penumpang kereta api di Jawa yang digunakan dalam makalah ini merupakan data runtun waktu (periode bulanan). Setelah dilakukan uji kestasioneran dan dibuat grafik datanya, data tersebut bersifat tidak stasioner dan berpola musiman. Pemodelan *Seasonal Autoregressive Integrated Moving Average* (SARIMA) merupakan salah satu metode yang dapat digunakan untuk memodelkan data yang bersifat tidak stasioner dan memuat pola musiman (Jeong, Koo, dan Hong, 2014). Pemodelan SARIMA tersebut yang sering digunakan untuk memodelkan peramalan dengan sifat-sifat tersebut seperti yang dilakukan Kafara, Rumlawang, dan Sinay (2017) dan referensi yang dapat digunakan antara lain Wei (2006). Namun, jika dilakukan pengecekan lebih lanjut terhadap sifat kelinieran data, metode SARIMA hanya tepat untuk data yang bersifat linier. Ketepatan tersebut dapat dilihat dari bentuk hubungan antar lag pada rumusan model SARIMA yang menunjukkan hubungan linier antar lag-nya. Oleh karena itu, jika hubungan antar lag bersifat nonlinier maka model SARIMA tidaklah tepat. Setelah dilakukan pengecekan terhadap data penumpang kereta api yang dibahas pada makalah ini, data bersifat nonlinier sehingga model SARIMA kurang tepat jika digunakan. Pemodelan yang hasilnya lebih tepat adalah pemodelan nonlinier.

Salah satu bentuk pemodelan nonlinier adalah dengan menggunakan *Neural Network* (NN). NN tidak memerlukan asumsi awal (Alexandridis dan Zaprans, 2013) dan dapat mendekati sebarang fungsi kontinu (Zhang, 2012). Pada NN, setiap lapisan terdiri dari satu atau lebih banyaknya neuron. Setiap neuron (kecuali neuron pada lapisan *input*) diaktivasi dengan suatu fungsi aktivasi yang biasanya merupakan fungsi nonlinier. Oleh karena fungsi aktivasi nonlinier tersebut, NN termasuk ke dalam pemodelan nonlinier. Banyak tipe NN berdasar arsitekturnya yang dikembangkan untuk masalah-masalah yang berbeda dan *Recurrent Neural Network* (RNN) salah satu diantaranya. RNN merupakan NN dengan arsitektur terdiri atas satu lapisan *input*, setidaknya satu lapisan tersembunyi, lapisan *recurrent*, dan satu lapisan *output*.

Salah satu bentuk RNN adalah RNN Elman (Elman, 1990). Pada RNN Elman, lapisan tambahan (*recurrent*) merupakan lapisan yang *input*-nya dari lapisan tersembunyi dan *output*-nya juga menuju lapisan tersembunyi. Lapisan *recurrent* menyimpan informasi dari iterasi sebelumnya untuk menjadi *input* di lapisan tersembunyi pada iterasi berikutnya. Artinya, iterasi pada RNN menyimpan informasi masa lalu.

Input pada NN memiliki peranan penting karena pola atau model yang terbentuk bergantung padanya. Pada *input* RNN untuk peramalan *time series* adalah lag-lag data. Penentuan banyak *input* dan lag-lag yang digunakan menjadi salah satu masalah dalam NN. Menurut Giovanis (2010), lag untuk *input* dapat dipilih berdasar metode ARIMA terbaik. ARIMA terbaik dapat ditentukan dengan nilai *Root Mean Squared Error* (RMSE).

Seperti teknik pemodelan pada umumnya, persiapan data pada NN juga memiliki peranan yang cukup penting terutama untuk data yang memuat pola tren dan musiman. Zhang dan Qi (2005) menyatakan bahwa *differencing* dan deseasonalisasi dapat memperbaiki hasil NN. Tidak hanya dengan *differencing* dan deseasonalisasi, dekomposisi

data sebelum proses NN juga dapat meningkatkan keakuratan hasil peramalan (Reis dan Silva (2005), Liang dan Liang (2006), Zhang, Gencay, dan Yazgan (2017)). Hasil penelitian tersebut menyebabkan munculnya *hybrid* NN dan salah satunya adalah *hybrid* antara transformasi wavelet dan NN. Salah satu tipe transformasi wavelet adalah *Discrete Wavelet Transform* (Daubechies, 1992). Huang dan Wang (2018) menggunakan *Discrete Wavelet Transform* (DWT) untuk mendekomposisi data menjadi beberapa subbarisan kemudian setiap subbarisan hasil dekomposisi tersebut dilakukan *Stochastic Recurrent Wavelet Neural Network* (SRWNN). Hasil penelitiannya menyatakan bahwa metode DWT-SRWNN mampu memberikan akurasi prediksi *global energy price* yang lebih baik dibanding metode *Backpropagation Neural Network* (BPNN), *Wavelet Neural Network* (WNN), SRWNN, dan *Long Short-Term Memory* (LSTM).

Discrete Wavelet Transform (DWT) terbatas hanya pada data dengan banyak 2^J , J bilangan bulat positif. Untuk mengatasi kelemahan DWT dapat digunakan *Maximal Overlap Discrete Wavelet Transform* (MODWT) (Zhu, Wang, dan Fan, 2014) dan terbukti bahwa MODWT memberikan hasil peramalan yang lebih baik. Oleh karena data yang dibahas banyaknya bukan kelipatan dua, berbeda dengan yang dilakukan Huang dan Wang (2018) yang menggunakan DWT, dalam makalah ini akan digunakan MODWT. MODWT digunakan untuk pemrosesan awal data dan selanjutnya hasil MODWT dimodelkan dengan menggunakan RNN sebagai model nonlinier. Penggabungan MODWT dan metode RNN pada makalah ini disebut MODWT-RNN. Pemodelan menggunakan SARIMA dan RNN juga akan dilakukan untuk mendukung pernyataan bahwa pemodelan MODWT-RNN lebih akurat dibandingkan kedua metode tersebut untuk data tidak stasioner dan nonlinier.

2. TINJAUAN PUSTAKA

2.1. *Maximal Overlap Discrete Wavelet Transform* (MODWT)

MODWT merupakan salah satu bentuk modifikasi dari DWT. Berbeda dengan DWT yang mensyaratkan data harus memenuhi kelipatan dua, MODWT dapat digunakan untuk sebarang ukuran sampel N . Oleh karena itu MODWT memiliki kelebihan yaitu mampu menghindari pengurangan data (*downsampling*). Dalam MODWT ada sebanyak N koefisien wavelet dan koefisien skala pada masing-masing levelnya (Percival dan Walden, 2000).

Filter wavelet MODWT $\{\tilde{h}_l\}$ dengan $\tilde{h}_l \equiv h_l/\sqrt{2}$ dan filter skala MODWT adalah $\{\tilde{g}_l\}$ dengan $\tilde{g}_l \equiv g_l/\sqrt{2}$. h_l dan g_l masing-masing adalah filter wavelet dan filter skala pada DWT. Filter wavelet MODWT harus memenuhi sifat berikut:

$$\sum_{l=0}^{L-1} \tilde{h}_l = 0, \sum_{l=0}^{L-1} \tilde{h}_l^2 = \frac{1}{2}, \text{ dan } \sum_{l=-\infty}^{\infty} \tilde{h}_l \tilde{h}_{l+2n} = 0 \quad (1)$$

Filter skala MODWT harus memenuhi sifat:

$$\sum_{l=0}^{L-1} \tilde{g}_l = 1, \sum_{l=0}^{L-1} \tilde{g}_l^2 = \frac{1}{2}, \text{ dan } \sum_{l=-\infty}^{\infty} \tilde{g}_l \tilde{g}_{l+2n} = 0 \quad (2)$$

untuk setiap bilangan bulat n berlaku

$$\sum_{l=-\infty}^{\infty} \tilde{g}_l \tilde{h}_{l+2n} = 0 \quad (3)$$

Didefinisikan matriks \mathbf{A}_j berukuran $N \times N$ berisi filter \tilde{g}_l dan matriks \mathbf{B}_j ukuran $N \times N$ berisi filter \tilde{h}_l . Matriks \mathbf{A}_j dan \mathbf{B}_j adalah matriks MODWT level j . Suatu filter $\{\tilde{h}_l: l = 0, \dots, L-1\}$ dengan lebar $2^{j-1}(L-1)+1$ mempunyai deret

$\tilde{h}_0, 0, \dots, 0, \tilde{h}_1, 0, \dots, \tilde{h}_{L-2}, 0, \dots, 0, \tilde{h}_{L-1}$ (nol diantara dua filter tak nol sebanyak $2^{j-1} - 1$) seperti pada Subanar dan Suhartono (2009). Tabel 1 menunjukkan banyak nol pada selang dua nilai filter tak nol.

Tabel 1 Penentuan Banyak Selang Nol

Level (j)	$2^{j-1} - 1$
1	0
2	2
3	3
...	...

Berdasar Tabel 1, untuk level dua ada sebuah nol di antara dua filter tak nol. Berpedoman pada Tabel 1 juga dapat diturunkan matriks A_j dan B_j untuk berbagai level j , ukuran N , dan ukuran L . Dengan j dan B_j , transformasi MODWT dari \tilde{V}_{j-1} ke \tilde{V}_j dan \tilde{W}_j dapat dinyatakan sebagai berikut:

$$\begin{bmatrix} \tilde{W}_j \\ \tilde{V}_j \end{bmatrix} = P_j \tilde{V}_{j-1} = \begin{bmatrix} B_j \\ A_j \end{bmatrix} \tilde{V}_{j-1} \quad (4)$$

dengan $P_j = \begin{bmatrix} B_j \\ A_j \end{bmatrix}$, \tilde{V}_{j-1} dapat diperoleh kembali dengan

$$\tilde{V}_{j-1} = P_j^T \begin{bmatrix} \tilde{W}_j \\ \tilde{V}_j \end{bmatrix} = \begin{bmatrix} B_j^T & A_j^T \end{bmatrix} \begin{bmatrix} \tilde{W}_j \\ \tilde{V}_j \end{bmatrix} = B_j^T \tilde{W}_j + A_j^T \tilde{V}_j \quad (5)$$

Selanjutnya $\tilde{V}_0 = X$, maka dengan mengaplikasikan tahap-tahap seperti di atas sampai level J , X dapat ditemukan kembali dari MODWT melalui:

$$\begin{aligned} X &= \tilde{V}_0 \\ &= B_1^T \tilde{W}_1 + A_1^T \tilde{V}_1 \\ &= B_1^T \tilde{W}_1 + A_1^T (B_2^T \tilde{W}_2 + A_2^T \tilde{V}_2) \\ &= B_1^T \tilde{W}_1 + A_1^T B_2^T \tilde{W}_2 + A_1^T A_2^T \tilde{V}_2 \\ &\vdots \\ &= B_1^T \tilde{W}_1 + \dots + A_1^T \dots A_{J-1}^T A_J^T \tilde{V}_J \end{aligned} \quad (6)$$

Dari persamaan tersebut, sinyal detail dan smooth pada MODWT level ke- J dapat dinyatakan sebagai:

$$D_j = A_1^T \dots A_{j-1}^T A_j^T \tilde{W}_j \quad (7)$$

$$S_j = A_1^T \dots A_{j-1}^T A_j^T \tilde{V}_j \quad (8)$$

sehingga dapat dinyatakan

$$X = \sum_{j=1}^J D_j + S_j \quad (9)$$

D_j disebut wavelet detail dan S_j disebut wavelet *smooth* MODWT.

2.2. Recurrent Neural Network (RNN) Elman

RNN merupakan NN dengan *feedback*. Gambar 1 menunjukkan arsitektur RNN Elman dengan k neuron (misalkan $x_{t-1}; x_{t-2}; \dots; x_{t-k}$) di lapisan *input*, p neuron di lapisan tersembunyi, dan satu neuron di lapisan *output*, dengan

$i = 1, 2, \dots, k$, i menunjukkan indeks neuron di lapisan input.

$j = 1, 2, \dots, p$, j menunjukkan indeks neuron di lapisan tersembunyi.

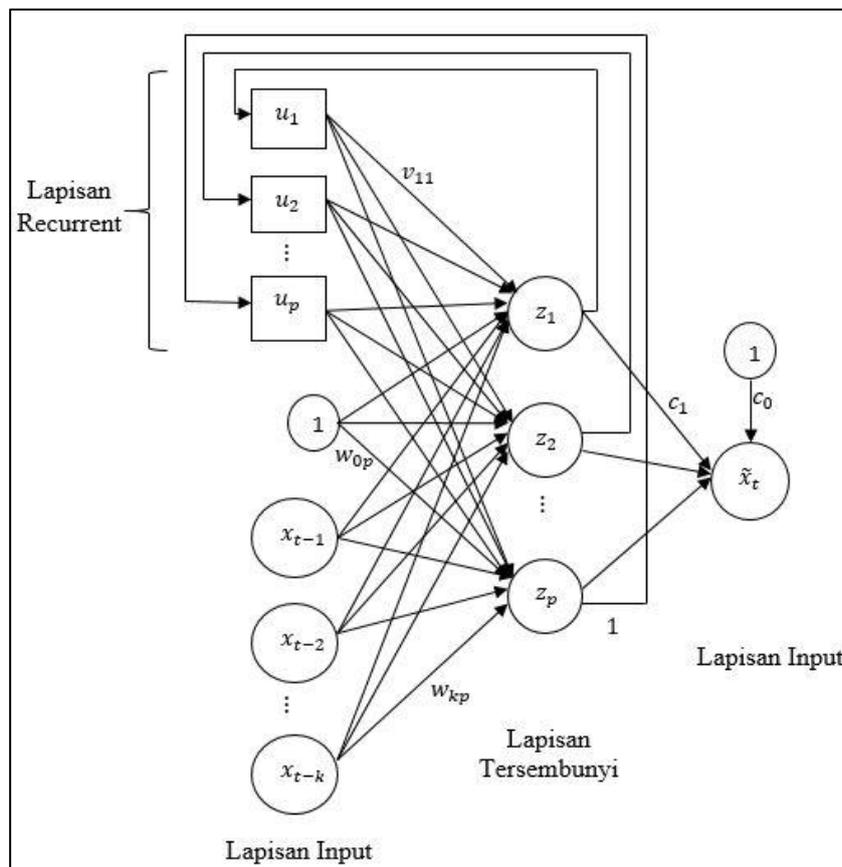
w_{0j} = bobot bias neuron ke- j pada lapisan tersembunyi.

w_{ij} = bobot yang menghubungkan neuron *input* ke- i (x_{t-i}) dengan neuron lapisan tersembunyi ke- j .

z_j = Hasil aktivasi dengan fungsi aktivasi f di lapisan tersembunyi.

$z_j = f(\text{net}_j)$, dengan f merupakan fungsi aktivasi.

net_j = neuron lapisan tersembunyi ke- j .



Gambar 1 Arsitektur RNN Elman

u_l = neuron lapisan recurrent ke- l .

$l = 1, 2, \dots, p$

v_{jl} = bobot yang menghubungkan neuron lapisan tersembunyi ke- j dengan neuron lapisan recurrent ke- l .

- c_0 = bobot bias neuron pada lapisan *output*.
 c_j = bobot yang menghubungkan neuron hasil aktivasi ke- j pada lapisan tersembunyi dengan neuron *output*.
 \tilde{x}_t = neuron *output* (hasil)

Berdasarkan arsitektur pada Gambar 1 dapat diperoleh:

$$net_j = w_{0j} + \sum_{i=1}^k w_{ij} x_{t-i} + \sum_{l=1}^p v_{jl} u_l \quad (10)$$

$$z_j = f(net_j) \quad (11)$$

$$\tilde{x}_t = c_0 + \sum_{j=1}^p c_j z_j \quad (12)$$

dengan f menyatakan fungsi aktivasi pada lapisan tersembunyi.

Estimasi bobot model RNN dilakukan dengan menggunakan metode *Gradient Descent* dengan momentum dan *adaptive learning rate*. Momentum digunakan untuk mengatasi perubahan yang drastis antara galat ke- i dengan galat ke- $i + 1$ dengan peredaman (Moller, 1997). *Adaptive learning rate* digunakan untuk meningkatkan kecepatan pembelajaran dan menjaga agar pembelajaran tetap stabil (Utomo, D., Pujiono, dan Soeleman, M. A., 2017). Tujuan akhir dari metode *Gradient Descent* dengan momentum dan *adaptive learning rate* adalah untuk meminimumkan E (seperti pada persamaan (13)) yaitu hingga nilainya kurang dari galat yang ditetapkan sebagai batas minimum.

$$E = \frac{1}{2N} \sum_{t=1}^N (\tilde{x}_t - x_t)^2 \quad (13)$$

Pembelajaran RNN dapat dilakukan dengan memperbarui (meng-*update*) bobot hubungan w_{ij} , v_{jl} , dan c_j . Berdasar aturan pembaharuan bobot, gradien dari bobot hubungan w_{ij} adalah

$$\Delta w_{ij}^{(k)} = m_0 \Delta w_{ij}^{(k-1)} + \eta m_0 \frac{\partial E}{\partial w_{ij}} \quad (14)$$

Sehingga persamaan yang digunakan untuk perbaikan bobot-bobot dalam RNN adalah:

$$\begin{aligned} w_{0j}^{(k+1)} &= w_{0j}^{(k)} + \Delta w_{0j}^{(k)} \\ &= w_{0j}^{(k)} + m_0 \Delta w_{0j}^{(k-1)} + \eta m_0 \frac{c_j}{N} \sum_{t=1}^N (\tilde{x}_t - x_t) f'(net_j) \end{aligned} \quad (15)$$

Perbaikan bobot dilakukan sampai galat mencapai nilai yang ditetapkan atau banyaknya iterasi telah mencapai batas maksimum yang ditentukan.

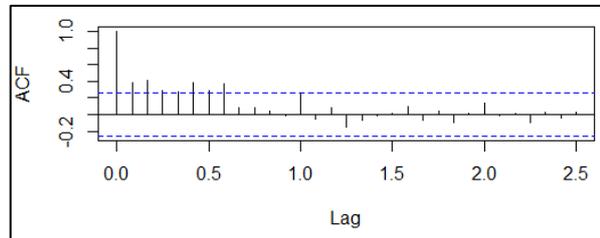
3. METODE PENELITIAN

3.1. Data

Data yang digunakan adalah data penumpang kereta api di Pulau Jawa (kecuali Jabodetabek) dari Januari 2013 hingga November 2018 yang diperoleh dari bps.go.id. Data tersebut merupakan data runtun waktu bulanan. Penentuan model ramalan menggunakan data banyak penumpang Januari 2013-September 2017 dan data Oktober 2017-November 2018 digunakan sebagai *testing* data.

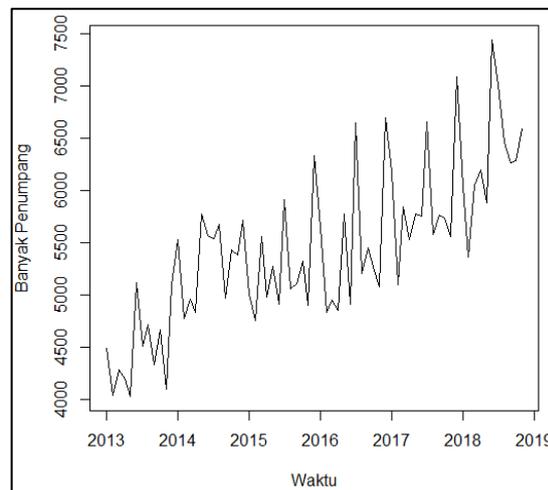
3.2. Langkah Analisis

Langkah awal yang dilakukan adalah menganalisis karakteristik data yang meliputi sifat kestasioneran, pola data, dan sifat nonlinier.



Gambar 2 Grafik ACF Data Penumpang Kereta di Pulau Api Jawa

Berdasarkan Gambar 2, data penumpang kereta api Jawa Januari 2013-September 2018 bersifat tidak stasioner dilihat dari plot ACF setelah lag 0 yang turun perlahan.



Gambar 3 Grafik Data Penumpang Kereta Api di Pulau Jawa

Terlihat dari Gambar 3 bahwa data bersifat musiman ditandai banyaknya penumpang di setiap Bulan Januari di atas sementara Februari selalu di bawah, begitu juga bulan lainnya memiliki pola yang hampir sama setiap tahunnya. Oleh karena itu, data termasuk memuat pola musiman.

Setelah mengetahui sifat kestasioneran dan pola data, langkah berikutnya adalah menentukan model ARIMA terbaik untuk mengetahui lag yang signifikan berpengaruh. Kemudian dilakukan Uji Terasvirta untuk mengetahui apakah data memuat hubungan nonlinier diantara lag-lag yang signifikan berpengaruh.

Tabel 2 menunjukkan bahwa lag 1 bersifat linier ($p - value > 0.05$) terhadap data ke- t sedangkan lag-12 dan lag-13 bersifat nonlinier.

Tabel 2 Hasil Uji Terasvirta Lag-1, Lag-12, dan Lag-13

Lag ke-	X-squared	df	p-value
1	3,3852	2	0.184
12	<i>Inf</i>	442	$< 2,2 \times 10^{-16}$
13	<i>Inf</i>	546	$< 2,2 \times 10^{-16}$

Setelah diketahui bahwa data bersifat tidak stasioner dan nonlinier, selanjutnya akan dilakukan pemodelan ramalan dengan MODWT-RNN. MODWT merupakan tahap awal untuk mengurangi ketidakstasioneran data sebelum dilakukan pemodelan RNN. MODWT-RNN merupakan RNN dengan neuron-neuron *input*-nya adalah sinyal hasil MODWT. Pada arsitektur RNN pada Gambar 1, neuron *input* adalah $x_{t-1}, x_{t-2}, \dots, x_{t-k}$, sedangkan pada MODWT-RNN masing-masing neuron *input* tersebut ditransformasi MODWT level 1 terlebih dahulu sehingga dihasilkan $D_1(t-1), S_1(t-1), \dots, D_1(t-k), S_1(t-k)$.

Hasil MODWT kemudian dijadikan *input* pada RNN. Sinyal-sinyal yang sejenis hasil MODWT (misal $D_1(t-1), D_1(t-2), \dots, D_1(t-k)$) dikelompokkan kemudian masing-masing kelompok sinyal dimodelkan dengan RNN. Pada kelompok sinyal S_1 juga dilakukan RNN sehingga diperoleh output \tilde{S}_1 . *Output* akhir peramalan dengan MODWT-RNN adalah penjumlahan dari \tilde{D}_1 dan \tilde{S}_1 . Tahapan yang dilakukan setelah diperoleh lag-lag yang akan digunakan sebagai *input* (misalkan diperoleh lag yang signifikan adalah lag-1, lag-2, dan lag-3) adalah sebagai berikut:

- MODWT untuk setiap lag yang signifikan dengan filter Haar level 1 sehingga diperoleh dua kelompok sinyal (sinyal detail (D_1) dan sinyal *smooth* (S_1)).
- Melakukan pemodelan RNN untuk setiap kelompok sinyal sehingga dihasilkan \tilde{D}_1 dan \tilde{S}_1 . RNN D_1 menggunakan $D_1(t-1), D_1(t-2), D_1(t-3)$ sebagai *input* sedangkan $D_1(t)$ sebagai target. Begitu juga dengan RNN S_1 menggunakan $S_1(t-1), S_1(t-2)$, dan $S_1(t-3)$ sebagai *input* sedangkan $S_1(t)$ sebagai target.

c. Hasil pemodelan MODWT-RNN adalah $\tilde{x}_t = \tilde{D}_1 + \tilde{S}_1$ dengan $\tilde{D}_1 = \tilde{c}_0^D + \sum_{j=1}^p \tilde{c}_j^D \tilde{z}_j^D$ dan $\tilde{S}_1 = \tilde{c}_0^S + \sum_{j=1}^p \tilde{c}_j^S \tilde{z}_j^S$ dengan $\tilde{c}_0^D, \tilde{c}_j^D, \tilde{z}_j^D, \tilde{c}_0^S, \tilde{c}_j^S, \tilde{z}_j^S$ adalah bobot-bobot yang diperoleh pada iterasi terakhir. Sedangkan $\tilde{z}_j^D = f(\tilde{n}et_j^D), \tilde{n}et_j^D = \tilde{w}_{0j}^D + \sum_{i=1}^3 \tilde{w}_{ij}^D D_1(t-i) + \sum_{l=1}^p \tilde{v}_{jl}^D u_l^D$, dan $f(\tilde{n}et_j^D) = \frac{1-e^{-2\tilde{n}et_j^D}}{1+e^{-2\tilde{n}et_j^D}}$. Sementara $\tilde{z}_j^S = f(\tilde{n}et_j^S), \tilde{n}et_j^S = \tilde{w}_{0j}^S + \sum_{i=1}^3 \tilde{w}_{ij}^S S_1(t-i) + \sum_{l=1}^p \tilde{v}_{jl}^S u_l^S$ dan $f(\tilde{n}et_j^S) = \frac{1-e^{-2\tilde{n}et_j^S}}{1+e^{-2\tilde{n}et_j^S}}$.

4. HASIL DAN PEMBAHASAN

4.1. Model SARIMA, RNN, dan MODWT-RNN Terbaik

Tujuan dari makalah ini adalah untuk menunjukkan bahwa peramalan dengan MODWT-RNN lebih akurat untuk data tidak stasioner dan nonlinier dibandingkan metode yang pernah ada (SARIMA dan RNN). Berdasarkan analisis plot ACF/PACF diperoleh beberapa model SARIMA yang mungkin dan dari beberapa model tersebut dipilih yang menghasilkan RMSE terkecil sebagai pemodelan SARIMA terbaik. ARIMA

(0,1,1)(1,0,0)[12] adalah pemodelan SARIMA terbaik untuk data banyak penumpang kereta, dan dari model tersebut diperoleh bahwa lag yang signifikan berpengaruh adalah lag-1, lag-12, dan lag-13. Hasil uji Terasvirta terhadap ketiga lag tersebut menunjukkan bahwa terhadap lag-1 data memiliki hubungan linier, sedangkan terhadap lag-12 dan lag-13 hubungannya nonlinier. Penentuan model RNN terbaik dilakukan dengan cara coba-coba dan dipilih arsitektur RNN yang menghasilkan RMSE terkecil. RNN (3,2,1) adalah arsitektur terbaik untuk data ini dan selanjutnya juga dilakukan pemodelan MODWT-RNN (3,2,1).

4.2. RMSE Model SARIMA, RNN, dan MODWT-RNN

Kriteria kebaikan model peramalan yang digunakan pada penelitian ini adalah nilai RMSE. Semakin kecil nilai RMSE suatu model, semakin akurat hasil peramalan. Tabel 2 menunjukkan perbandingan nilai RMSE ketiga model yang telah digunakan untuk memodelkan data banyak penumpang kereta api.

Tabel 3 Perbandingan RMSE Peramalan Penumpang Kereta Api

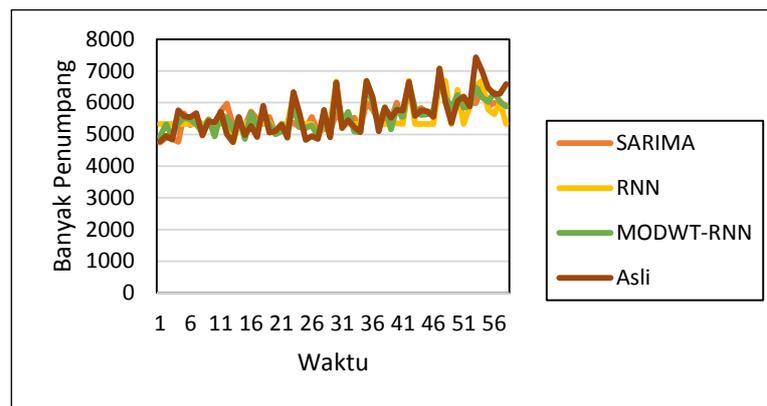
Model	RMSE	
	<i>Training</i>	<i>Testing</i>
SARIMA (0,1,1)(1,0,0)[12]	434,97	527,31
RNN (3,2,1)	320,48	602,05
MODWT-RNN (3,2,1)	252,85	454,70

Berdasarkan Tabel 2 diperoleh bahwa pemodelan MODWT-RNN merupakan pemodelan terbaik untuk data banyak penumpang kereta api karena nilai RMSE terkecil. Pemodelan nonlinier RNN memberikan hasil yang lebih baik dibanding model SARIMA pada data training. Hal ini menunjukkan bahwa untuk data yang memuat hubungan nonlinier (diantara lag-nya), pemodelan linier (SARIMA) memberikan hasil yang kurang akurat. Suatu pilihan yang tepat jika pemodelan nonlinier seperti RNN yang digunakan untuk memodelkan ramalan data yang memuat hubungan nonlinier.

Data penumpang kereta yang digunakan juga merupakan data yang bersifat tidak stasioner. Tujuan MODWT pada penelitian ini adalah untuk membuat data menjadi lebih stasioner sehingga data lebih mudah dianalisis. Berdasarkan hasil pemodelan data banyak penumpang kereta api tersebut terbukti bahwa ketika dilakukan MODWT sebelum dilakukan pemodelan nonlinier RNN hasilnya lebih baik dibandingkan pemodelan tanpa MODWT terlebih dulu. Hal ini menunjukkan bahwa untuk data yang bersifat stasioner, proses awal untuk membuat data lebih stasioner sebelum dilakukan pemodelan adalah pilihan terbaik agar hasil peramalan lebih akurat.

4.3. Perbandingan Hasil Peramalan Model SARIMA, RNN, dan MODWT-RNN

Hasil peramalan ketiga model yang digunakan juga terlihat dari Gambar 4. Berdasarkan Gambar 5, garis hijau paling dekat dengan garis hitam (data asli). Hasil ini menunjukkan bahwa peramalan banyak penumpang kereta api di Jawa dengan MODWT-RNN adalah pemodelan yang terbaik dibandingkan model SARIMA maupun RNN.



Gambar 5. Grafik Data Banyak Penumpang Kereta dan Hasil Peramalan Model ARIMA, RNN, dan MODWT-RNN

5. KESIMPULAN

Berdasarkan nilai RMSE, model MODWT-RNN merupakan model terbaik dalam meramalkan banyaknya penumpang kereta di Jawa (kecuali Jabodetabek) dibandingkan model ARIMA dan RNN. Peneliti lain dapat mengembangkan penelitian ini dengan menggunakan filter dan level lain. Selain itu, juga dapat dilakukan pemodelan MODWT-RNN dengan menggunakan Uji Wald atau Uji F untuk menentukan banyaknya neuron tersembunyi.

DAFTAR PUSTAKA

- Alexandridis, A. K. dan Zaprakis, A. D. 2013. Wavelet Neural Network: A practical Guide, *Neural Networks*. Vol. 42, hal. 1–27.
- Daubechies, I. 1992. *Ten Lectures of Wavelets*. New York: Springer Verlag.
- Elman, J. L. 1990. Finding Structure in Time. *Cognitive Science 14*, hal. 179–211.
- Giovanis, E. 2010. Application of Feed-Forward Neural Networks Autoregressive Models with Genetic Algorithm in Gross Domestic Product Prediction, *World Academy of Science, Engineering and Technology*. 64.
- Huang, L. dan Wang, J. 2018. Forecasting Energy Fluctuation Model by Wavelet Decomposition and Stochastic Recurrent Wavelet Neural Network, *Neurocomputing*. 309 (2018), hal. 70–82.
- Jeong, K., Koo, C., dan Hong, T. 2014. An Estimation Model for Determining the Annual Energy Cost Budget in Educational Facilities using SARIMA (Seasonal Autoregressive Integrated Moving Average) and ANN (Artificial Neural Network). *Energy*. Vol. 71, hal. 71–79. <https://doi.org/10.1016/j.energy.2014.04.027>
- Kafara, Z., Rumlawang, F. Y., dan Sinay, L. J. 2017. Peramalan Curah Hujan dengan Pendekatan Seasonal Autoregressive Integrated Moving Average (SARIMA) (Studi Kasus : Curah Hujan Bulanan Di Kota Ambon, Provinsi Maluku). *Jurnal Ilmu Matematika dan Terapan*, Vol. 11, No. 1, hal. 63–74.

- Liang, Y. dan Liang, X. 2006. Improving Signal Prediction Performance of Neural Networks Through Multiresolution Learning Approach. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 36, No. 2.
- Moller, M. 1997. Thesis: Efficient Training of Feed-Forward Neural Networks. Computer Science Department, Aarhus University.
- Percival, D. B. dan Walden, A. T. 2000. *Wavelet Methods for Time Series Analysis*. New York: Cambridge University Press.
- Reis, A. J. R. dan Silva, A. P. A. 2005. Feature Extraction via Multiresolution Analysis for Short-Term Load Forecasting. *IEEE Transactions on Power Systems*, Vol. 20, No. 1.
- Subanar dan Suhartono. 2009. *Wavelet Neural Networks untuk Peramalan Data Time Series Finansial*, Program Penelitian Ilmu Dasar Perguruan Tinggi, FMIPA UGM, Yogyakarta.
- Utomo, D., Pujiono, dan Soeleman, M. A. 2017. Stock Price Prediction Using Back Propagation Neural Network Based On Gradient Descent With Momentum and Adaptive Learning Rate. *Journal Of Internet Banking and Commerce*, vol 22, No. 3.
- Wei, W.S. 2006. *Time Series Analysis: Univariate and Multivariate Methods, 2nd edition*. Boston. Addison Wesley.
- Zhang, G. P. 2012. Neural Network for Time Series Forecasting. *Handbook of Neural Computing*. Springer, Germany, hal. 461–477.
- Zhang, G. P. dan Qi, M. 2005. Neural Network Forecasting for Seasonal and Trend Time Series, *European Journal of Operational Reserch*. 160, hal. 501–514.
- Zhang, K., Gencay, R., dan Yazgan, M. E. 2017. Application of Wavelet Decomposition in Time-Series Forecasting, *Economics Letters*. 158, hal. 41–46.
- Zhu, L., Wang, Y., dan Fan, Q. 2014. MODWT-ARMA Model for Time Series Prediction, *Applied Mathematical Modelling*, 38 (2014), hal. 1859–1865.