

PEMODELAN *FORECASTING CONTAINER THROUGHPUT* DENGAN METODE ARIMA-BOX JENKINS, JARINGAN SYARAF TIRUAN DAN HYBRIDA ARIMA-BOX JENKINS-JARINGAN SYARAF TIRUAN

Aris Gunaryati¹⁾, Arrahmah Aprilia²⁾

^{1), 2)} Teknik Informatika Universitas Nasional Jakarta
Jl. Sawo Manila No. 61, Pejaten, Jakarta Selatan
Email : arisgunaryati@yahoo.co.id¹⁾, ama.apriliah@gmail.com²⁾

Abstrak

Untuk memperkirakan *throughput container* di pelabuhan dengan akurat, diperlukan metode peramalan/*forecasting* yang sesuai dan cocok dalam pemodelan data *throughput container* pada suatu periode. Pemodelan yang sesuai dan cocok akan menghasilkan suatu nilai peramalan yang akurat dan akan membantu pihak manajer dan supervisor dalam mengambil keputusan terbaik untuk mengatasi masalah penumpukan *container*.

Dalam penelitian ini akan dikembangkan sebuah metode peramalan untuk data runtun waktu *throughput container* dengan metode ARIMA-Box Jenkins, Jaringan Syaraf Tiruan (Neural Network) dan metode hybrid ARIMA-Box Jenkins dan Jaringan Syaraf Tiruan. Nilai akurasi peramalan digunakan sebagai parameter penilaian model. Berdasarkan nilai rata-rata akurasi maka diperoleh model dengan metode ARIMA-Box Jenkins memiliki rata-rata akurasi 88,73%, model dengan metode Jaringan Syaraf Tiruan (Neural Network) memiliki rata-rata akurasi 57,47%, model hybrid ARIMA-Box Jenkins dan Jaringan Syaraf Tiruan dengan input adalah hasil dari metode ARIMA-Box Jenkins memiliki rata-rata akurasi 76,43%, dan yang terakhir, model hybrid ARIMA-Box Jenkins dan Jaringan Syaraf Tiruan dengan input adalah hasil dari metode Jaringan Syaraf Tiruan memiliki rata-rata akurasi 80,68%.

Dari hasil penelitian ini dapat disimpulkan bahwa model yang paling sesuai dan cocok untuk data runtun waktu *throughput container* adalah model ARIMA-Box Jenkins, yaitu ARIMA (0,0,18).

Kata kunci: *throughput container*, pemodelan, peramalan, ARIMA Box Jenkins, Jaringan Syaraf Tiruan, runtun waktu, hybrid ARIMA-JST

1. Pendahuluan

Salah satu issue penting dalam pengelolaan pelabuhan dan lapangan penumpukan kontainer adalah memperkirakan *throughput* kontainer dengan akurat [1] [2]. Pengukuran *throughput* adalah pengukuran jumlah pergerakan kontainer ketika melalui terminal atau lapangan mengalami penumpukan kontainer. Langkah ini

berguna dalam memberikan informasi bagi manajer dan supervisor tentang penumpukan kontainer yang dianggap produktif atau tidak.

Untuk memperkirakan *throughput container* dengan akurat, diperlukan metode peramalan/*forecasting* yang sesuai dan cocok dalam pemodelan data *throughput container* pada suatu periode. Pemodelan yang sesuai dan cocok akan menghasilkan suatu nilai peramalan yang akurat dan akan membantu pihak manajer dan supervisor dalam mengambil keputusan terbaik untuk mengatasi masalah penumpukan *container*.

Salah satu metode peramalan yang paling dikembangkan saat ini adalah runtun waktu (*time series*), yakni menggunakan pendekatan kuantitatif dengan data masa lampau dikumpulkan dan dijadikan acuan untuk peramalan masa depan. Teknik peramalan data runtun waktu terbagi menjadi dua bagian. Pertama, model peramalan yang didasarkan pada model matematika statistik seperti *moving average*, *exponential smoothing*, regresi (parametrik dan non parametrik), serta yang paling sering digunakan adalah ARIMA (Box Jenkins). Kedua, model peramalan yang didasarkan pada kecerdasan buatan seperti *neural network*, algoritma genetika, *simulated annealing*, *genetic programming*, klasifikasi dan hybrid.

Metode-metode tersebut memiliki kekurangan dan keunggulan yang berbeda. Terlebih lagi, masalah dalam dunia nyata seringkali merupakan masalah yang kompleks dan satu model mungkin tidak mampu mengatasi hal tersebut dengan baik (DT Wiyanti & R Pulungan, 2012). Untuk itu, beberapa tahun terakhir telah dilakukan penelitian untuk menggabungkan beberapa model menjadi satu agar menghasilkan ramalan dengan tingkat akurasi yang lebih baik secara rata-rata dibandingkan dengan model tunggal.

Oleh karena permasalahan yang ditemui di dunia nyata adalah permasalahan data yang tidak stasioner atau tidak linier, maka dibutuhkan suatu alat yang akurat dan efektif untuk meramalkan perilaku data yang tidak stasioner atau tidak linier tersebut. Namun terdapat permasalahan lain dalam model jaringan syaraf tiruan yang sering ditemui yaitu masalah *overfitting*, di mana model yang dibuat hanya menghasilkan output yang baik untuk data yang dilatih saja dan tidak untuk data yang divalidasi (data yang tidak masuk proses training). Ini

adalah fenomena *overfitting* yang termasuk masalah yang sering ditemui ketika menggunakan model jaringan syaraf tiruan. Di sisi lain, permasalahan tersebut dapat diminimalisir dengan pengembangan yaitu mengkombinasikannya dengan model lain. Kombinasi yang pernah dicobakan untuk peramalan runtun waktu adalah model jaringan syaraf tiruan dengan model ARIMA [6].

Ada tiga hal yang menjadi alasan penggunaan pengkombinasian model ARIMA dan neural network [7]. Pertama sering kali terjadi kesulitan untuk menerapkan penggunaan model linier atau model non linier pada suatu permasalahan runtun waktu, sehingga model kombinasi ini menjadi alternatif yang lebih mudah. Kedua, dalam kenyataannya, runtun waktu jarang yang linier atau non linier saja tetapi sering mengandung keduanya, di mana tidak hanya model ARIMA dan *neural network* masing-masing dapat memodelkan setiap kasusnya, sehingga pengkombinasian ini dapat digunakan untuk memodelkan runtun waktu yang mengandung linier dan non linier. Ketiga, dalam beberapa literature peramalan menyatakan bahwa tidak ada model tunggal yang terbaik pada setiap situasi.

Dalam penelitian ini akan dikembangkan sebuah metode peramalan untuk data runtun waktu *throughput container* dengan metode ARIMA-Box Jenkins, Jaringan Syaraf Tiruan (Neural Network) dan metode hybrid ARIMA-Box Jenkins dan Jaringan Syaraf Tiruan.

2. Pembahasan

2.1 Definisi Forecasting

Surjasa [3] menyatakan secara umum *forecasting* didefinisikan sebagai proses menganalisis data saat ini dan data pada masa lalu untuk menentukan tren di masa depan.

2.2 Metode ARIMA Box Jenkins

Model *Autoregresif Integrated Moving Average* (ARIMA) Box Jenkins adalah model yang secara penuh mengabaikan independen variabel dalam membuat peramalan dan hanya menggunakan nilai masa lalu dan sekarang dari variabel dependen untuk peramalan. ARIMA cocok jika observasi dari deret waktu (*time series*) secara statistik berhubungan satu sama lain (*dependent*). Model ARIMA terdiri dari tiga langkah dasar, yaitu tahap identifikasi, tahap penaksiran dan pengujian, dan pemeriksaan diagnostik. Selanjutnya model ARIMA dapat digunakan untuk melakukan peramalan jika model yang diperoleh memadai. Metode ARIMA-Box Jenkins ini sangat baik ketepatannya untuk peramalan jangka pendek, sedangkan untuk peramalan jangka panjang ketepatan peramalannya kurang baik. Biasanya akan cenderung *flat* (mendatar/konstan) untuk periode yang cukup panjang.

2.3 Bentuk Umum Model ARIMA Box Jenkins

Secara umum, bentuk matematis dari model ARIMA(p,d,q) dapat ditulis sebagai berikut [4] [5]:

$$(1 - \phi_1 B - \dots - \phi_p B^p)(1-B)^d Z_t = \theta_0 + (1 - \theta_1 B - \dots - \theta_q B^q) a_t \dots (1)$$

dengan B adalah operator mundur, yaitu $B^k Z_t = Z_{t-k}$

Penentuan orde p dan q dari model ARIMA pada suatu data runtun waktu dilakukan dengan mengidentifikasi plot Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF) dari data yang sudah stasioner.

Berikut ini adalah petunjuk umum untuk penentuan orde p dan q pada suatu data runtun waktu yang sudah stasioner.

Tabel 1. Petunjuk Penentuan Orde p dan q dalam Model ARIMA-Box Jenkins

Proses	ACF	PACF
AR (p)	<i>Dies down</i> (turun cepat secara eksponensial / sinusoidal)	<i>Cuts off after lag p</i> (terputus setelah lag p)
MA (q)	<i>Cuts off after lag q</i> (terputus setelah lag q)	<i>Dies down</i> (turun cepat secara eksponensial / sinusoidal)
ARMA (p,q)	<i>Dies down</i> (turun cepat secara eksponensial / sinusoidal)	<i>Dies down</i> (turun cepat secara eksponensial / sinusoidal)
AR (p) atau MA (q)	<i>Cuts off after lag q</i> (terputus setelah lag q)	<i>Cuts off after lag p</i> (terputus setelah lag p)
White noise	Tidak ada yang signifikan (tidak ada yang keluar batas)	Tidak ada yang signifikan (tidak ada yang keluar batas)

2.4 Model Peramalan Container Throughput Dengan Metode Arima-Box Jenkins

Pengolahan data dilakukan dengan menggunakan bantuan *Expert Modeller* yang tersedia pada perangkat lunak SPSS versi 21.0 Dengan menggunakan *Expert Modeller*, diperoleh hasil sebagai berikut :

Model Description										
Model ID	data throughput	Model_1	Model Type							
			ARIMA(0,0,18)							
Model Statistics										
Model	Number of Predictors	Model Fit statistics				Ljung-Box Q(18)			Number of Outliers	
		Stationary R-squared	R-squared	RMSE	MAPE	MAE	Statistics	DF		Sig.
data throughput Model_1	0	.089	.089	334.469	42.275	273.684	19.586	17	.296	0

Gambar 1. Hasil output pemodelan ARIMA dengan *Expert Modeller SPSS 21.0*

ARIMA Model Parameters				Estimate	SE	t	Sig.
data throughput-Model_1	data throughput	No Transformation	Constant	1.003E3	24.136	45.301	.000
		MA	Lag 18	.294	.113	2.609	.010

Forecast							
Model		111	112	113	114	115	116
data throughput-Model_1	Forecast	917	1108	1030	1053	1133	1314
	UCL	1577	1768	1690	1713	1793	1974
	LCL	257	448	370	393	473	654

For each model, forecasts start after the last non-missing in the range of the requested estimation period, and end at the last period for which non-missing values of all the predictors are available or at the end date of the requested forecast period, whichever is earlier.

Gambar 2. Hasil output parameter model ARIMA dan peramalan dengan model ARIMA

Tabel 2. Perbandingan Hasil Real dan Prediksi Throughput Kontainer untuk 6 minggu dengan model ARIMA

Minggu ke-	X-true	X-Pred	Residual	Accuracy
111	873	916.8772	-43.88	94.97%
112	1534	1108.176	425.82	72.24%
113	933	1029.673	-96.67	89.64%
114	1130	1053.182	76.82	93.20%
115	1089	1132.662	-43.66	95.99%
116	1156	1313.899	-157.90	86.34%
Rata-rata				88.73%

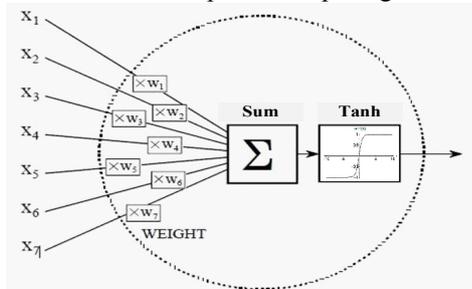
Pada tabel di atas kita dapat lihat rata-rata ke akuratan peramalan dengan metode Arima-Box Jenkins adalah 88,73%.

2.5 Jaringan Syaraf Tiruan Backpropagation

Rojas [4] menyatakan bahwa algoritma Backpropagation dapat dipecah menjadi empat langkah utama, yaitu:

i) Perhitungan Umpan-maju

Awal perhitungan dengan menginput nilai data mentah input. Setiap node dari lapisan input terhubung ke node dari lapisan tersembunyi dan setiap simpul dari lapisan tersembunyi terhubung ke simpul pada lapisan output. Ada satu bobot(w) yang terkait dengan setiap koneksi. Hal ini dapat dilihat pada gambar berikut:



Gambar 3. Model Neuron dan fungsi aktivasi

Input layer merupakan informasi mentah yang dimasukkan ke jaringan. Bagian dari jaringan ini tidak pernah berubah nilai-nilainya.

Hidden layer menerima data dari lapisan Input berupa nilai jumlah(SUM) dari perkalian tiap input training dengan bobot perkoneksinya dengan rumus:

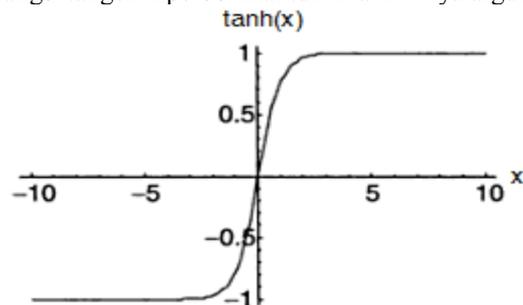
$$SUM = net_j = \sum_{i=1}^{n+1} w_{ij}^{(1)} \hat{o}_i \tag{1}$$

Dimana $w_{ij}^{(1)}$ adalah bobot koneksi dari input i ke node j, dan \hat{o}_i adalah input.

Setiap Input tunggal ke jaringan diduplikasi dan turunkan ke node pada lapisan tersembunyi dengan fungsi aktivasi. Karlik [5] menyatakan bahwa fungsi aktivasi hiperbolik tangent memiliki akurasi yang paling tinggi daripada fungsi aktivasi yang umumnya digunakan. Fungsi ini mudah didefinisikan sebagai rasio antara sinus hiperbolik dan fungsi cosinus atau diperluas sebagai rasio dari setengah perbedaan dan setengah jumlah dari dua fungsi eksponensial dalam poin net_j dan $-net_j$ sebagai berikut:

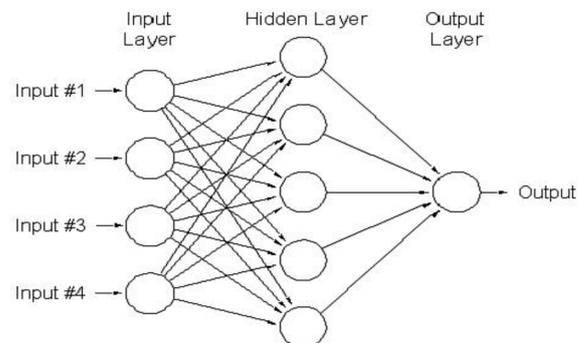
$$f(net_j) = \frac{\sin(net_j)}{\cos(net_j)} = \frac{e^{net_j} - e^{-net_j}}{e^{net_j} + e^{-net_j}} \tag{2}$$

Hiperbolik Tangent Fungsi ini mirip dengan fungsi sigmoid. Kisaran outputnya antara -1 dan 1 seperti yang terlihat dalam Gambar. Berikut ini adalah grafik dari fungsi tangen hiperbolik untuk nilai riil-nya argumen :



Gambar 4. Hyperbolic Tangent Function

Perhitungan fungsi aktivasi ini dilakukan sesuai dengan jumlah layer yang digunakan. Pada penelitian ini menggunakan 1 input layer, 1 hiden layer dan 1 output layer seperti yang digambarkan oleh arsitektur jaringan berikut:



Gambar 5. Arsitektur Jaringan Syaraf Tiruan Multi Layer.

ii) Propagasi balik ke lapisan output

Pada saat ini dilakukan proses perhitungan error output unit dengan rumus:

$$\delta_{pk}^o = (y_{pk} - o_{pk}) f_{pk}^{\prime}(net_{pk}^o) \quad (3)$$

Dimana δ_{pk}^o adalah nilai error output, y_{pk} adalah nilai target dari output layer, o_{pk} adalah output aktual dan $f_{pk}^{\prime}(net_{pk}^o)$ adalah nilai turunan fungsi aktivasi pada output layer.

iii) Propagasi balik ke lapisan tersembunyi

Tahap ini adalah proses menghitung error untuk tiap node pada hidden layer, dengan persamaan:

$$\delta_{pj}^h = f_{pj}^{\prime}(net_{pj}^h) \sum_k \delta_{pk}^o w_{kj}^o \quad (4)$$

Dimana δ_{pj}^h adalah nilai error ke hidden layer, $f_{pj}^{\prime}(net_{pj}^h)$ adalah turunan dari fungsi aktivasi pada node hidden layer, $\sum_k \delta_{pk}^o w_{kj}^o$ adalah jumlah delta inputnya tiap-tiap unit tersembunyi (dari unit-unit yang berada di lapisan atasnya).

Dengan catatan bahwa perhitungan error hidden layer ini dilakukan sebelum bobot ke output layer di perbarui.

iv) Update Bobot

Setelah error diketahui, selanjutnya dilakukan perhitungan propagasi balik dan penyesuaian bobot. Hal ini terdiri dari dua langkah proses. Kesalahan disebarkan dari output layer ke hidden layer. Di sinilah tingkat pembelajaran (*learning rate*) dan momentum dibawa ke persamaan. Proses pertama adalah menghitung perubahan bobot :

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta \delta_{pk}^o f_{pk}^{\prime}(net_{pk}^o) \quad (5)$$

Dimana $w_{kj}^o(t+1)$ adalah nilai bobot ke output layer terkoreksi, $w_{kj}^o(t)$ adalah nilai bobot ke output layer sebelumnya, $f_{pk}^{\prime}(net_{pk}^o)$ adalah output dari hidden layer, dan η adalah tingkat pembelajaran,.

Mengatur tingkat pembelajaran yang tepat bisa menjadi tugas sulit, jika tingkat pembelajaran terlalu kecil, algoritma mungkin memakan waktu lama untuk konvergen. Di sisi lain, memilih tingkat belajar yang besar bisa memiliki efek sebaliknya, algoritma bisa menyimpang. Kadang-kadang pada jaringan syaraf tiruan, semua beban memiliki tingkat belajar sendiri. Makalah ini akan menggunakan tingkat 0,05 dan 0,1.

Cara lain untuk meningkatkan kecepatan dan konvergensi adalah dengan menggunakan teknik yang disebut dengan momentum [6], dengan fungsi :

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta \delta_{pk}^o net_{pk}^o + \alpha \Delta_p w_{kj}^o(t-1) \quad (6)$$

Dimana α adalah momentum, $\Delta_p w_{kj}^o(t-1)$ adalah selisih nilai bobot sebelumnya.

Freeman [6] menyatakan besar jangka momentum akan mempengaruhi penyesuaian dalam bobot saat bergerak ke arah yang sama dengan penyesuaian sebelumnya, sehingga mengusahakan solusi yang dapat adalah solusi global minimum. Selanjutnya pada makalah ini akan digunakan momentum 0,5 dan 0,4.

Langkah terakhir pada satu iterasi adalah update bobot dari input ke hidden layer dengan persamaan:

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \eta \delta_{pj}^h x_i \quad (7)$$

Dimana $w_{ji}^h(t+1)$ adalah nilai bobot ke hidden layer terkoreksi, $w_{ji}^h(t)$ adalah nilai bobot ke hidden layer sebelumnya, x_i adalah nilai input.

Algoritma ini berhenti ketika nilai fungsi kesalahan telah menjadi cukup kecil. Fungsi kesalahan dihitung dengan persamaan :

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (8)$$

2.6 Pemodelan Forecasting Container Throughput Dengan Metode Jaringan Saraf Tiruan

Pengolahan data dilakukan dengan mengubah nilai jumlah *input layer*, *hidden layer*, *learning rate*, dan nilai momentum . Berdasarkan nilai MSE dan MAE dari perancangan model jaringan prediksi maka diperoleh model yang fit sebagai berikut:

Tabel 3. Hasil Percobaan dan Standar Error dengan MSE, dan MAE dengan Model Jaringan Syaraf Tiruan Backpropagation

NO	Fungsi Aktivasi	Neuron Counts		Learning			Criteria		
		Input Layer	Hidden Layer	Learning Rate	Momentum	Max Iteration	Error	MSE	MAE
1	Hyperbolic Tangent	12	12	0.05	0.5	10000	0.000004	0.096426	0.194686
2	Hyperbolic Tangent	12	12	0.05	0.4	10000	0.000004	0.092022	0.164802
3	Hyperbolic Tangent	12	12	0.1	0.5	10000	0.000001	0.085698	0.221893
4	Hyperbolic Tangent	12	12	0.1	0.4	10000	0.000002	0.091662	0.177695
5	Hyperbolic Tangent	12	11	0.05	0.5	10000	0.000037	1.783843	0.937968
6	Hyperbolic Tangent	12	11	0.05	0.4	10000	0.000006	0.368694	0.447501
7	Hyperbolic Tangent	12	11	0.1	0.5	10000	0.000001	0.091084	0.262448
8	Hyperbolic Tangent	12	11	0.1	0.4	10000	0.000001	0.063481	0.198176
9	Hyperbolic Tangent	12	10	0.05	0.5	10000	0.000003	0.096119	0.22489
10	Hyperbolic Tangent	12	10	0.05	0.4	10000	0.000017	0.558054	0.401451
11	Hyperbolic Tangent	12	10	0.1	0.5	10000	0.000001	0.071778	0.205472
12	Hyperbolic Tangent	12	10	0.1	0.4	10000	0.000001	0.096659	0.241314
13	Hyperbolic Tangent	12	9	0.05	0.5	10000	0.000013	0.317539	0.390538
14	Hyperbolic Tangent	12	9	0.05	0.4	10000	0.000033	1.815662	1.086474
15	Hyperbolic Tangent	12	9	0.1	0.5	10000	0.000013	3.580141	1.478143
16	Hyperbolic Tangent	12	9	0.1	0.4	10000	0.000001	0.097757	0.248433
17	Hyperbolic Tangent	12	8	0.05	0.5	10000	0.001453	35.05879	3.646136
18	Hyperbolic Tangent	12	8	0.05	0.4	10000	0.001201	30.10512	3.753743
19	Hyperbolic Tangent	12	8	0.1	0.5	10000	0.000279	15.1199	2.841656
20	Hyperbolic Tangent	12	8	0.1	0.4	10000	0.000027	0.82792	0.62162
21	Hyperbolic Tangent	13	12	0.05	0.5	10000	0.000003	0.095995	0.168393

Dari tabel di atas diperoleh model terbaik adalah dengan nilai jumlah *input layer* 12, *hidden layer* 11, *learning rate* 0,1 , dan nilai momentum 0,4. Model ini dapat digunakan

sebagai model peramalan untuk throughput kontainer pada periode yang akan datang karena memiliki nilai MSE dan MAE terkecil.

Langkah selanjutnya untuk memverifikasi model dapat dibandingkan persentase kebenaran (*Accuracy*) dari hasil pelatihan dan pengujian prediksi yang ada dengan data *real* yang diperoleh dari throughput kontainer. Pada penelitian ini digunakan data 6 minggu

Tabel 4. Perbandingan Hasil Real dan Prediksi Throughput kontiner 6 minggu dengan Metode Jaringan Syaraf Tiruan

Minggu ke-	X-true	X-Pred	Residual	Accuracy
111	873	958.1612	-85.16	90.24%
112	1534	753.2656	780.73	49.10%
113	933	1513.628	-580.63	37.77%
114	1130	932.6539	197.35	82.54%
115	1089	641.2074	447.79	58.88%
116	1156	2008.187	-852.19	26.28%
Rata-rata				57.47%

Pada tabel di atas kita dapat lihat bahwa terdapat beberapa data yang tingkat keakuratannya hampir sempurna. Rata-rata ke akuratan peramalan dengan metode jaringan syaraf tiruan adalah 57,47%.

2.7 Model Hybrid ARIMA-Box Jenkins – Jaringan Syaraf Tiruan (JST)

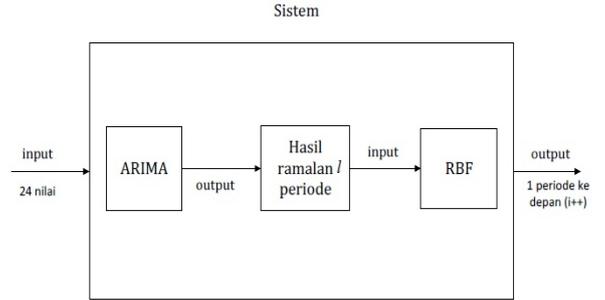
ARIMA dalam statistik dan *neural network* masing-masing telah dibuktikan dan dinyatakan menjadi metode yang handal dalam *time series forecasting*. Dengan mengkombinasikan dua metode tersebut diharapkan akan menghasilkan tingkat keakuratan yang lebih tinggi dibandingkan jika metode tersebut digunakan masing-masing tanpa dikombinasikan. Selain itu alasan penggabungan kedua model ini (ARIMA dan JST) adalah karena adanya asumsi bahwa model tunggal tidak dapat secara total mengidentifikasi semua karakteristik dari time series [13]. Hipotesis dalam penelitian ini adalah bahwa gabungan model ARIMA dengan Jaringan Syaraf Tiruan akan lebih akurat karena struktur autokorelasi kompleks pada data dapat dimodelkan.

Model hybrid ARIMA-Box Jenkins dan Jaringan Syaraf Tiruan merupakan metode gabungan, yaitu hasil ramalan dari metode ARIMA kemudian dikombinasikan dengan hasil ramalan dari model JST. Secara matematis, hasil ramalan secara keseluruhan yang diperoleh adalah sebagai berikut

$$\hat{Z}_t = \hat{L}_t + \hat{N}_t \tag{10}$$

Z_t merupakan hasil peramalan yang merupakan gabungan nilai ramalan dari model ARIMA dan nilai ramalan dari model JST.

Berikut ini adalah arsitektur model peramalan hybrid ARIMA-JST



Gambar 6. Arsitektur model peramalan hybrid ARIMA-JST

2.8 Model hybrid ARIMA-Box Jenkins dan Jaringan Syaraf Tiruan dengan input adalah hasil dari metode ARIMA-Box Jenkins

Pengolahan data model hybrid yang pertama ini dilakukan dengan memperkirakan hasil dari metode ARIMA-Box Jenkins. Hasil peramalan metode ARIMA-Box Jenkins dibentuk dalam 110 data yang selanjutnya di input ke dalam model jaringan syaraf tiruan dengan mengubah nilai jumlah *input layer*, *hidden layer*, *learning rate*, dan nilai momentum. Berdasarkan nilai MSE dan MAE dari perancangan model jaringan prediksi maka diperoleh model yang fit sebagai berikut:

Tabel 5. Hasil Percobaan dan Standar Error dengan MSE, dan MAE dengan Model hybrid ARIMA-Box Jenkins dan Jaringan Syaraf Tiruan dengan input adalah hasil dari metode ARIMA-Box Jenkins

NO	Fungsi Aktivasi	Neuron Counts		Learning			Criteria		
		Input Layer	Hidden Layer	Learning Rate	Momentum	Max Iteration	Error	MSE	MAE
1	Hyperbolic Tangent	12	12	0.05	0.5	10000	0.039954	50.545127	2.108539
2	Hyperbolic Tangent	12	12	0.05	0.4	10000	0.0447	72.195277	2.403272
3	Hyperbolic Tangent	12	12	0.1	0.5	10000	0.047815	91.196591	2.666726
4	Hyperbolic Tangent	12	12	0.1	0.4	10000	0.048516	96.635657	2.80914
5	Hyperbolic Tangent	12	11	0.05	0.5	10000	0.040866	53.44055	2.228568
6	Hyperbolic Tangent	12	11	0.05	0.4	10000	0.04323	70.141102	2.356328
7	Hyperbolic Tangent	12	11	0.1	0.5	10000	0.049217	99.555071	2.968363
8	Hyperbolic Tangent	12	11	0.1	0.4	10000	0.050411	110.620942	3.094759
9	Hyperbolic Tangent	12	10	0.05	0.5	10000	0.051576	119.904087	3.075436
10	Hyperbolic Tangent	12	10	0.05	0.4	10000	0.041553	56.351504	2.294242
11	Hyperbolic Tangent	12	10	0.1	0.5	10000	0.048072	91.019702	2.75413
12	Hyperbolic Tangent	12	10	0.1	0.4	10000	0.051017	115.911138	3.082062
13	Hyperbolic Tangent	12	9	0.05	0.5	10000	0.041152	54.404637	2.219857
14	Hyperbolic Tangent	12	9	0.05	0.4	10000	0.040893	52.542042	2.323272
15	Hyperbolic Tangent	12	9	0.1	0.5	10000	0.046153	79.848583	2.601338
16	Hyperbolic Tangent	12	9	0.1	0.4	10000	0.050074	109.077154	3.099628
17	Hyperbolic Tangent	12	8	0.05	0.5	10000	0.043277	54.669303	2.453341
18	Hyperbolic Tangent	12	8	0.05	0.4	10000	0.049746	53.203506	3.218147
19	Hyperbolic Tangent	12	8	0.1	0.5	10000	0.046922	84.608254	2.691707
20	Hyperbolic Tangent	12	8	0.1	0.4	10000	0.053882	133.542146	3.53765
21	Hyperbolic Tangent	13	12	0.05	0.5	10000	0.042242	55.667245	2.012749

Verifikasi model dengan perbandingan persentase kebenaran (*Accuracy*) dari hasil pelatihan dan pengujian prediksi yang ada dengan data 6 minggu *real* yang diperoleh dari throughput kontainer.

Tabel 6. Perbandingan Hasil Real dan Prediksi Throughput kontiner 6 minggu –Metode ARIMA - Jaringan Syaraf Tiruan dengan input ARIMA

Periode	Real	Forecaste	Residual	Accuracy
15/02/14	873	1311.058	-438	49.82%
22/02/14	1534	1153.6	380	75.20%
1/3/2014	933	1312.779	-380	59.29%
8/3/2014	1130	1080.88	49	95.65%
15/03/14	1089	993.0172	96	91.19%
22/03/14	1156	1301.469	-145	87.42%
Rata-rata				76.43%

Rata-rata ke akuratan peramalan dengan metode Metode ARIMA-Box Jenkins - Jaringan Syaraf Tiruan adalah 76,43%. Hal ini menunjukkan nilai ini lebih besar daripada metode jaringan syaraf tiruan, namun lebih kecil daripada metode ARIMA-Box Jenkins.

3 Kesimpulan

Dari penelitian ini dapat disimpulkan bahwa model yang paling sesuai untuk data throughput container menggunakan model ARIMA bulan Januari 2012 sampai Januari 2014 adalah Model MA (18) dengan nilai RMSE sebesar 334,469 dan MAE sebesar 273,884. Hasil peramalan untuk nilai data *throughput container* 6 minggu mendatang memiliki tingkat akurasi yang cukup baik yaitu rata-rata sebesar 88,712%. Sementara itu jika menggunakan Jaringan Syaraf Tiruan Backpropagation diperoleh arsitektur jaringan syaraf tiruan terbaik dengan jumlah *input layer* 12, *hidden layer* 12, *learning rate* 0,1, dan nilai momentum 0,5. Model merupakan model yang dapat digunakan namun dengan penyempurnanaan. Model Hybrid yang sebaiknya digunakan adalah model hybrid dengan model pertama Jaringan Syaraf Tiruan (arsitektur jaringan input layer 12, hidden layer 12, learning rate 0,05 dan momentum 0,5) dan selanjutnya digunakan model ARIMA (2,0,0).

Daftar Pustaka

- [1] Syafi'i, Kuroda. K, Takebayashi.M. "Forecasting the demand of container throughput in Indonesia". *Memoirs of construction engineering research institute*. Vol.47. Nov.2005.
- [2] Esmer, S. "Performance measurements of container terminal operations". *Dokuz Eylul Üniversitesi .sosyal bilimler enstitüsü dergisi*. Cilt 10, sayi. pp238-255. 2008
- [3] Surjasa, Dadang. 2011. "Rancang Bangun Model Sistem Penunjang Keputusan Cerdas Untuk Sistem Rantai Pasokan Beras Di Propinsi Dki Jakarta". Ph.D.thesis: IPB
- [4] Cryer, J.D., 1986. *Time Series Analysis*, Boston: PWS-KENT Publishing Company.
- [5] Wei, W.W.S., 1990. *Time Series Analysis: Univariate and Multivariate Methods*. AddisonWesley Publishing Co., USA.
- [6] Bowerman, B.L., O'Connel, 1993. *Forecasting and Time Series: An Applied Approach*, 3ed, Belmont, California: Duxbury Press.

Biodata Penulis

Aris Gunaryati, memperoleh gelar Sarjana Sains (S.Si), Jurusan Matematika, lulus tahun 2001. Memperoleh gelar Magister Manajemen Sistem Informasi (MMSI) Program Pasca Sarjana Magister Manajemen Sistem Informasi Bisnis Universitas Gunadarma, lulus tahun 2013. Saat ini menjadi Dosen di Universitas Nasional

Arrahmah Aprilia, memperoleh gelar Sarjana Teknik (S.T), Jurusan Teknik Industri, Universitas Andalas. Memperoleh gelar Master teknik (M.T) Program Pasca Sarjana Teknik Industri Institut Teknologi Bandung, lulus tahun 2011. Saat ini menjadi Dosen di Universitas Nasional