

PEMODELAN GENERAL REGRESSION NEURAL NETWORK UNTUK PREDIKSI TINGKAT PENCEMARAN UDARA KOTA SEMARANG

Budi Warsito¹, Agus Rusgiyono¹ dan M. Afif Amirillah²

¹Program Studi Statistika FMIPA UNDIP

²Alumni Program Studi Statistika FMIPA UNDIP

Abstract

This paper is discuss about General Regression Neural Network (GRNN) modelling to predict time series data, i.e. the air pollution rate in Semarang City comprises the floating dust, carbon monoxide (CO) and nitrogen monoxide (NO). The GRNN model have four processing layer that are input layer, pattern layer, summation layer and output layer. The input variable is determined by the ARIMA model. The result of GRNN modelling shows that the network have a good performance both at predict in sample and predict out of sample, that can be seen from the mean square error.

Keywords: GRNN, predict, air pollution

1. Pendahuluan

Udara yang tercemar dapat merusak lingkungan sekitarnya dan berpotensi mengganggu kesehatan masyarakat sekitar. Menurut Badan Lingkungan Hidup Dunia, *United Nations Environmental Program* (UNEP) pada tahun 1992 Indonesia berada pada urutan ketiga negara terpolusi di dunia setelah Mexico dan Thailand^[5]. Kota Semarang merupakan salah satu dari enam kota di Indonesia dengan kualitas udara yang mengkhawatirkan, bahkan dinyatakan hanya memiliki udara bersih antara 22-62 hari dalam satu tahun^[3]. Oleh karena itu perlu dilakukan studi untuk mengetahui model prediksi tingkat pencemaran udara di kota Semarang sehingga dapat dilakukan peramalan untuk masa-masa yang akan datang. Penelitian terdahulu¹ telah mengindikasikan data kualitas udara merupakan proses stokhastik time series sehingga dimungkinkan untuk membuat peramalan data historis. Dengan mengetahui prediksi masa depan diharapkan dapat dilakukan kebijakan yang tepat sebagai langkah antisipasi.

Tulisan ini membahas pemodelan General Regression Neural Network (GRNN) yang merupakan salah satu model Jaringan Syaraf Tiruan (JST) untuk keperluan prediksi data time series tingkat pencemaran udara kota Semarang. Model GRNN terdiri dari empat layer pemrosesan yaitu *input layer*, *pattern layer*, *summation layer* dan *output layer*. Sebagai input model dipilih berdasarkan model terbaik ARIMA. Penggunaan fungsi aktivasi dilakukan pada *pattern layer*, sedangkan hasil aktivasi diringkas dalam *summation layer* yang meliputi penjumlahan aritmatik dan penjumlahan terboboti. Dari *summation layer* dapat dihitung output jaringan yang merupakan nilai prediksinya.

2. General Regression Neural Network (GRNN)

Teori General Regression Neural Network diperoleh dari estimasi densitas kernel multivariat dan regresi kernel multivariat^[4]. Tujuan dari estimasi multivariat nonparametrik ini yaitu mengestimasi fungsi densitas probabilitas $F(z_1^*, \dots, z_m^*)$ dari m variabel acak $z = (z_1, \dots, z_m)^T$ dengan menggunakan n ukuran dari tiap variabel. Estimator densitas kernel multivariat pada kasus m dimensi didefinisikan sebagai

$$F(z^*) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_1 \dots h_m} K \left(\frac{z_{i1} - z_1^*}{h_1}, \dots, \frac{z_{im} - z_m^*}{h_{m1}} \right) \quad (1)$$

dimana K adalah fungsi kernel multivariat dan panjang bidang (parameter penghalusan) vektor $\mathbf{h} = (h_1, \dots, h_m)^T$.

Data asli $Z(X_i, Y_i); i=1, \dots, n$ akan dibagi menjadi himpunan data pelatihan dan himpunan data pengujian. Himpunan data pelatihan digunakan untuk pengembangan model, sedangkan himpunan data pengujian digunakan untuk estimasi model yang berkualitas. Diasumsikan bahwa data pelatihan berasal dari suatu proses sampling yang mengukur nilai output dengan *additive random noise*:

$$Z_i = E[Z|x, y] + \varepsilon_i \quad (2)$$

dimana $\varepsilon_i \sim \text{NID}(0, \sigma^2)$.

Mean bersyarat dari Z jika diberikan ke (x, y) yang dikenal sebagai suatu regresi Z pada (x, y) adalah suatu solusi yang meminimalkan MSE. Jika $f(x, y, Z)$ adalah fungsi densitas probabilitas kontinu bersama maka mean bersyarat tersebut adalah:

$$E[Z|x, y] = \frac{\int_{-\infty}^{\infty} Z \cdot f(x, y, Z) dZ}{\int_{-\infty}^{\infty} f(x, y, Z) dZ} \quad (3)$$

Fungsi densitas $f(x, y, Z)$ dapat diestimasi dari data dengan menggunakan estimator konsisten nonparametrik yang diusulkan oleh Parzen pada kasus dimensi banyak yang dikembangkan oleh Cacoullos^[6] sebagai berikut :

$$f(x, y, Z) = \frac{1}{[(2\pi)^{3/2} h^3 n]} \sum_{i=1}^n \exp(-D_i^2 / 2h^2) \exp[-(Z - Z^i)^2 / 2h^2] \quad (4)$$

Dengan n adalah banyaknya pengukuran dalam himpunan data pelatihan, h adalah suatu panjang bidang serta jarak metrik (D_i^2) adalah:

$$D_i^2 = (x - x^i)^2 + (y - y^i)^2 \quad (5)$$

Dengan mensubstitusi estimasi probabilitas bersama (4) kedalam mean bersyarat (3) diperoleh estimator kernel Nadaraya-Watson sebagai berikut :

$$Z_m(X, Y) = \frac{\sum_{i=1}^n Z_i \exp(-D_i^2 / 2h^2)}{\sum_{i=1}^n \exp(-D_i^2 / 2h^2)} \quad (6)$$

Struktur dan Arsitektur GRNN

GRNN terdiri dari empat lapisan unit pemroses, dimana tiap lapisan unit pemroses mempunyai fungsi komputasi khusus saat regresi *nonlinier* dibentuk. Lapisan pertama terdiri dari neuron-neuron input yang berfungsi untuk mengambil informasi. Neuron input ini bersifat unik untuk tiap variabel prediktor pada vektor input X . Tidak ada pengolahan data yang terjadi pada neuron-neuron input ini. Neuron-neuron input lalu meneruskan data menuju lapisan kedua yang dinamakan neuron pola. Sebuah neuron-neuron pola berfungsi mengkombinasikan dan memproses data secara sistematis dengan suatu fungsi aktivasi. Jumlah dari neuron pola sama dengan jumlah dari kasus dalam himpunan pelatihan. Neuron

pola i berisi data dari neuron-neuron input dan menghitung output θ_i dengan menggunakan fungsi aktivasi dengan rumus:

$$\theta_i = e^{-(X-U_i)'(X-U_i)/2\sigma^2} \tag{7}$$

dimana X adalah vektor input dari variabel prediktor untuk GRNN, U_i adalah vektor pelatihan yang diwakili neuron pola i dan σ adalah parameter penghalusan.

Output dari neuron pola lalu dikirim menuju lapisan ketiga yang disebut *summation neuron*. Secara teknik, ada dua tipe *summation* yaitu *simple arithmetic summation* (jumlahan aritmatik sederhana) dan *weighted summation* (jumlahan terboboti).

Pada topologi GRNN, unit-unit pelatihan memisah untuk menyelesaikan *simple arithmetic summation* dan *weighted summation*. Formula dari *simple arithmetic summation* dan *weighted summation* adalah sebagai berikut :

$$S_s = \sum_i \theta_i \tag{8}$$

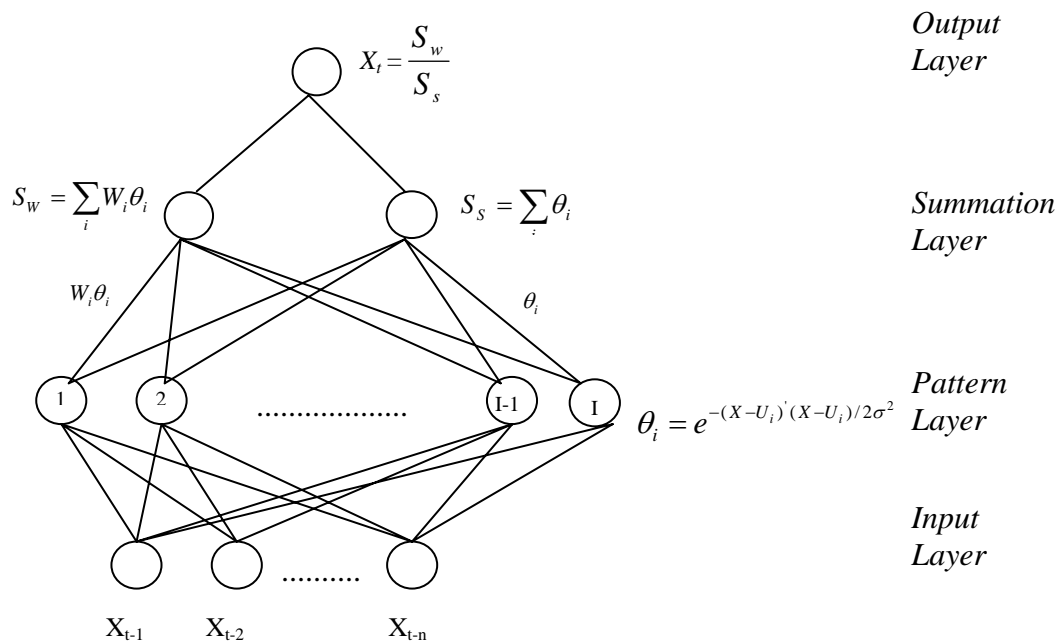
$$S_w = \sum_i W_i \theta_i \tag{9}$$

dimana S_s adalah *simple arithmetic summation* dan S_w adalah *weighted summation*.

Jumlahan dari neuron-neuron *summation* lalu dikirim menuju lapisan keempat yang dinamakan neuron *output*. Neuron output lalu melakukan pembagian berikut untuk memperoleh output GRNN y :

$$y = \frac{S_w}{S_s} \tag{10}$$

Arsitektur GRNN untuk satu unit output (univariat), yang digunakan untuk peramalan data runtun waktu disajikan pada gambar berikut :



Gambar 1. Arsitektur GRNN untuk satu unit output (univariat) yang digunakan untuk peramalan data runtun waktu

3. Prediksi pada Data Pencemaran Udara Kota Semarang

Dalam tulisan ini akan dijelaskan penentuan prediksi menggunakan metode GRNN untuk data pencemaran udara meliputi parameter debu, CO (Karbon Monoksida) dan NO (Nitrogen Monoksida) yang diambil pada Stasiun Pemantau Pencemaran Udara Banyumanik periode 1 Januari 2004 sampai periode 31 Mei 2004 di Kota Semarang. Variabel input dipilih berdasarkan model terbaik ARIMA untuk masing-masing parameter. Data diolah menggunakan software Matlab 6.5 menggunakan Neural Network Toolbox dengan fungsi *newgrnn*^[2] Hasil yang diperoleh dari model GRNN dibandingkan berdasarkan nilai MSE untuk prediksi dengan data pelatihan serta MSE untuk prediksi dengan data pengujian.

Pada parameter debu diperoleh model terbaik yaitu ARIMA (1,1,2) sehingga diambil input data ke t-1 dan data ke t-2. Pada parameter CO diperoleh model terbaik yaitu ARIMA (1,1,2) sehingga diambil input data ke t-1 dan data ke t-2. Sedangkan pada parameter NO diperoleh model terbaik yaitu ARIMA (1,0,0) sehingga diambil input data ke t-1.

Parameter Debu

Pelatihan jaringan GRNN dengan input data debu diambil data ke t-1 dan data ke t-2, jaringan yang terbentuk terdiri dari dua unit input dengan masing-masing 92 data atau vektor inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{92})$ dan $X_2 = (v_1, v_2, \dots, v_{92})$ serta terdapat 92 neuron pada lapisan pola (*pattern layer*), sama dengan jumlah data pada himpunan data pelatihan. Pada pelatihan jaringan GRNN, dilakukan uji coba penggantian nilai *spread* untuk memperoleh nilai peramalan yang mendekati target. Setelah pelatihan dan pembelajaran jaringan yang terjadi pada lapisan pola dan lapisan jumlahan, jaringan lalu menghitung peramalan data debu pada periode t. Kemudian dilanjutkan pada pelatihan jaringan GRNN dengan input data debu pada data ke t-1 sehingga jaringan yang terbentuk terdiri dari satu unit input dengan 93 data atau vektor inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{93})$ serta terdapat 93 neuron pada lapisan pola. Kemudian dilanjutkan pada pelatihan jaringan GRNN dengan input data debu pada data ke t-1, t-2 dan t-3 sehingga jaringan yang terbentuk terdiri dari tiga unit input dengan masing-masing 91 data atau vektor inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{91})$, $X_2 = (v_1, v_2, \dots, v_{91})$ dan $X_3 = (z_1, z_2, \dots, z_{91})$ serta terdapat 91 neuron pada lapisan pola. Tabel 1 menyajikan perbandingan MSE beberapa model GRNN dengan nilai *spread* yang berbeda. Pada tabel 2, disajikan perbandingan MSE beberapa model GRNN untuk input yang berbeda.

Tabel 1. Perbandingan MSE model GRNN data debu dengan nilai *spread* yang berbeda

NO	Nilai Spread	Parameter Debu Input X_{t-1}		Parameter Debu Input X_{t-1} dan X_{t-2}		Parameter Debu Input X_{t-1} , X_{t-2} dan X_{t-3}	
		MSE Pelatihan	MSE Pengujian	MSE Pelatihan	MSE Pengujian	MSE Pelatihan	MSE Pengujian
1	1	78.8688	171.3641	76.4791	159.8941	71.6935	155.7609
2	0.9	76.7337	165.7959	74.5994	154.4378	68.6694	151.2708
3	0.8	74.6341	160.4813	72.5995	149.5535	64.8129	147.8599
4	0.7	72.5785	155.4995	70.3853	145.6596	59.6570	146.2228
5	0.6	70.5657	150.7309	67.8189	143.1925	52.3903	147.0953
6	0.5	68.5654	146.0252	64.5453	142.8123	41.8401	151.8148
7	0.4	66.4648	141.5510	59.6707	145.7129	27.8307	163.4018
8	0.3	63.9931	137.8856	51.1878	152.8338	14.4119	183.2884

9	0.2	60.5180	136.1228	33.5733	161.7439	5.3908	203.5822
10	0.1	54.5008	141.2994	9.0069	170.2011	1.0813	215.4070

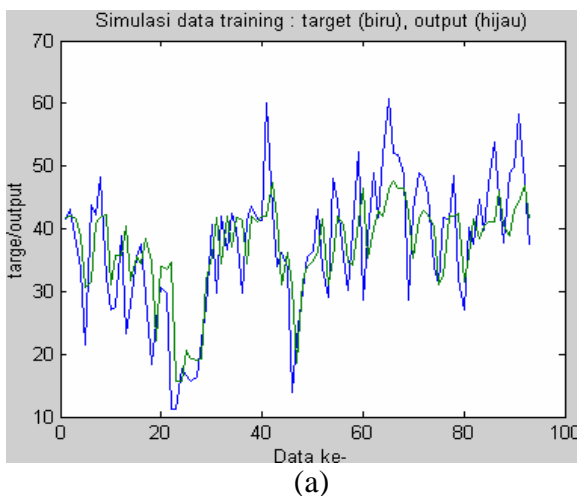
Pada tabel 1 nilai MSE pelatihan turun sebanding dengan mengecilnya nilai *spread*, nilai MSE pengujian dengan input X_{t-1} juga turun sampai pada nilai *spread* = 0.2, setelah nilai *spread* lebih kecil dari 0.2, nilai MSE pengujian menjadi lebih besar dari nilai sebelumnya sehingga diperoleh nilai optimal untuk input X_{t-1} yaitu pada nilai *spread* 0.2. Nilai MSE pengujian dengan input X_{t-1} dan X_{t-2} turun nilainya sebanding dengan mengecilnya nilai *spread* sampai pada nilai *spread* = 0.5. Setelah nilai *spread* lebih kecil dari 0.5, nilai MSE pengujian menjadi lebih besar dari nilai sebelumnya sehingga diperoleh nilai optimal untuk input X_{t-1} dan X_{t-2} yaitu pada nilai *spread* 0.5.

Nilai MSE pengujian dengan input X_{t-1} , X_{t-2} dan X_{t-3} nilainya turun sebanding dengan mengecilnya nilai *spread* sampai pada nilai *spread* = 0.7. Setelah nilai *spread* lebih kecil dari 0.7, nilai MSE pengujian menjadi lebih besar dari nilai sebelumnya sehingga diperoleh nilai optimal untuk input X_{t-1} , X_{t-2} dan X_{t-3} yaitu pada nilai *spread* 0.7.

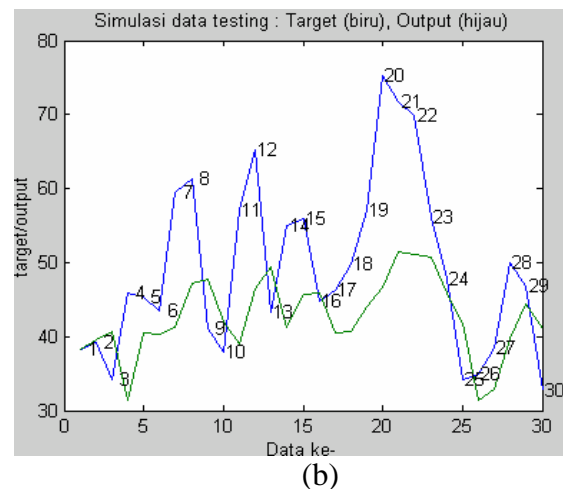
Tabel 2. Perbandingan MSE model GRNN untuk input yang berbeda pada data debu

Input	MSE Pelatihan	MSE Pengujian
X_{t-1}	60.5180	136.1228
X_{t-1} dan X_{t-2}	64.5453	142.8123
X_{t-1} , X_{t-2} dan X_{t-3}	59.6570	146.2228

Pada tabel 2 dapat dilihat bahwa nilai MSE pelatihan untuk input X_{t-1} , X_{t-2} dan X_{t-3} paling kecil, tetapi nilai MSE pengujian untuk input X_{t-1} yang paling kecil. Pemodelan GRNN ini lebih berfokus pada ketepatan peramalan yang ditujukan pada perolehan MSE pengujian terkecil sehingga model yang dipilih yaitu model dengan input X_{t-1} dan nilai *spread* = 0.2.



Gambar 2a. Plot hasil pelatihan model GRNN dengan input X_{t-1} pada data debu



Gambar 2b. Plot hasil pengujian model GRNN dengan input X_{t-1} pada data debu

Pada gambar 2a dan 2b dapat dilihat perbandingan antara target dengan output jaringan untuk data pelatihan dan data pengujian saling berdekatan.

Parameter CO

Pada pelatihan jaringan GRNN dengan input data CO diambil data ke t-1 dan t-2, jaringan yang terbentuk terdiri dari dua unit input dengan masing-masing 120 data atau vektor inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{120})$ dan $X_2 = (v_1, v_2, \dots, v_{120})$ serta terdapat 120 neuron pada lapisan pola (*pattern layer*), sama dengan jumlah data pada himpunan data pelatihan. Kemudian dilanjutkan pada pelatihan jaringan GRNN dengan input data ke t-1 sehingga jaringan yang terbentuk terdiri dari satu unit input dengan 121 data atau vektor inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{121})$ serta terdapat 121 neuron pada lapisan pola. Kemudian dilanjutkan pada pelatihan jaringan dengan input data ke t-1, t-2 dan t-3 sehingga jaringan yang terbentuk terdiri dari tiga unit input dengan masing-masing 119 data atau vektor inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{119})$, $X_2 = (v_1, v_2, \dots, v_{119})$ dan $X_3 = (z_1, z_2, \dots, z_{119})$ serta terdapat 119 neuron pada lapisan pola. Pada tabel 3, disajikan perbandingan MSE beberapa model GRNN dengan nilai *spread* yang berbeda. Pada tabel 4, disajikan perbandingan MSE beberapa model GRNN untuk input yang berbeda.

Tabel 3. Perbandingan MSE model GRNN data CO dengan nilai *spread* yang berbeda

NO	Nilai Spread	Parameter CO Input Xt-1		Parameter CO Input Xt-1 dan Xt-2		Parameter CO Input Xt-1; Xt-2 dan Xt-3	
		MSE Pelatihan	MSE Pengujian	MSE Pelatihan	MSE Pengujian	MSE Pelatihan	MSE Pengujian
1	1	0.0670	0.0538	0.0605	0.0536	0.0545	0.0535
2	0.9	0.0657	0.0536	0.0583	0.0535	0.0515	0.0535
3	0.8	0.0644	0.0535	0.0559	0.0535	0.0480	0.0537
4	0.7	0.0631	0.0536	0.0531	0.0539	0.0436	0.0542
5	0.6	0.0618	0.0539	0.0497	0.0547	0.0382	0.0554
6	0.5	0.0606	0.0545	0.0456	0.0561	0.0317	0.0588
7	0.4	0.0592	0.0554	0.0404	0.0587	0.0245	0.0688
8	0.3	0.0575	0.0569	0.0331	0.0630	0.0159	0.0926
9	0.2	0.0555	0.0586	0.0214	0.0718	0.0060	0.1253
10	0.1	0.0514	0.0570	0.0067	0.1114	0.0003	0.1407

Pada tabel 3 nilai MSE pelatihan turun sebanding dengan mengecilnya nilai *spread*, nilai MSE pengujian dengan input X_{t-1} juga turun sampai pada nilai *spread* = 0.8. Setelah nilai *spread* lebih kecil dari 0.8, nilai MSE pengujian menjadi lebih besar dari nilai sebelumnya sehingga diperoleh nilai optimal untuk input X_{t-1} pada nilai *spread* 0.8.

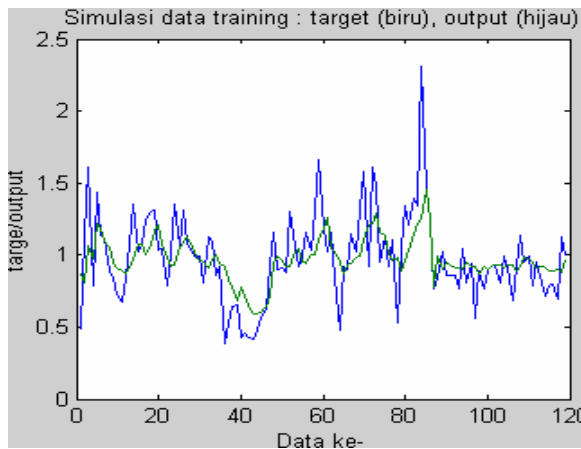
Nilai MSE pengujian dengan input X_{t-1} dan X_{t-2} turun nilainya sebanding dengan mengecilnya nilai *spread* sampai pada nilai *spread* = 0.8. Setelah nilai *spread* lebih kecil dari 0.8, nilai MSE pengujian menjadi lebih besar dari nilai sebelumnya sehingga diperoleh nilai optimal untuk input X_{t-1} dan X_{t-2} pada nilai *spread* 0.8.

Nilai MSE pengujian dengan input X_{t-1} , X_{t-2} dan X_{t-3} pada nilai *spread* = 1 nilainya sama dengan MSE pada nilai *spread* = 0.9. Setelah nilai *spread* lebih kecil dari 0.9, nilai MSE pengujian menjadi lebih besar dari nilai sebelumnya sehingga diperoleh nilai optimal untuk input X_{t-1} , X_{t-2} dan X_{t-3} pada nilai *spread* 0.9.

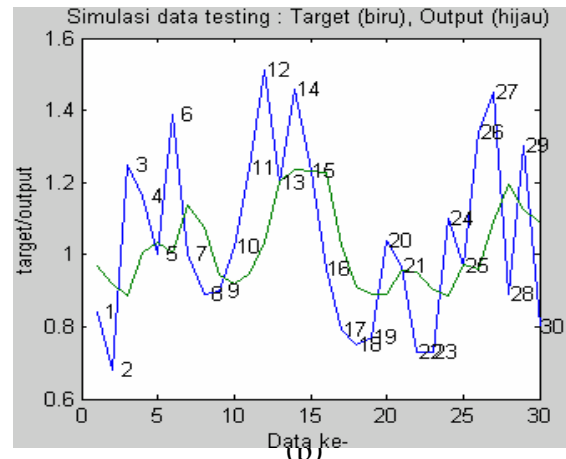
Tabel 4. Perbandingan MSE model GRNN untuk input yang berbeda pada data CO

Input	MSE Pelatihan	MSE Pengujian
X_{t-1}	0.0644	0.0535
X_{t-1} dan X_{t-2}	0.0559	0.0535
X_{t-1}, X_{t-2} dan X_{t-3}	0.0515	0.0535

Pada tabel 4 dapat dilihat bahwa MSE pelatihan input X_{t-1} , X_{t-2} dan X_{t-3} paling kecil daripada input yang lain. Pada MSE pengujian, nilainya sama untuk semua input. Sehingga model yang dipilih yaitu model dengan input X_{t-1} , X_{t-2} dan X_{t-3} dan nilai *spread* = 0.9.



Gambar 3a. Plot hasil pelatihan model GRNN dengan input data X_{t-1} , X_{t-2} dan X_{t-3} pada data CO



Gambar 3b. Plot hasil pengujian model GRNN dengan input data X_{t-1} , X_{t-2} dan X_{t-3} pada data CO

Pada gambar 3 dapat dilihat perbandingan antara target dengan output jaringan untuk data pelatihan dan data pengujian data CO saling berdekatan.

Parameter NO

Pada pelatihan jaringan GRNN data NO dengan input data ke t-1, jaringan yang terbentuk terdiri dari satu unit input dengan 121 data atau vektor inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{121})$ serta terdapat 121 neuron pada lapisan pola (*pattern layer*). Kemudian dilanjutkan pada pelatihan jaringan dengan input data ke t-1 dan t-2 sehingga jaringan yang terbentuk terdiri dari dua unit input dengan masing-masing 120 data atau vektor inputnya dapat ditulis sebagai $X_1 = (x_1, x_2, \dots, x_{120})$, $X_2 = (v_1, v_2, \dots, v_{120})$ serta terdapat 120 neuron pada lapisan pola. Tabel 5 menyajikan perbandingan MSE beberapa model GRNN dengan nilai *spread* yang berbeda. Pada tabel 6, disajikan perbandingan MSE beberapa model GRNN untuk input yang berbeda.

Tabel 5. Perbandingan MSE model GRNN data NO dengan nilai *spread* yang berbeda

NO	Nilai Spread	Parameter NO Input X_{t-1}		Parameter NO Input X_{t-1} dan X_{t-2}	
		MSE Pelatihan	MSE Pengujian	MSE Pelatihan	MSE Pengujian
1	1	41.4328	47.3044	37.8336	49.2643
2	0.9	40.7203	46.5182	36.1810	48.6128
3	0.8	39.9014	45.7891	34.2050	48.2263
4	0.7	38.9760	45.2227	31.8953	48.4863
5	0.6	38.0011	45.0732	29.3465	50.1060
6	0.5	37.0987	45.8161	26.6337	54.1733
7	0.4	36.2692	47.5509	23.5238	61.4773
8	0.3	35.1134	49.5201	19.2901	72.0858
9	0.2	32.9648	52.8411	12.5560	84.2256
10	0.1	27.5247	59.3128	3.7909	92.9568
11	0.09	26.5418	60.3825	2.9953	93.7294
12	0.08	25.4669	61.5747	2.2815	94.5376
13	0.07	24.2993	63.1172	1.6818	95.3250

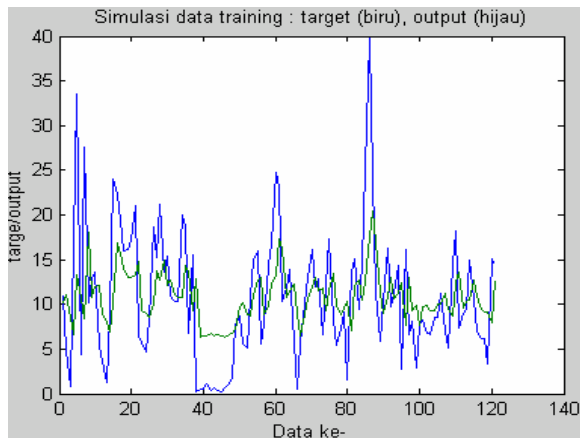
Pada tabel 5 nilai MSE pelatihan turun sebanding dengan mengecilnya nilai *spread*, nilai MSE pengujian dengan input X_{t-1} juga turun sampai pada nilai *spread* = 0.6. Setelah nilai *spread* lebih kecil dari 0.6, nilai MSE pengujian menjadi lebih besar dari nilai sebelumnya sehingga diperoleh nilai optimal untuk input X_{t-1} pada nilai *spread* 0.6.

Nilai MSE pengujian dengan input X_{t-1} dan X_{t-2} turun nilainya sebanding dengan mengecilnya nilai *spread* sampai pada nilai *spread* = 0.8. Setelah nilai *spread* lebih kecil dari 0.8, nilai MSE pengujian menjadi lebih besar dari nilai sebelumnya sehingga diperoleh nilai optimal untuk input X_{t-1} dan X_{t-2} pada nilai *spread* 0.8.

Tabel 6. Perbandingan MSE model GRNN untuk input yang berbeda pada data NO

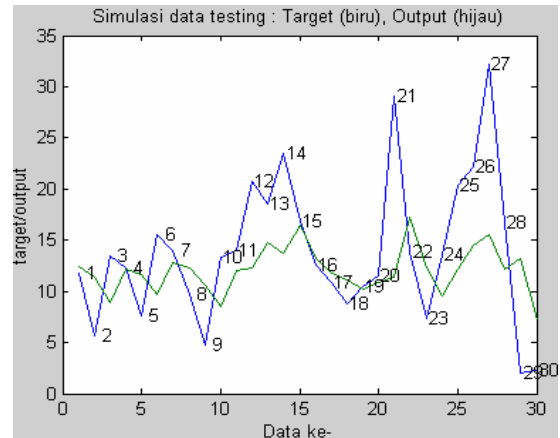
Input	MSE Pelatihan	MSE Pengujian
X_{t-1}	38.0011	45.0732
X_{t-1} dan X_{t-2}	34.2050	48.2263

Pada tabel 6 dapat dilihat bahwa MSE pelatihan input X_{t-1} dan X_{t-2} paling kecil, akan tetapi nilai MSE pengujian untuk input X_{t-1} lebih kecil daripada input X_{t-1} dan X_{t-2} . Pada pemodelan GRNN ini lebih berfokus pada ketepatan peramalan yang ditujukan pada perolehan MSE pengujian terkecil sehingga model yang dipilih yaitu model dengan input X_{t-1} dengan nilai *spread* = 0.6.



(a)

Gambar 4a. Plot hasil pelatihan model GRNN dengan input data X_{t-1} pada data NO



(b)

Gambar 4b. Plot hasil pengujian model GRNN dengan input data X_{t-1} pada data NO

Pada gambar 4a dan 4b dapat dilihat perbandingan antara target dengan output jaringan untuk data pelatihan dan data pengujian data NO saling berdekatan.

5. KESIMPULAN

Pembahasan pada tulisan ini meliputi pembentukan model General Regression Neural Network (GRNN) dan penerapannya pada data runtun waktu. Hasil perhitungan terhadap data pencemaran udara di kota Semarang untuk parameter debu, CO dan NO menunjukkan bahwa model GRNN menghasilkan keakuratan prediksi yang cukup baik.

DAFTAR PUSTAKA

1. Barai, S.V., Dikshit, A.A., and Sharma, S., Neural Network Models for Air Quality Prediction : A Comparative Study, *Working Paper*, 2006.
2. Demuth, H. and Beale M., *Neural Network Toolbox for Use in MATLAB*, Mathwork, Inc., USA, 1998.
3. *Kompas*, Polusi Udara Makin Mencemaskan – Kota Semarang Paling Parah, 3 Februari 2005.
4. Leung, M. T., Chen A. S. and Daouk H, *Forecasting Exchange Rates Using General Regression Neural Network*, University of Texas, USA.
5. *Ozon*, Indonesia Tak Jadi BebasTimbel, Februari 2003, Vol. 4, No. 5.
6. Specht, F. D., A General Regression Neural Network, *IEEE Trans. Neural Network.*, Nov. 1991, Vol. 2, No.6: 568-576.