

PREDIKSI CUACA MENGGUNAKAN ALGORITMA PARTICLE SWARM OPTIMIZATION-NEURAL NETWORK (PSONN)

Dinita Rahmalia¹⁾, Teguh Herlambang²⁾

¹⁾Program Studi Matematika, Universitas Islam Darul Ulum Lamongan
Jl. Airlangga 3 Sukodadi Lamongan
dinitarahmalia@gmail.com

²⁾Program Studi Sistem Informasi, Universitas Nahdlatul Ulama Surabaya
Jl. Jemursari 51-57 Surabaya
teguh@unusa.ac.id

Abstract— Prediksi cuaca terutama dalam menentukan intensitas matahari merupakan hal yang sangat penting pada masalah pemanfaatan energi. Intensitas matahari dipengaruhi oleh faktor seperti suhu udara dan kelembaban udara. Algoritma Neural Network menggunakan Backpropagasi telah dilakukan dalam prediksi cuaca. Cara kerja algoritma ini menyerupai sistem syaraf manusia. Dalam algoritma ini, terdapat nilai bobot yang diupdate dengan cara propagasi maju dan propagasi mundur pada proses training dan testing sehingga menghasilkan nilai prediksi yang mendekati nilai target. Dalam penelitian ini, akan digunakan metode hybrid Particle Swarm Optimization-Neural Network (PSONN) pada masalah prediksi cuaca. Dalam PSONN akan dibentuk populasi dari matriks bobot awal yang menghubungkan input, hidden layer, dan output. Kemudian akan diupdate berdasarkan nilai Mean Square Error (MSE) pada jumlah iterasi yang ditentukan. Hasil simulasi menunjukkan bahwa algoritma PSONN dapat menghasilkan nilai prediksi cuaca yang mendekati nilai target.

Keywords— Prediksi Cuaca, Particle Swarm Optimization, Neural Network, Backpropagasi

I. PENDAHULUAN

Prediksi cuaca merupakan hal yang sangat penting bagi beberapa aktivitas. Dalam prediksi cuaca, terdapat banyak kondisi yang dapat diamati misalkan suhu udara, kelembaban udara, intensitas matahari, dan sebagainya. Prediksi cuaca terutama dalam menentukan intensitas matahari merupakan hal yang sangat penting pada masalah pemanfaatan energi. Menurut data dari kementerian ESDM pada tahun 2005-2010, jumlah konsumsi bahan bakar minyak (BBM) di Indonesia lebih besar daripada jumlah produksi BBM yang mengakibatkan pemerintah melakukan impor BBM. Sebagai upaya untuk mengendalikan jumlah pemakaian BBM, saat ini telah dilakukan pemanfaatan energi matahari menjadi energi listrik melalui sel surya. Dalam penelitian ini, akan dibahas tentang prediksi cuaca yang berupa prediksi intensitas matahari.

Neural Network (NN) diperkenalkan oleh McCulloch dan Pitts pada tahun 1943. Cara kerja neural network mirip dengan sistem syaraf manusia. Salah satu jenis NN yang digunakan untuk masalah prediksi adalah Backpropagasi dimana model Backpropagasi adalah terdapat hidden layer di antara input dan output (Siang, 2009).

Algoritma Particle Swarm Optimization (PSO) diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995. Cara kerja PSO menyerupai sekumpulan burung atau ikan dalam menemukan sumber makanan dimana setiap individu disebut partikel dan populasi disebut swarm (koloni). PSO diinisialisasi dengan sekumpulan partikel sebagai kandidat solusi pada posisi acak. Setiap partikel diberikan posisi awal dan kecepatan awal. Ketika suatu partikel menemukan arah sumber makanan, partikel lainnya akan mengikuti menuju sumber makanan. Salah satu aplikasi PSO adalah untuk menemukan pendekatan solusi optimal pada masalah optimisasi dengan tidak berkendala maupun dengan kendala (Rahmalia, 2017).

Pada beberapa penelitian sebelumnya, telah dilakukan penelitian tentang estimasi intensitas matahari dengan input suhu udara, kelembaban udara (Hidayatullah, 2014), penelitian tentang prediksi curah hujan dengan input suhu udara, kelembaban udara (Lu, 2014). Karena hasil prediksi ditentukan oleh pemilihan matriks bobot awal (Siang, 2009), maka pada penelitian ini akan digunakan algoritma Particle Swarm Optimization-Neural Network (PSONN) pada penentuan matriks bobot awal. Algoritma PSONN adalah algoritma yang menggabungkan algoritma PSO dan algoritma NN dalam menemukan matriks bobot awal yang optimum. Dalam PSONN akan dibentuk populasi dari matriks bobot awal yang menghubungkan input, hidden layer, dan output. Kemudian akan diupdate berdasarkan nilai Mean Square Error (MSE) pada jumlah iterasi yang ditentukan.

Hasil simulasi menunjukkan bahwa algoritma PSONN dapat menghasilkan nilai prediksi cuaca yang mendekati nilai target.

II. NEURAL NETWORK

Neural Network (NN) diperkenalkan oleh McCulloch dan Pitts pada tahun 1943. Cara kerja neural network mirip dengan sistem syaraf manusia yaitu :

1. Sinyal dikirim di antara neuron melalui penghubung.
2. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.
3. Untuk menentukan keluaran, setiap neuron menggunakan fungsi aktivasi yang dikenakan pada jumlahan masukan yang diterima.

Dalam NN, fungsi aktivasi dipakai untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah kombinasi linier antara masukan dengan bobot.

$$net = \sum_i x_i w_i \quad (1)$$

$$f(net) = f\left(\sum_i x_i w_i\right) \quad (2)$$

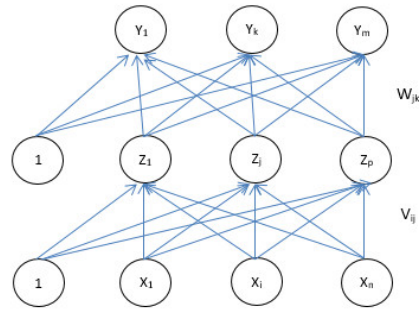
Pada umumnya, sifat fungsi aktivasi adalah : kontinu, terdiferensial, dan tidak turun. Pada penelitian ini, fungsi aktivasi yang dipakai adalah sigmoid biner dengan range (0-1) (Siang, 2009).

$$f(net) = \frac{1}{1 + e^{-net}} \quad (3)$$

$$f'(net) = f(net)(1 - f(net)) \quad (4)$$

Algoritma Backpropagasi

Backpropagasi adalah salah satu jenis NN yang sering digunakan dalam proses prediksi. Algoritma Backpropagasi terdiri dari beberapa unit input x_1, x_2, \dots, x_n , beberapa unit hidden layer z_1, z_2, \dots, z_p , dan beberapa unit output y_1, y_2, \dots, y_m . Dalam unit input dan hidden layer terdapat bias yang bernilai 1. Bobot v_{ij} menghubungkan unit input x_i menuju unit hidden layer z_j . Bobot w_{jk} menghubungkan unit hidden layer z_j menuju unit output y_k . Dalam algoritma Backpropagasi, terdapat tiga fase perhitungan yaitu : propagasi maju, propagasi mundur, dan update matriks bobot. Model Backpropagasi dapat dilihat pada Gambar 1.



Gambar 1. Model Backpropagasi

Pada propagasi maju, perhitungan menggunakan fungsi aktivasi dilakukan mulai dari input, hidden layer, dan output secara berurutan. Pada propagasi mundur, perhitungan faktor kesalahan dilakukan mulai dari output, hidden layer, dan input secara berurutan. Setelah itu, dilakukan perubahan nilai pada matriks bobot (Fausett, 1994).

Algoritma Backpropagasi menggunakan satu hidden layer dapat dinyatakan sebagai berikut :

$$bpggt(V, W)$$

1. Inialisasi matrik bobot V dan W dengan bilangan acak kecil antara -0,5 sampai 0,5.
 $e = 1$

while($e \leq \max epoch$ && $MSE \geq \min MSE$)

for($d = 1 : \text{datasize}$)

2. Tiap unit input menerima sinyal dan meneruskan ke unit hidden layer.

Propagasi Maju

3. Hitung semua keluaran di unit hidden layer $z_j, j = 1, 2, \dots, p$

$$z_net_j = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad (5)$$

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \quad (6)$$

4. Hitung semua keluaran di unit output $y_k, k = 1, 2, \dots, m$

$$y_net_k = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad (7)$$

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \quad (8)$$

Propagasi Mundur

5. Hitung faktor δ unit output berdasarkan kesalahan di setiap unit output $y_k, k = 1, 2, \dots, m$

$$\delta_k = (t_k - y_k) f'(y_net_k) \quad (9)$$

6. Hitung perubahan bobot

$$\Delta w_{jk} = \alpha \delta_j z_j \quad (10)$$

$$k = 1, 2, \dots, m \quad j = 0, 1, 2, \dots, p$$

7. Hitung faktor δ unit hidden layer berdasarkan kesalahan di setiap unit layer tersembunyi $z_j, j=1, 2, \dots, p$

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (11)$$

$$\delta_j = \delta_{net_j} f'(z_{net_j})$$

8. Hitung perubahan bobot

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (12)$$

$$j = 1, 2, \dots, p \quad i = 0, 1, 2, \dots, n$$

Update Matriks Bobot

9. Hitung matriks bobot yang baru

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk} \quad (13)$$

$$v_{ij} \leftarrow v_{ij} + \Delta v_{ij} \quad (14)$$

end

10. Hitung nilai MSE

$$MSE = \frac{1}{datasize} \sum_{d=1}^{datasize} \frac{1}{m} \sum_{k=1}^m (T_{dk} - Y_{dk})^2 \quad (15)$$

11. $e \leftarrow e + 1$

end

Proses Prediksi

Salah satu aplikasi dari algoritma Backpropagasi adalah untuk proses prediksi. Pada penelitian ini, data cuaca berupa suhu udara, kelembaban udara, serta intensitas matahari akan digunakan. Tahapan dalam proses prediksi cuaca adalah sebagai berikut (Siang, 2009) :

Persiapan Data : Dari data cuaca yang diperoleh, tentukan input dan output menggunakan korelasi. Berdasarkan hasil korelasi, intensitas matahari dipengaruhi oleh suhu udara dan kelembaban udara, maka input yaitu : suhu udara (Celcius) dan kelembaban udara (%) serta output yaitu intensitas matahari (Watt/m²). Pada penelitian ini partisi data yang akan digunakan untuk data training sebesar 80% dari seluruh data dan data testing sebesar 20% dari seluruh data.

Transformasi Input dan Output ke dalam Range [0.1-0.9] : Fungsi aktivasi yang digunakan dalam proses prediksi adalah fungsi sigmoid biner yang memiliki nilai minimum 0 dan maksimum 1. Namun karena fungsi sigmoid biner tidak pernah menyentuh nilai 0 maupun 1, maka input dan

output harus ditransformasikan ke dalam range [0.1-0.9] sebelum proses training dimulai.

Inisialisasi Matriks Bobot: Sebelum proses training dimulai, tentukan matriks bobot awal yang menghubungkan input, hidden layer dan output dengan bilangan acak kecil antara -0,5 sampai 0,5.

Proses Training: Proses training menggunakan data training yang telah ditentukan sebelumnya. Aturan pemberhentian dalam proses training dapat berupa maksimum epoch atau nilai MSE terkecil telah dicapai.

Matriks Bobot Baru: Proses training pada data training menghasilkan nilai matriks bobot yang baru. Matriks bobot inilah yang akan digunakan pada proses training data testing.

Proses Testing: Proses training menggunakan data testing yang telah ditentukan sebelumnya. Aturan pemberhentian adalah 1 epoch untuk menghitung nilai MSE. Nilai MSE adalah nilai rata-rata dari kuadrat selisih nilai target dengan nilai prediksi. Nilai MSE dapat dihitung dari Eq. 15.

De-transformasi Data: Setelah nilai MSE terkecil tercapai, lakukan de-transformasi pada data testing untuk mengembalikan data ke dalam bentuk asalnya sehingga dapat diinterpretasikan.

III. PARTICLE SWARM OPTIMIZATION-NEURAL NETWORK (PSO-NN)

Pada penelitian ini, akan digunakan metode PSO-NN, yang menggabungkan algoritma PSO dan algoritma NN dalam menemukan matriks bobot awal yang optimum dengan nilai MSE yang optimum.

Dalam PSO-NN, suatu partikel dapat direpresentasikan ke dalam bentuk pasangan matriks bobot awal yang menghubungkan input, hidden layer, dan output. Kemudian dihimpun membentuk kumpulan partikel (swarm). Proses optimisasi dijalankan dengan nilai fitness adalah nilai MSE.

Particle Swarm Optimization

PSO diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995. Cara kerja PSO menyerupai sekumpulan burung atau ikan dalam menemukan sumber makanan dimana setiap individu disebut partikel dan populasi disebut swarm (koloni). PSO diinisialisasi dengan sekumpulan partikel sebagai kandidat solusi pada posisi acak. Setiap partikel diberikan posisi awal dan kecepatan awal. Ketika suatu partikel menemukan arah sumber makanan, partikel lainnya akan mengikuti menuju sumber makanan.

Dalam PSO, nilai posisi partikel $X(t+1)$ diupdate berdasarkan nilai kecepatan partikel $U(t+1)$ dengan persamaan berikut (Kennedy dan Eberhart, 1995):

$$U^s(t+1) = \omega U^s(t) + c_1 r_1 (Pbest^s - X^s(t)) + c_2 r_2 (Gbest - X^s(t)) \quad (16)$$

$$X^s(t+1) = X^s(t) + U^s(t+1) \quad (17)$$

Nilai ω adalah bobot inersia dengan nilai 0,9-1,2. Variabel $Pbest^s$ adalah local best particle dari partikel S , yaitu posisi partikel S yang terbaik dari waktu awal sampai waktu sekarang. Variabel $Gbest$ adalah global best particle yang dipilih dari $Pbest^s$ yang memiliki posisi terbaik. Nilai r_1 dan r_2 adalah bilangan acak seragam antara 0-1, sedangkan $c_1 = c_2 = 2$ (Kennedy dan Eberhart, 1995).

Algoritma PSONN

Algoritma PSONN menggabungkan algoritma PSO dan algoritma NN dalam menemukan matriks bobot awal yang optimum (Rajkumar, 2015).

Dalam PSONN, suatu partikel dapat direpresentasikan ke dalam bentuk pasangan matriks bobot awal yang menghubungkan input, hidden layer, dan output sebagai berikut :

$$X = (V, W) = \left(\begin{bmatrix} b_1 & \dots & b_p \\ v_{11} & \dots & v_{1p} \\ \vdots & \vdots & \vdots \\ v_{n1} & \dots & v_{np} \end{bmatrix}, \begin{bmatrix} b_1 & \dots & b_m \\ w_{11} & \dots & w_{1m} \\ \vdots & \vdots & \vdots \\ w_{p1} & \dots & w_{pm} \end{bmatrix} \right) \quad (18)$$

Dimana v_{ij} adalah bobot yang menghubungkan input x_i menuju hidden layer z_j sedangkan w_{jk} adalah bobot yang menghubungkan hidden layer z_j menuju output y_k . Nilai b adalah bobot bias.

Langkah-langkah algoritma PSONN adalah sebagai berikut :

1. Inisialisasi sejumlah NS posisi partikel X berupa pasangan matriks bobot V dan W dengan bilangan acak antara -0,5 sampai 0,5.

$$X^s(0) = (V^s(0), W^s(0)), \quad s = 1, 2, \dots, NS \quad (19)$$

2. Inisialisasi kecepatan partikel U

$$U^s(0), s = 1, 2, \dots, NS \quad (20)$$

3. Tentukan local best particle $Pbest^s$

$$Pbest^s = X^s(0), s = 1, 2, \dots, NS \quad (21)$$

4. Hitung fitness function menggunakan algoritma Backpropagation

$$f(Pbest^s) = bpggt(V^s(0), W^s(0)) \quad s = 1, 2, \dots, NS \quad (22)$$

5. Tentukan global best particle $Gbest$

$$Gbest = \underset{Pbest}{\operatorname{argmin}}(f(Pbest^s), s = 1, 2, \dots, NS) \quad (23)$$

6. Update posisi dan kecepatan partikel
for $t = 0 : \max_t - 1$
for $s = 1 : NS$
 - Hitung kecepatan partikel berdasarkan persamaan Eq. 16.
 - Update posisi partikel berdasarkan persamaan Eq. 17.
 - Hitung fitness function posisi partikel menggunakan algoritma Backpropagation

$$f(X^s(t+1)) = bpggt(V^s(t+1), W^s(t+1)) \quad s = 1, 2, \dots, NS \quad (24)$$

- Tentukan local best particle $Pbest^s$

$$Pbest^s = \underset{X}{\operatorname{argmin}}(f(X^s(0)), f(X^s(1)), \dots, f(X^s(t+1))) \quad s = 1, 2, \dots, NS \quad (25)$$

end

- Tentukan global best particle $Gbest$

$$Gbest = \underset{Pbest}{\operatorname{argmin}}(f(Pbest^s), s = 1, 2, \dots, NS) \quad (26)$$

end

Kemudian akan diperoleh matriks bobot awal yang optimum dimana fitness function yang digunakan adalah nilai Mean Square Error (MSE) (Rajkumar, 2015).

$$MSE_s = \frac{1}{N} \sum_{i=1}^N \frac{1}{m} \sum_{k=1}^m (T_{ik} - P_{ik})^2, s = 1, 2, \dots, NS \quad (27)$$

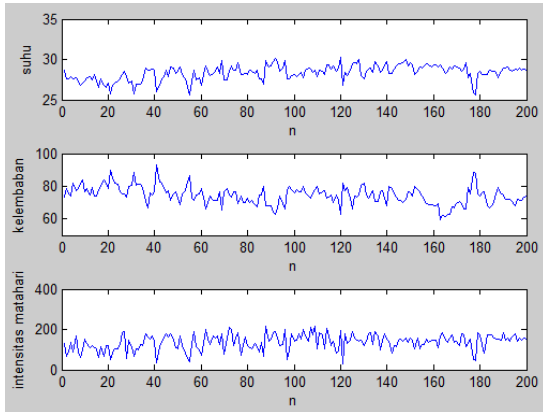
Dengan N adalah jumlah pola training, T_{ik} adalah nilai target pada pola i di keluaran y_k , P_{ik} adalah nilai prediksi pada pola i di keluaran y_k .

IV. HASIL SIMULASI

Hasil simulasi adalah berupa nilai prediksi yang dibandingkan dengan nilai target menggunakan PSONN.

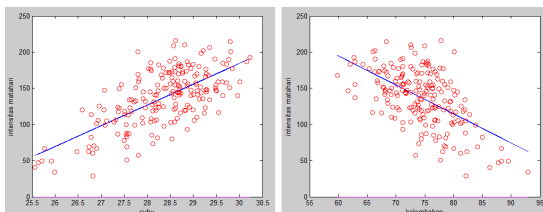
Pola Hubungan pada Variabel

Data yang digunakan adalah data cuaca pada bulan Januari-Juli 2012 yang diambil pada daerah Jakarta. Informasi tentang cuaca terbagi dalam : suhu udara (Celcius), kelembaban udara (%), dan intensitas matahari (Watt/m²). Gambar 2. menunjukkan suhu udara, kelembaban udara, dan intensitas matahari selama bulan Januari-Juli 2012 yang terdiri dari 200 hari.



Gambar 2. Data suhu udara, kelembaban udara, dan intensitas matahari selama Januari-Juli 2012.

Dari parameter suhu udara, kelembaban udara, dan intensitas matahari, akan dianalisis hubungan antara suhu udara, kelembaban udara terhadap intensitas matahari. Pola hubungan antara suhu udara x_1 dan intensitas matahari y dapat dilihat pada Gambar 3(kiri) sedangkan pola hubungan kelembaban udara x_2 dan intensitas matahari y dapat dilihat pada Gambar 3 (kanan). Gambar 4 menunjukkan pola gabungan hubungan suhu udara x_1 dan kelembaban udara x_2 terhadap intensitas matahari y .



Gambar 3. Pola hubungan antara suhu udara dan intensitas matahari (kiri). Pola hubungan antara kelembaban udara dan intensitas matahari (kanan)

Untuk mengetahui hubungan antara suhu udara dan intensitas matahari, dapat dihitung nilai korelasi antara suhu udara dengan intensitas matahari.

$$r_{x_1,y} = \frac{S_{x_1,y}}{\sqrt{S_{x_1,x_1} S_{yy}}} \quad (28)$$

Dengan :

$$S_{x_1,y} = \frac{1}{n} \sum_{i=1}^n (x_{1i} - \bar{x}_1) (y_i - \bar{y})$$

$$S_{x_1,x_1} = \frac{1}{n} \sum_{i=1}^n (x_{1i} - \bar{x}_1)^2$$

$$S_{yy} = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$$

Dari hasil perhitungan, diperoleh korelasi positif sebesar $r_{x_1,y} = 0,676$ yang artinya semakin tinggi suhu udara, intensitas matahari meningkat.

Untuk mengetahui hubungan antara kelembaban udara dan intensitas matahari, dapat dihitung nilai korelasi antara kelembaban udara dengan intensitas matahari.

$$r_{x_2,y} = \frac{S_{x_2,y}}{\sqrt{S_{x_2,x_2} S_{yy}}} \quad (29)$$

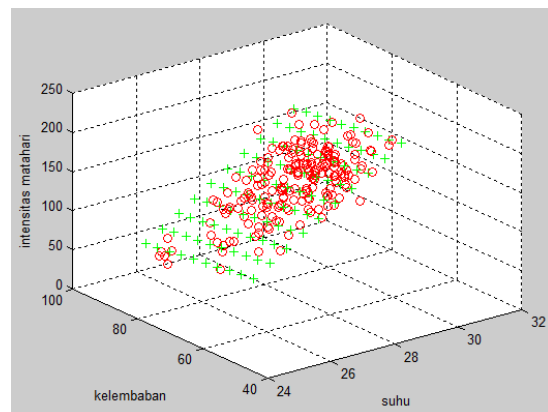
Dengan :

$$S_{x_2,y} = \frac{1}{n} \sum_{i=1}^n (x_{2i} - \bar{x}_2) (y_i - \bar{y})$$

$$S_{x_2,x_2} = \frac{1}{n} \sum_{i=1}^n (x_{2i} - \bar{x}_2)^2$$

$$S_{yy} = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$$

Dari hasil perhitungan, diperoleh korelasi negatif sebesar $r_{x_2,y} = -0,565$ yang artinya semakin tinggi kelembaban udara, intensitas matahari menurun.

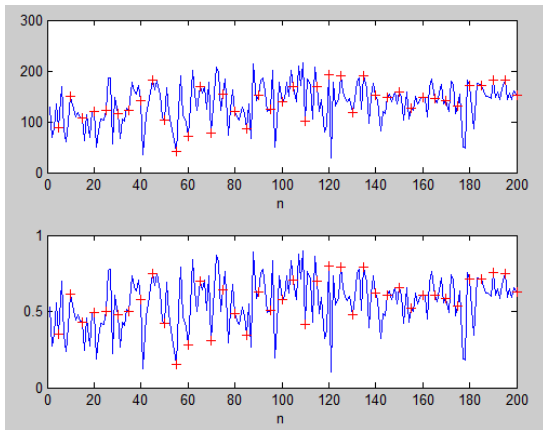


Gambar 4. Pola hubungan gabungan antara suhu udara dan kelembaban udara terhadap intensitas matahari.

Dari hasil korelasi, dapat dibentuk input yang digunakan pada proses prediksi adalah suhu udara (Celcius) dan kelembaban udara (%) sedangkan output yang digunakan adalah intensitas matahari (Watt/m²).

Partisi pada Data

Pada proses prediksi menggunakan Backpropagasi, data cuaca dipartisi menjadi data training sebesar 80% dan data testing sebesar 20%. Dari data cuaca sebesar 200 data, yang akan digunakan sebagai data training adalah data yang bukan kelipatan 5, sedangkan data testing adalah data kelipatan 5 (5,10,15,...,200) sehingga jumlah data testing berjumlah 40. Bentuk partisi data pada output intensitas matahari dapat dilihat Gambar 5.



Gambar 5. Partisi pada data dengan warna biru adalah data training dan warna merah adalah data testing. Atas : Sebelum transformasi. Bawah : Setelah transformasi

Model Jaringan Backpropagasi

Model Backpropagasi yang digunakan dalam penelitian ini terdiri dari dua input (suhu udara x_1 dan kelembaban udara x_2) dan bias. Kemudian terdapat dua hidden layer serta satu output yaitu intensitas matahari y . Hidden layer pertama terdapat empat unit dan satu bias. Hidden layer kedua terdapat tiga input dan satu bias. Karena menggunakan dua hidden layer, maka representasi partikel pada Eq. 18 dimodifikasi menjadi tiga matriks bobot awal. Matriks bobot awal pertama (sub partikel 1) menghubungkan unit input dengan hidden layer pertama, matriks bobot awal kedua (sub partikel 2) menghubungkan hidden layer pertama dengan hidden layer kedua, matriks bobot awal ketiga (sub partikel 3) menghubungkan hidden layer kedua dengan unit output.

Parameter PSO

Algoritma PSO menggabungkan algoritma PSO dan algoritma NN dalam menemukan matriks bobot awal yang optimum. Algoritma NN yang digunakan adalah Backpropagation yang digunakan untuk menghitung nilai MSE sebagai nilai fitness.

Representasi partikel pada PSO adalah pasangan tiga matriks bobot awal, yaitu matriks bobot awal yang menghubungkan input dan hidden layer pertama, matriks bobot awal yang

menghubungkan hidden layer pertama dan hidden layer kedua, matriks bobot awal yang menghubungkan hidden layer kedua dan output.

Parameter yang digunakan pada PSO dapat dilihat pada Tabel 1 dan Tabel 2:

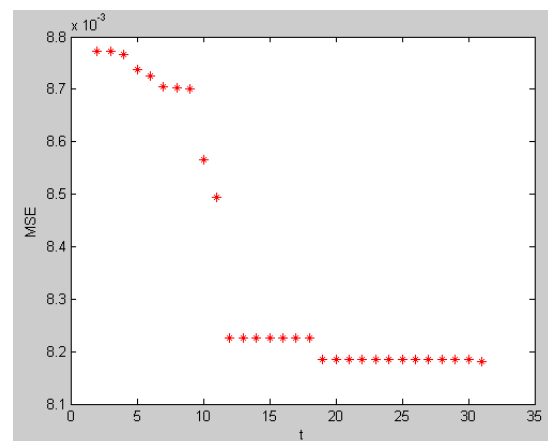
TABEL 1. PARAMETER PSO

Dimensi partikel		
	Sub partikel 1	3×4
	Sub partikel 2	5×3
	Sub partikel 3	4×1
Jumlah partikel dalam swarm		20
Maksimum iterasi		30

TABEL 2. PARAMETER NN DALAM PSO

Proses Training	
Laju pemahaman α	0,2
Jumlah data (80% × 200)	160
Maksimum epoch	1000
Proses Testing	
Laju pemahaman α	0,2
Jumlah data (20% × 200)	40
Maksimum epoch	1

Perhitungan fitness function pada PSO menggunakan algoritma Backpropagation yang melibatkan data training dan data testing. Ketika suatu partikel yang terdiri dari tiga matriks bobot memasuki algoritma Backpropagation, terjadi proses training pada data training dan proses testing pada data testing sehingga menghasilkan nilai MSE sebagai nilai fitness.



Gambar 6. Proses optimisasi pada Particle Swarm Optimization-Neural Network (PSO)

Hasil simulasi pada PSO dapat dilihat pada Gambar 6. Pada iterasi pertama, algoritma PSO menunjukkan nilai MSE yang relatif besar. Pada proses optimisasi, posisi partikel

diupdate berdasarkan kecepatan partikel sehingga nilai MSE menurun sampai iterasi maksimum.

Hasil Prediksi

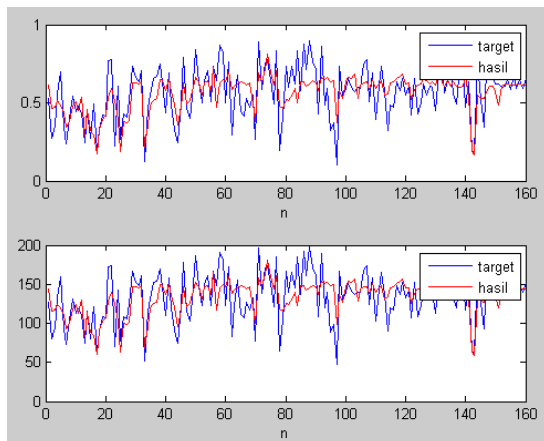
Simulasi dimulai dengan menjalankan simulasi PSONN. Pada akhir iterasi PSONN, diperoleh global best particle sebagai bobot awal yang optimal. Bobot awal optimal tersebut ditraining ulang dan di testing ulang untuk memperbaiki hasil prediksi dengan memperpanjang jumlah epoch menggunakan algoritma Backpropagasi dengan parameter seperti Tabel 3.

TABEL 3. PARAMETER NN PADA PROSES PENGULANGAN

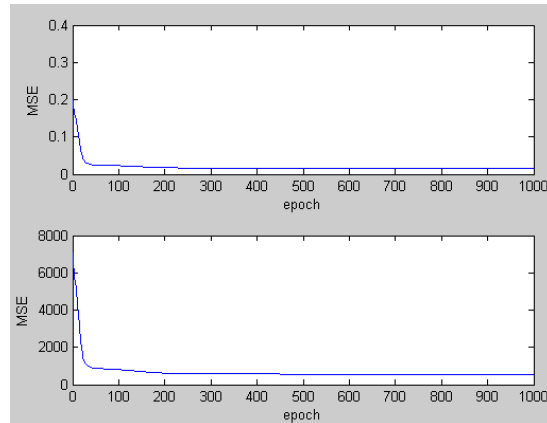
Proses Training	
Laju pemahaman α	0,2
Jumlah data ($80\% \times 200$)	160
Maksimum epoch	1000
Proses Testing	
Laju pemahaman α	0,2
Jumlah data ($20\% \times 200$)	40
Maksimum epoch (extended)	1000

Hasil simulasi terdiri dari hasil simulasi pada data training dan hasil simulasi pada data testing. Data training diambil dari data yang bukan kelipatan 5. Kemudian dilanjutkan proses training sampai 1000 epoch. Hasil simulasi pada data training dapat dilihat pada Gambar 7 dimana dapat dilihat perbandingan antara nilai target (biru) dan nilai prediksi (merah).

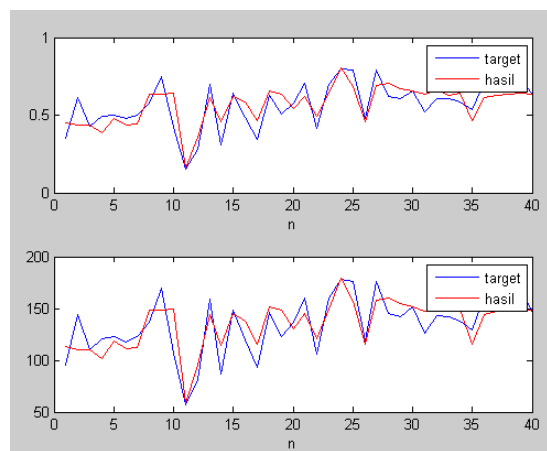
Data testing diambil dari data kelipatan 5. Kemudian dilanjutkan proses training sampai 1000 epoch (extended) untuk memperbaiki hasil prediksi dari PSONN. Gambar 9. menunjukkan perbandingan nilai target (biru) dengan nilai prediksi (merah). Gambar 8 menunjukkan konvergensi nilai MSE pada proses training. Dapat dilihat jika nilai MSE menurun seiring dengan bertambahnya epoch. Gambar 11 menunjukkan nilai MSE pada data testing yang menunjukkan adanya perbaikan dari 0,00818 menjadi 0,00797.



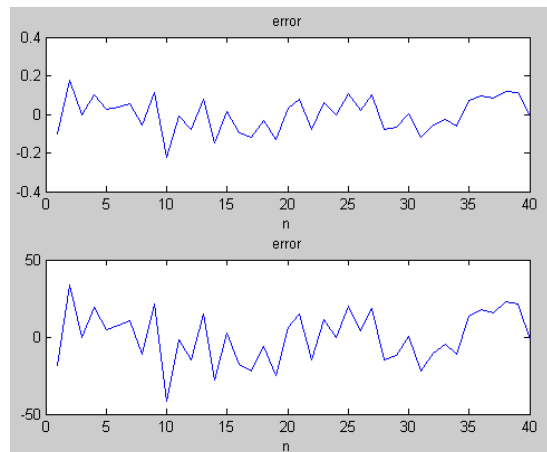
Gambar 7. Simulasi pada data training. Atas : Sebelum detransformasi. Bawah : Setelah detransformasi



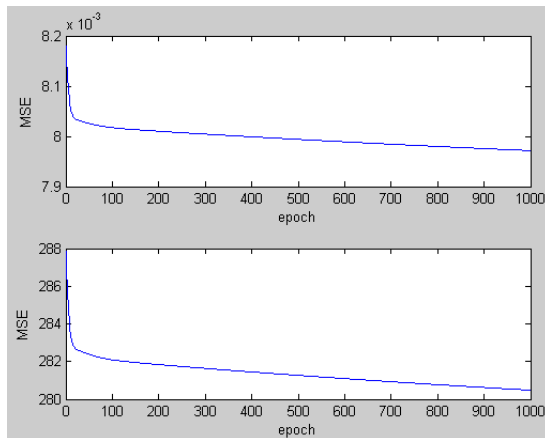
Gambar 8. Nilai Mean Square Error (MSE) pada data training. Atas : Sebelum detransformasi. Bawah : Setelah detransformasi



Gambar 9. Simulasi pada data testing. Atas : Sebelum detransformasi. Bawah : Setelah detransformasi



Gambar 10. Nilai selisih antara target dan hasil. Atas : Sebelum detransformasi. Bawah : Setelah detransformasi



Gambar 11. Nilai Mean Square Error (MSE) pada data testing. Atas : Sebelum detransformasi. Bawah : Setelah detransformasi

V. KESIMPULAN

NN menggunakan Backpropagasi dapat digunakan pada masalah prediksi. Algoritma PSONN adalah algoritma yang menggabungkan algoritma PSO dan algoritma NN dalam menemukan matriks bobot awal yang optimum. Dalam PSONN akan dibentuk populasi dari matriks bobot awal yang menghubungkan input, hidden layer, dan output. Kemudian akan diupdate berdasarkan nilai Mean Square Error (MSE) pada jumlah iterasi yang ditentukan. Hasil simulasi menunjukkan bahwa algoritma PSONN dapat menghasilkan nilai prediksi cuaca yang mendekati nilai target.

DAFTAR PUSTAKA

- Fausett, L., 1994, *Fundamentals of Neural Networks*, Prentice-Hall
- Hidayatullah, N.U., dan Ya'umar, 2014, "*Estimasi*

Radiasi Matahari Perjam pada Permukaan Horizontal dengan Extreme Learning Machine", Jurnal Teknik POMITS, ISSN 2337-3539 Vol. 2 No.2, 2014 p.1-6.

Kennedy, J., dan Eberhart, R.C., 1995, "*Particle Swarm Optimization*", Proc IEEE Int. Conf. Neural Networks, p. 1942-1948

Lu, J., Xue, S., Zhang, X., Zhang, S., dan Lu, W., 2014, "*Neural Fuzzy Inference System-Based Weather Prediction Model and Its Precipitation Prediction Experiment*", Atmosphere, ISSN 2073-4433 Vol. 5, November 2014 p.788-805.

Matarneh, L.A., Sheta, A., Bani-Ahmad, S., Alshaer, J., dan Al-oqily, I., 2014, "*Development of Temperature-based Weather Forecasting Models Using Neural Network and Fuzzy Logic*", International Journal of Multimedia and Obiquitous Engineering, ISSN 1975-0080 Vol. 9 No. 12, 2014 p.343-366.

Rahmalia, D., 2017, "*Particle Swarm Optimization-Genetic Algorithm (PSOGA) on Linear Transportation Problem*", AIP Conference Proceedings, ISBN 978-0-7354-1547-8 Vol. 1867, August 2017 p. (020030)1-12

Rajkumar, R., Albert, A.J, dan Chandrakala, D., 2015, "*Weather Forecasting using Fuzzy Neural Network (FNN) and Hierarchy Particle Swarm Optimization Algorithm (HPSO)*", Indian Journal of Science and Technology, ISSN 0974-5645 Vol. 18, No.2, June 2015 p.1-8.

Rao, S.S., 2009, *Engineering Optimization : Theory and Practice*, John Wiley and Sons, New Jersey.

Siang, J.J., 2009, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*, Penerbit ANDI, Yogyakarta.