OPTIMALISASI *NEURAL NETWORK* BERBASIS *PARTICLE SWARM OPTIMIZATION* UNTUK MEMPREDIKSI LAMA PENYINARAN MATAHARI DALAM MEMENUHI KEBUTUHAN ENERGI

Wahyudin Hasyim¹, Alter Lasarudin²

E-Mail : wahyudin_hasyim@umgo.ac.id¹, alterlasarudin@umgo.ac.id²

^{1,2} Program Studi Sistem Informasi Universitas Muhammadiyah Gorontalo

ABSTRACT - The high electricity load that reaches 325 MegaWatt, this is an important concern for the Gorontalo Province government in the need for electrical energy, it is necessary to predict the length of solar radiation in an area, one of which depends on the length of solar cell energy. Among them by designing prediction models. The prediction method that has the smallest error value is the Neural Network, but there are still weaknesses during training to achieve convergence and overfitting. Then it is necessary to optimize the network weights using Particle Swarm Optimization, which is one of the best methods of optimization. With the use of optimization which is measured through the acquisition of Root Mean Square Error (RMSE). The results of testing the algorithm show that the value of RMSE uses Neural Network 0.131, while the application of optimization with particle swarm optimization results RMSE 0.127. By applying the optimization method, it can reduce the error value

Keywords: Prediction, Sun Exposure Time, Neural Network, Particle Swarm Optimization

ABSTRAK - Tingginya beban listrik yang mencapai 325 MegaWatt, hal ini merupakan perhatian penting bagi pemerintah Provinsi Gorontalo dalam kebutuhan energi listrik, maka perlu memprediksi lama penyinaran matahari pada suatu daerah, Energi sel surya salah satunya bergantung pada lamanya penyinaran cahaya matahari. Diantaranya dengan melakukan perancangan model prediksi. Metode prediksi yang memiliki nilai error terkecil adalah Neural Network, akan tetapi masih adanya kelemahan pada waktu pelatihan untuk mencapai konvergen dan overfitting. Maka perlu dilakukan optimalisasi pada bobot jaringan dengan menggunakan Particle Swarm Optimazition, yang merupakan salah satu metode terbaik dalam optimasi. Dengan penggunaan optimasi yang diukur melalui hasil peroleha Root Mean Square Error (RMSE). Hasil pengujian terhadap algoritma menunjukkan bahwa nilai RMSE menggunakan Neural Network 0,131, sedangkan dengan penerapan optimasi dengan particle swarm optimization hasil RMSE 0,127. Dengan penerapan metode optimasi terserbut dapat mengurangi nilai error

Kata kunci: Prediksi, Lama Penyinaran Matahari, Neural Network, Particle Swarm Optimization

I. PENDAHULUAN

Tingginya beban listrik yang mencapai 325 MegaWatt, hal ini merupakan perhatian penting bagi pemerintah Provinsi Gorontalo dalam kebutuhan energi listrik yang mengalami peningkatan disebabkan semakin meningkatnya jumlah penduduk. Terutama di daerah pedalaman yang notabene masih terkendala oleh permasalahan transportasi maupun sarana prasarana pendukung ketersediaan sumber energi listrik. Di sisi lain ketersediaan sumber energi listrik bagi masyarakat akan mempengaruhi banyak aspek kehidupan, diantaranya persoalan ekonomi dan keberlangsungan hidup masyarakat [1].

Pemecahan solusi terkait permasalahan ini dapat melalui penggunaan listrik terbaharukan seperti pemanfaatan sel surya sebagai sumber energi listrik. Agar pemanfaatan sel surya ini dapat berjalan maksimal, maka perlu memprediksi lama penyinaran matahari di suatu daerah, hal ini disebabkan karena energi surya salah satunya bergantung terhadap lamanya penyinaran cahaya matahari. Suhu permukaan matahari memiliki kurang lebih 6000K memancarkan radiasi sebesar 73,5 MW/m². Energi radiasi yang sampai dipermukaan atmosfer bumi sekitar 1368 Wm⁻². Namun tidak semua energi matahari mencapai permukaan bumi karena sebagian energi radiasi dipantulkan dan diserap oleh atmosfer bumi,

oleh karena itu perlu untuk mengetahui berapa besar energi yang dihasilkan dari lama penyinaran matahari yang diperlukan untuk dijadikan sumber energi listrik [2].

Dalam memenuhi data lama penyinaran matahari yang memadai, diperlukan suatu cara yang efektif dan efisien. Diantaranya dengan melakukan perancangan model prediksi berdasarkan data-data yang di peroleh dari pengamatan stasiun BMKG Provinsi Gorontalo.

Penelitian [3] menunjukkan bahwa model prediksi yang didasarkan pada persamaan Angstrom yang menggunakan data lama penyinaran matahari periode 1994-2003. Hasil dari prediksi dengan persamaan Angstrom yang melakukan uji terhadap nilai MBE (*Mean Bias Error*) -0,01 % , RMSE (*Root Mean Square Error*) 0,03% dan bias error 5%. Tanda negatif pada MBE menunjukkan bahwa model tersebut tergolong *under estimate* terhadap hasil pengukuran, namun masih berada didalam rentang yang dapat diterima. Dalam melakukan perhitungan lama penyinaran matahari yang dilakukan oleh [4] yang menggunakan metode segmentasi dengan *otsu threshold* mampu melakukan segmentasi objek penyinaran berlubang pada kartu bias dengan jumlah objek 1.4429 dengan nilai RMSE = 0,51 terhadap hasil perhitungan lama penyinaran matahari hal ini dikarenakan masih memiliki noise yang berada disekitar region objek.

Penentuan metode prediksi merupakan penentu utama dalam memperoleh model pola terbaik. Penerapan metode yang akan digunakan adalah *Neural Network* berbasis *Particle Swarm Optimization* untuk memprediksi lama penyinaran matahari dalam memenuhi kebutuhan energi kedepan khususnya pada provinsi Gorontalo.

Tujuan utama dalam penelitian ini mendapatkan suatu model prediksi terbaik dengan nilai RMSE terkecil yang mampu memprediksi lama penyinaran matahari pada waktu mendatang. Hal ini adalah sangat penting, karena pada masa yang akan datang memungkinkan untuk penggunaan penghematan energi listrik, khususnya juga bagi para perencana dan perancang peralatan energi surya dapat memprediksi lama penyinaran matahari yang merupakan sumber energi listrik dengan menggunakan model prediksi yang akan diterapkan.

II. LANDASAN TEORI

A. *Neural Network*

Neural Network merupakan salah satu algoritma yang model dari proses pengeolahan informasi seperti kerja sistem saraf manusia, dimana otak merupakan pemroses utama dalam saraf manusia. *Neural Network* mampu mengenali sinyal input yang dikirimkan melalui penghubung neuron yang lain. Apabila ada salah satu *neuron* yang rusak maka akan dilatih. Struktur dari *neuron Neural Network* memiliki fungsi penjumlahan matematis, *activation function*, dan fungsi keluaran (*output*) [5]

Dalam tahapan algoritma *Neural Network* memiliki beberapa langkah yaitu sebagai berikut :

Langkah 1

Langkah awal dari *neural network* yaitu penentuan sigmoid. Nilai bobot terkecil akan melakukan proses *trainng*. Hal ini dikarenakan dengan menentukan nilai *range momentum* dengan *learning rate* 0 sampai dengan 1.[6]

Langkah 2

Pada tahapan ini menentukan nilai dari atribut serta kelas yang akan lakukan untuk proses training. Selanjutnya akan dilakukan perhitungan nilai bobot luaran neuron ke node dalam *Hidden Layer*, dengan menggunakan persamaan :

$$Z_{\text{net}} = f(Wj_o + \sum_{i=1}^n x_i w_i) \quad (1)$$

Selanjutnya akan dilakukan perhitungan terhadap seluruh jaringan dari unit *output* dengan menggunakan bentuk persamaan sebagai berikut :

$$Y_{net} = f(Vj_o + \sum_{i=1}^n xi wi) \quad (2)$$

Nilai dari Fungsi f atau *sigmoid* akan digunakan dari hasil perhitungan dengan memberikan nilai 0 sampai dengan 1. Adapun persamaan dari fungsi f yang digunakan dapat dijabarkan sebagai berikut :[7]

$$y = \frac{1}{1+e^{-x}} \quad (3)$$

Langkah 3

Langkah berikutnya tahapan propagasi mundur. Pada Tahap 3 akan melakukan perbandingan nilai yang didapatkan dari nilai hasil *output* dengan sebenarnya, selisih nilai dari sebenarnya dengan hasil hitungan sebenarnya adalah nilai *error* dari selisih akan diberikan terhadap bobot dalam tiap node, agar model yang akan diperoleh akan menjadi lebih baik dari sebelumnya. Adapun perhitungan nilai bobot menggunakan rumus persamaan sebagai berikut :

$$\delta k = (tk - yk) f'(y_{netk}) = (tk - yk) yk (1 - yk) \quad (4)$$

Nilai dari ΔW_{ij} adalah perubahan dari penambahan nilai *learning rate* dengan δ (delta) kemudian akan selanjutnya akan dikalikan dengan nilai masukan. Apabila diperoleh nilai momentum, maka akan dilakukan perkalian dengan perubahan nilai yang ada sebelumnya. Dengan menggunakan rumus persamaan sebagai berikut:[8]

$$\Delta W_{ij} = \eta \delta k Z_j + \alpha \Delta W (n - 1) \quad (5)$$

Adapun perhitungan nilai δ (delta) dapat dibedakan atas dua yaitu *output layer* dan *hidden layer*. Untuk menghitung bobot bias dapat digunakan rumus persamaan yang sama, akan tetapi tidak dengan menggunakan perhitungan momentum, serta nilai dari bias sehingga akan menjadi seperti pada rumus persamaan dibawah ini:

$$W_{oj.new} = W_{oj.current} + (\eta \delta_j) \quad (6)$$

Langkah 4

Langkah selanjutnya yaitu apabila semua bobot baru sudah diketahui, maka pada langkah 2 akan dijalankan kembali, sampai dengan mendapatkan jumlah iterasi [9]

B. Particle Swarm Optimization

Particle swarm optimization (PSO) diperkenalkan oleh Eberhart dan Kennedy pada tahun 1995 yang merupakan salah satu metode optimasi. *Particle Swarm Optimization* terinspirasi dari perilaku sosial koloni hewan seperti rayap, lebah, ikan atau burung. Perilaku social terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Individu yang berperilaku secara terdistribusi dengan menggunakan kecerdasannya (*intelligence*) sendiri dan juga yang dipengaruhi oleh kelompok [10].

Sebagai contoh jika terdapat seekor burung yang menemukan jalan yang tepat (*optimal*) akan menuju ke sumber makanan, maka sisa dari kawanan kelompok yang lainnya juga akan mengikuti jalan tersebut walaupun lokasi mereka berada yang didalam kawanan tersebut tidak saling berdekatan.[11]

Model dari *Particle swarm optimization* ini dikembangkan berdasarkan model kawanan kelompok tersebut, dimana partikel yang akan mendekati target minimum atau maksimum dari fungsi tujuan, maka kawanan partikel tersebut memberikan informasi secara cepat kepada partikel kawanan lainnya, sehingga kawanan partikel yang lainnya akan mengikuti arah yang menuju ke target atau tujuan dari nilai tersebut akan tetapi tidak menuju secara langsung pada target tersebut.[12]

Dalam inisialisai $Pbest$ dan $Gbest$, pada iterasi yang ke-0 akan di samakan dengan nilai pada posisi awal kawan kelompok atau partikel. Sedangkan $Gbest$ dipilih dari salah satu $Pbest$ dengan yang nilai $fitness$ yang paling tinggi. Untuk menentukan kecepatan $update$ dapat digukan dengan persamaan berikut :[13]

$$V_{k+1} = V_k + C_1.R_1.(Pbest_k - X_k) + C_2.R_2.(Gbest_k - X_k) \quad (7)$$

$$X_{k+1} = X_k + V_{k+1} \quad (8)$$

Tahapan selanjutnya degan menggunakan konsep bobot inersia (*inertia weight*) yang dikembangkan oleh Shi dan Eberhart pada tahun 1998 untuk pengendalian yang lebih baik dalam eksplorasi dan eksploitasi agar dapat menghilangkan kebutuhan Vmax.[14]

III. METODOLOGI PENELITIAN

Alur penelitian yang akan digunakan yaitu sebagai berikut :

IV. HASIL DAN PEMBAHASAN

A. Pengumpulan data

Pengumpulan data dilakukan pada Badan Meteorologi, Klimatologi dan Geofisika (BMKG) Provinsi Gorontalo. Dari data yang diperoleh tercatat bahwa pemantauan yang dilakukan pada BMKG Provinsi Gorontalo diperoleh rekapan data dari tahun 1985 sampai 2018. Data tersebut adalah rata2 lama penyinaran matahari perbulan yang akan digunakan dalam proses pengolahan dan analisis data nanti, data yang diambil memenuhi kriteria dalam pengolahan dengan *Neural Network berbasis Particle Swarm Optimization*.

B. Analisis data

Berdasarkan data pengamatan yang dilakukan pada BMKG Provinsi Gorontalo dari tahun 1985 sampai dengan 2018. Variabel dari data tersebut berjumlah 408 *record* dari Januari 1985 sampai dengan Desember 2018. Dimana, *dataset* 1985 sampai 2017 untuk data training sedangkan data 2018 sebagai data testing. *Output* yang dihasilkan dapat digunakan dalam prediksi untuk tahun selanjutnya.

C. Tahap *Preprocessing dan Normalisasi*

Tahapan pada proses awal dilakukan adalah tahap normalisasi data. Normalisasi ini dilakukan bertujuan untuk mengubah dataset *time series* menjadi angka biner *range* antara 0 sampai 1. Hal tersebut dilakukan agar pengolahan data dapat lebih baik karena menghindari *missing value* atau nilai dominan dari suatu dataset. Berikut persamaan normalisasi yang digunakan:[15]

$$X' = \frac{0.8(X-b)}{(a-b)} + 0.1 \quad (9)$$

Di mana nilai dari ,

x' merupakan hasil normalisasi

x merupakan data awal

a merupakan nilai dari maksimum data awal dan

b merupakan nilai dari minimum data asli

tahapan selanjutnya data akan diubah kedalam bentuk *time series univariate* ke *multivariate*, yang terdiri dari 1 *variabel independent*, 12 *variabel independen* yang memiliki 1 *variabel dependen*. Selanjutnya data ini yang akan digunakan dalam proses *training* data memperoleh pola arsitektur atau model terbaik. Pola terbaik akan ditentukan dengan melihat nilai dari *Root Mean Squared Error* (RMSE) terkecil dari hasil analisis n. Tujuan dari *time series multivariate* yaitu akan memodelkan perubahan terhadap data yang berkorelasi serta peubah tersebut dapat dinotasikan dalam tabel pola data *time series* berikut.

Tabel 1. Pola perubahan data *univariate* ke *multivariate*

<i>Pola</i>	<i>Input</i>	<i>Output</i>
1	$X_{t-1}, x_{t-2}, x_{t-3}, x_{t-5}, x_{t-6} x_t$	x_{t+1}
2	$X_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, \dots x_{t+1}$	x_{t+2}
3	$X_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, \dots x_{t+2}$	x_{t+3}
4	...	

5	...	...
6		
...		
$z-t$	$xz-t, xz-t+1, \dots, xz-1$	xz

D. Pengujian dan Evaluasi Algoritma *Neural Network*

Tahapan berikut adalah dari data yang telah dilakukan normalisasi dan *preprocessing*, data tersebut selanjut akan diproses dengan menerapkan algoritma *Neural Network* untuk memperoleh pola arsitektur baik. Pada proses penerapan algoritma ini menggunakan *tolls rapidminer* dengan mengukur hasil nilai dari RMSE dengan menentukan parameter nilai pada *Neural Network*. Adapun untuk penentuan nilai dari parameter pada metode *Neural Network* yaitu sebagai berikut :

1. *Input Layer*

Dalam penentuan parameter nilai *input layer* pada percobaan ini yaitu menerapkan jumlah masukan dimulai dari 1 sampai 12 periode, akan tetapi tidak melakukan perubahan pada parameter nilai yang berada pada *number of validation* yaitu 10, *Training Cycle* 500, *Learning Rate* 0,3, dan *Momentum* 0,2 dan serta nilai dari *neuron size* yang berada pada *hidden layer* juga tidak dilakukan perubahan. Adapun hasil dari uji coba penentuan jumlah *input layer* didapatkan hasil pengujian yang diukur dengan *Root Mean Square Error* terendah diperoleh pada *input layer* $xt-4$ dengan perolehan nilai RMSE 0,131. Jumlah *input layer* yang akan diterapkan dalam uji coba penentuan parameter untuk mendapatkan pola data yang paling terbaik yang terdiri dari 4 periode masing-masing yaitu variable $xt-1, xt-2, xt-3, xt-4$.

2. *Hidden layer*

Penentuan *Hidden layer* yang akan digunakan yaitu 1 dengan menentukan jumlah *neuron size*. Jumlah *neuron size* yang diterapkan dalam pengujian ini adalah nilai *neuron size* mulai dari 2 sampai 20 dengan nilai yang ada pada *number of validation* 10, *Training Cycle* 500, *Learning Rate* 0,3, dan *Momentum* 0,2. Berdasarkan hasil percobaan yang diterapkan pada 1 *hidden layer* yang memperoleh nilai RMSE terendah terdapat pada *neuron size* 17 dengan nilai sebesar 0,134. Hasil pemelihan berdasarkan nilai RMSE terkecil merupakan hasil perolehan yang dibandingkan dengan jumlah neuron lainnya

3. *Training Cyle*

Pada tahapan ini berdasarkan hasil yang diperoleh dari percobaan pada *hidden layer* dengan jumlah *neuron size* 17 dilakukan uji coba untuk penentuan *Training Cyle* dengan nilai *range* 100 dengan kelipatan 100 sampai dengan 1000. Berdasarkan hasil uji coba yang dilakukan bahwa yang memperoleh nilai RMSE terkecil pada *Training Cyle* dengan nilai 100 yang mendapatkan RMSE sebesar 0,133. Dari hasil terbaik pada *Training Cyle* yang diperoleh akan diterapkan dalam menentukan nilai yang ada pada parameter *learning rate*.

4. *Learning Rate*

Dalam penentuan nilai parameter *learning rate*, dengan menerapkan parameter nilai *learning rate* pada *range* nilai 0,1 sampai 1, dengan menggunakan nilai *neuron size* dan *training cycle* yang telah diperoleh pada percobaan sebelumnya. Berdasarkan hasil uji coba yang dilakukan pada tahapan ini yang memiliki nilai *Root Mean Square Error* (RMSE) terkecil pada *learning rate* adalah nilai 0,1 yang memperoleh nilai RMSE sebesar 0,132. Hasil terbaik pada *learning rate* yang diperoleh akan diterapkan pada tahapan selanjutnya untuk menentukan nilai yang ada pada parameter *momentum*

5. Momentum

Dalam penentuan parameter nilai *momentum* akan dilakukan dengan menerapkan parameter nilai *momentum* dari *range* nilai 0.1 sampai 0.9, dengan menggunakan nilai *neuron size*, *training cycle*, dan *learning rate* yang telah diperoleh pada percobaan sebelumnya. Berdasarkan hasil uji coba yang dilakukan bahwa yang memiliki nilai *Root Mean Square Error* (RMSE) terkecil pada *momentum* dengan nilai 0,1 yang memperoleh nilai RMSE sebesar 0,131.

Dengan demikian setelah dilakukan tahap percobaan dalam penentuan nilai setiap parameter pada *Neural Network* yang sudah dilakukan tahapan proses sebelumnya nilai dari parameter yang didapatkan dalam model terbaik akan digunakan dalam menentukan prediksi dengan menerapkan setiap nilai parameter pada arsitektur *neural network* adalah *input layer 4*, *hidden layer 1 neuron size 17*, *training cycle 100*, *learning rate 0.1* dan *momentum 0.1*, yang nanti nilai yang didapatkan dari parameter inilah yang akan digunakan untuk melakukan prediksi yang diperoleh dari arsitektur terbaik pada metode *Neural Network* yang diukur pada nilai RMSE sebesar 0,131.

Tabel 2. Pola penerapan parameter nilai terbaik dari *Neural Network*

<i>Input Layer</i>	4
<i>Hidden Layer</i>	1
<i>Neuron Size</i>	17
<i>Training Cycle</i>	100
<i>Learnig Rate</i>	0,1
<i>Momentum</i>	0.1
<i>RMSE</i>	0.131 +/- 0.011 (mikro: 0.131 +/- 0.000)

E. Tahapan Pengujian Optimasi pada *Neural Network* menggunakan *Particle Swarm Optimization*

Pada tahapan ini merupakan proses pencarian model optimasi dengan *Particle Swarm Optimization* dengan menerapkan hasil yang sudah diperoleh dari model parameter nilai terbaik *Neural Network* sebelumnya.

Adapun untuk pencarian solusi optimal maka akan dilakukan sampai semua partikel mendapatkan solusi yang sama atau pada saat iterasi maksimum sudah tercapai. Untuk formulasi *update* kecepatan dan posisi pada iterasi dimisalkan akan melakukan optimasi terhadap variabel x_{t-1} yang digunakan pada data testing adapun tahapan algoritma *Particle Swarm Optimization* adalah sebagai berikut :

Dalam penentuan parameter pada *population size* maka dilakukan uji coba dengan menerapkan nilai *population size* dari *range* 5 sampai dengan 15 akan tetapi dengan tidak melakukan perubahan terhadap parameter nilai yang berada pada *maximum number of generation* 30, serta nilai *local best weight*, nilai *g-best weight* yaitu 1.0, sedangkan parameter pembobotan nilai *inertia weight* adalah 1.0, nilai *max weight* adalah 1.0, untuk penentuan nilai dari *min weight* 0.0. Berdasarkan hasil uji coba yang dilakukan bahwa yang memiliki nilai *Root Mean Square Error* (RMSE) terkecil pada *population size* dengan nilai 9 yang memperoleh nilai RMSE sebesar 0,127. Selanjutnya hasil terbaik pada *population size* yang diperoleh akan digunakan untuk melakukan penentuan nilai dari *maximum number of generation*.

Pada tahapan selanjutnya dalam menentukan nilai *maximum number of generation* pada percobaan ini dilakukan dengan menerapkan nilai *range* nilai 10, 20, 30 sampai dengan 100, dengan menggunakan nilai *population size* yang telah diperoleh pada percobaan sebelumnya, akan tetapi dengan tidak melakukan perubahan terhadap parameter nilai *local best weight* dan nilai *g-best weight* yaitu nilai 1.0, sedangkan pada parameter pembobotan nilai parameter *inertia weight* dan nilai *max weight* masing-masing 1.0, nilai *min weight* 0.0 . Berdasarkan hasil uji coba yang dilakukan dengan menerapkan nilai yang ditentukan bahwa yang memiliki nilai 30 merupakan *Root Mean Square Error* (RMSE) terkecil pada *maximum number of generation* dengan perolehan nilai RMSE sebesar 0,127.

F. Evaluasi *Neural Network* yang di optimasi menggunakan *Particle Swarm Optimization*

Berdasarkan hasil pengujian yang dilakukan sebelumnya diperoleh optimalisasi *Neural Network* dengan *Particle Swarm Optimization* pada parameter yang berada pada *Particle Swarm Optimization*, proses optimasi yang dilakukan terhadap parameter-parameter yang ada mampu menghasilkan nilai yang berpengaruh dalam proses penentuan posisi terbaik dalam melakukan prediksi.

Hal ini dibuktikan dengan adanya penurunan nilai *Root Mean Square Error* (RMSE) pada uji coba yang dilakukan, hal ini dikarenakan *Particle Swarm Optimization* merupakan salah satu algoritma optimasi yang mengadaptasi perilaku sekumpulan hewan bergerak untuk mencapai suatu tujuan dengan melakukan pencarian pada posisi yang terbaik.

Particle Swarm Optimization merupakan metode optimasi yang pada dasarnya adalah *swarm intelligence*, dimana penentuan algoritma ini diawali dengan populasi yang bersifat random atau acak yang disebut juga dengan *Particle*. Dimana setiap *Particle* yang dimiliki mempunyai karakteristik tersendiri yaitu posisi dan kecepatan. *Particle Swarm Optimization* ini juga melakukan inisialisasi bobot nilai yang diproses dari *Neural Network* yang kemudian akan mengoptimasi bobot yang dilakukan pada proses sehingga akan menghasilkan model arsitektur terbaik.

Dari hasil pengujian yang diterapkan pada metode *Neural Network* yang dioptimasi dengan *Particle Swarm Optimization* yang mendapatkan pola terbaik adalah parameter yang diperoleh pada *Neural Network* yaitu *input layer 4*, *hidden layer 1 neuron size 17*, *training cycle 100*, *learning rate 0.1* dan *momentum 0.1* dan *momentum 0.1* serta pada *Particle Swarm Optimization* dengan memasukkan nilai *population size 9*, *maximum number of generation 30*, dan nilai *local best weight* dan *global best weight* adalah masing-masing 1.0, sedangkan pada parameter nilai *inertia weight* 1.0, nilai *max weight* 1.0, dan nilai *min weight* adalah 0.0. Dengan menerapkan pola data tersebut dalam pengujian mampu memperoleh nilai RMSE pada metode *Neural Network* yaitu 0,131 dan hasil penerapan metode *Neural Network* yang dilakukan optimasi dengan *Particle Swarm Optimization* mampu memperoleh nilai RMSE yang lebih rendah sebesar 0,127.

Sehingga dengan menerapkan model optimasi pada metode *Neural Network* dengan menggunakan *Particle Swarm Optimization* hal ini mampu membuktikan bahwa hasil nilai yang diperoleh dari hasil pengujian jauh lebih baik jika dibandingkan dengan menggunakan metode *Neural Network*.

Tabel 3. Perbandingan Hasil RMSE

Metode	Hasil RMSE
<i>Neural Network</i>	0,131
<i>Neural Network yang dilakukan optimasi dengan Particle Swarm Optimization</i>	0,127

Gambar 2. Hasil Perbandingan nilai RMSE

Dengan demikian hasil penerapan parameter nilai yang diperoleh pada metode *Neural Network* yang dilakukan optimasi dengan *Particle Swarm Optimization* dengan pola terbaik ini akan dilakukan *testing* dengan menggunakan dataset training pada tahun 1985 sampai 2017 untuk mendapatkan hasil prediksi tahun 2018 yang dimulai dari bulan januari sampai desember. Hasil prediksi yang didapatkan akan dibanding dengan nilai aktual dari dataset tahun 2018.

Tabel 4. Perbandingan data aktual dan hasil peramalan

Bulan	Data Aktual 2018	Hasil Peramalan	Error
Januari	0.2091	0.3170	0.1079
Februari	0.1800	0.2788	0.0988
Maret	0.2527	0.2618	0.0090
April	0.1927	0.2618	0.0690
Mei	0.2102	0.2372	0.0270
Juni	0.2655	0.2549	-0.0106
Juli	0.3618	0.3078	-0.0540
Agustus	0.4236	0.3111	-0.1125
September	0.4382	0.2866	-0.1516
Oktober	0.3000	0.2884	-0.0116
November	0.2818	0.3255	0.0437
Desember	0.2745	0.3245	0.0500

Tabel 5. Data aktual 2018 dalam denormalisasi

Bulan	Data Aktual 2018	Hasil Peramalan	Error
Januari	137.0	166.7	29.7
Februari	129.0	156.2	27.2
Maret	149.0	151.5	2.5
April	132.5	151.5	19.0
Mei	137.3	144.7	7.4
Juni	152.5	149.6	-2.9
Juli	179.0	164.1	-14.9
Agustus	196.0	165.1	-30.9
September	200.0	158.3	-41.7
Oktober	162.0	158.8	-3.2
November	157.0	169.0	12.0
Desember	155.0	168.7	13.7

Gambar 3. Hasil Perbandingan Data Aktual dan Prediksi

G. Hasil Prediksi

Dengan menerapkan hasil nilai parameter terbaik yang telah diperoleh dari metode *Neural Network* yang dilakukan optimasi dengan *Particle Swarm Optimization* akan diterapkan untuk mendapatkan hasil prediksi tahun 2019 mulai dari Januari sampai Desember. Dengan menggunakan dataset *training* dan *testing* tahun 1985 sampai 2018 yang akan di jadikan data *training* tahun 2019 yang mampu mendapatkan *output* atau berupa hasil prediksi.

Tabel 6. Hasil Prediksi Lama Penyinaran Matahari Tahun 2019

Prediksi 2019		
Bulan	Normalisasi	Denormalisasi
Januari	0.3616	178.9
Februari	0.3637	179.5
Maret	0.4117	192.7
April	0.3967	188.6
Mei	0.3642	179.7
Juni	0.3262	169.2
Juli	0.2968	161.1
Agustus	0.2877	158.6
September	0.3145	166.0
Oktober	0.2887	158.9
November	0.2982	161.5
Desember	0.2767	155.6

Gambar 4. Hasil Prediksi Lama Penyinaran Matahari Tahun 2019

V. PENUTUP

A. KESIMPULAN

Berdasarkan hasil ujicoba dan analisa yang telah dilakukan bahwa *Neural Network* dengan *Particle Swarm Optimization* mampu mendapatkan pola terbaik yaitu parameter yang diperoleh dari metode *Neural Network* dengan input layer 4 , hidden layer 1 neuron size 17, training cycle 100, learning rate 0.1 dan momentum 0.1 dan momentum 0.1 serta pada *Particle Swarm Optimization* dengan memasukkan nilai population size 9, maximum number of generation yaitu 30, dan nilai local best weight dan global best weight adalah 1.0, sedangkan pada parameter nilai inertia weight adalah 1.0, nilai max weight adalah 1.0, dan nilai min weight adalah 0.0. Mampu memperoleh hasil nilai RMSE) dengan menerapkan pola pada metode *Neural Network* sebesar 0,131 dan hasil pengujian terhadap metode

Neural Network yang dioptimasi dengan Particle Swarm Optimization mampu memperoleh nilai RMSE yang terendah sebesar 0,127. Sehingga dengan melakukan optimasi pada metode *Neural Network* yang menggunakan *Particle Swarm Optimization* terbukti mampu menghasilkan nilai yang lebih baik jika dibandingkan dengan menggunakan metode *Neural Network*. Dengan penerapan pola yang diperoleh dari metode optimasi tersebut mampu mendapatkan nilai error terendah sehingga dapat digunakan dalam memprediksi lama penyinaran matahari.

B. SARAN

Adapun saran pada peneliti selanjutnya yaitu dengan melakukan penerapan pola dan metode *Neural Network* yang dioptimasi oleh *Particle Swarm Optimization* pada dataset lainnya. Dana dapat juga dilakukan uji coba dengan menggunakan algoritma optimasi lainnya

REFERENSI

- [1] N. Arumsari and F. A. Pamuji, "Peramalan Irradiance Cahaya Matahari Pada Sel Surya Untuk Memenuhi Kebutuhan Energi Listrik Dengan Metode Support Vector Regression (Svr)," *J. Nas. Tek. Elektro*, vol. Vol : 6, N, no. 1, 2017.
- [2] R. Pratama and Z. Kamus, "Pembuatan Sistem Pengukuran Durasi Penyinaran Matahari Berbasis Mikrokontroler Atmega8535 Menggunakan Sensor Ldr," *J. pillar physics, Jur. Fis. Univ. Negeri Padang*, vol. 2, pp. 99–106, 2013.
- [3] Y. S. Utomo, "Prediksi radiasi surya global bulanan kota bandung menggunakan data lpm (lama penyinaran matahari)," *J. Mater. dan Energi Indones. FMIPA Univ. Padjadjaran*, vol. 07, no. 02, pp. 21–27, 2017.
- [4] A. Pujiastuti, "Sistem Perhitungan Lama Penyinaran Matahari (Studi Kasus : ST . KLIMATOLOGI BARONGAN)," *ugm, FMIPA*, vol. Volume 5, 2016.
- [5] E. Noersasongko, F. T. Julfia, A. Syukur, , P., R. A. Pramunendar, and C. Supriyanto, "A Tourism Arrival Forecasting using Genetic Algorithm based Neural Network," *Indian J. Sci. Technol.*, vol. 9, no. 4, pp. 3–7, 2016.
- [6] S. Sma, P. Prestasi, S. Sman, Y. A. Lesnussa, S. Latuconsina, and E. R. Persulesy, "Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Prestasi Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Prestasi Siswa SMA (Studi kasus : Prediksi Prestasi Siswa SMAN 4 Ambon)," *J. Mat. Integr.*, vol. 11, no. 2 Oktober, pp. 149–160, 2015.
- [7] D. Rahmalia and T. Herlambang, "Prediksi Cuaca Menggunakan Algoritma Particle Swarm Optimization-Neural Network (Psonn)," pp. 41–48, 2017.
- [8] Y. D. Lestari, "Jaringan syaraf tiruan untuk prediksi penjualan jamur menggunakan algoritma backpropagation," *J. ISD*, vol. 2, no. 1, pp. 40–46, 2017.
- [9] B. Di, P. Kota, S. Tiruan, B. Di, and P. Kota, "Prediksi Distribusi Air PDAM Menggunakan Metode Jaringan Syaraf Tiruan," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 2 January, pp. 83–87, 2016.
- [10] I. Cholissodin, S. Sutrisno, A. A. Soebroto, L. Hanum, and C. A. Caesar, "Optimasi Kandungan Gizi Susu Kambing Peranakan Etawa (PE) Menggunakan ELM-PSO Di UPT Pembibitan Ternak Dan Hijauan Makanan Ternak Singosari-Malang," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 1, p. 31, 2017.
- [11] R. F. Margeritha, R. S. Hartati, and N. P. Satriya Utama, "AnalisisPenyambungan Distributed Generation Guna Meminimalkan Rugi-Rugi Daya Menggunakan Metode Particle Swarm Optimization (PSO)," *Maj. Ilm. Teknol. Elektro*, vol. 16, no. 3, p. 122, 2017.
- [12] C. A. Caesar, L. Hanum, and I. Cholissodin, "Perbandingan Metode ANN-PSO Dan ANN-GA Dalam Pemodelan Komposisi Pakan Kambing Peranakan Etawa (PE) Untuk Optimasi Kandungan Gizi," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 3, p. 216, 2016.
- [13] H. Muhamad, C. A. Prasajo, N. A. Sugianto, L. Surtiningsih, and I. Cholissodin, "Optimasi Naïve Bayes Classifier Dengan Menggunakan Particle Swarm Optimization Pada Data Iris," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 3, p. 180, 2017.
- [14] Irvan Muzakkir, A. Syukur, and I. N. Dewi, "Backpropagation Dengan Seleksi Fitur Particle Swarm Optimization Dalam Prediksi Pelanggan Telekomunikasi," *Pseudocode*, vol. 1, pp. 1–10, 2014.
- [15] A. Wanto, A. P. Windarto, D. Hartama, and I. Parlina, "Use of Binary Sigmoid Function And Linear Identity In Artificial Neural Networks For Forecasting Population Density," *IJISTECH (International J. Inf. Syst. Technol.*, vol. 1, no. 1, p. 43, 2017.