

PERANCANGAN ALGORITMA BELAJAR JARINGAN SYARAF TIRUAN MENGGUNAKAN PARTICLE SWARM OPTIMIZATION (PSO)

Nurmahaludin⁽¹⁾

⁽¹⁾ Staf Pengajar Jurusan Teknik Elektro Politeknik Negeri Banjarmasin

Ringkasan

Jaringan syaraf tiruan menggunakan konsep belajar untuk menyelesaikan permasalahan. Dalam proses belajar tersebut, jaringan akan mengatur bobot-bobotnya agar mampu mengenali pola hubungan antara input dan output yang diberikan. Modifikasi bobot-bobot jaringan dilakukan dengan menggunakan suatu algoritma yang disebut dengan algoritma belajar. Pada makalah ini digunakan algoritma Particle Swarm Optimization (PSO) sebagai algoritma belajar jaringan syaraf tiruan. Jaringan kemudian diujikan pada aplikasi prakiraan temperatur udara harian menggunakan model time series. Hasil pengujian menunjukkan bahwa penggunaan algoritma belajar PSO memberikan hasil yang cukup baik dengan rata-rata kesalahan prakiraan 2.41% pada prakiraan temperatur udara minimum dan 3.81% pada prakiraan temperatur udara maksimum. Jika dibandingkan dengan algoritma Backpropagation, error pada proses pelatihan dan rata-rata error pada proses pengujian yang dihasilkan tiap iterasi relatif lebih baik, walaupun waktu komputasi yang diperlukan lebih lama.

Kata Kunci : Jaringan Syaraf Tiruan, Algoritma Belajar, Particle Swarm Optimization

1. PENDAHULUAN

Latar Belakang

Jaringan syaraf tiruan pada dasarnya dapat dianggap sebagai sebuah model yang akan memetakan data input terhadap output. Proses komputasi dalam jaringan syaraf tiruan dilakukan untuk mencari hubungan matematis antara input dan output melalui mekanisme pengaturan bobot. Proses tersebut berlangsung selama fase pelatihan menggunakan algoritma tertentu, yang disebut dengan algoritma belajar. Jika error yang dihasilkan pada proses pelatihan masih besar, berarti jaringan belum mampu mengenali suatu pola hubungan antara input-output sehingga perlu dilakukan adanya *update* bobot.

Dari uraian tersebut dapat disimpulkan bahwa permasalahan utama dalam jaringan syaraf tiruan adalah bagaimana menemukan bobot interkoneksi yang tepat.

Penggunaan algoritma optimasi sebagai algoritma belajar menjadi pilihan karena algoritma tersebut mampu menemukan bobot optimal melalui pencarian secara iteratif. Salah satu algoritma optimasi global yang berkembang saat ini adalah Particle Swarm Optimization (PSO). Pada makalah ini PSO digunakan dan diujikan sebagai algoritma belajar pada jaringan syaraf tiruan. Hasil pengujian yang dilakukan kemudian dibandingkan dengan algoritma Backpropagation.

Tujuan

Penelitian ini bertujuan menganalisis penggunaan algoritma Particle Swarm Optimization guna mendapatkan bobot optimal jaringan syaraf tiruan pada aplikasi prakiraan temperatur udara.

2. LANDASAN TEORI

Tinjauan Pustaka

Sejumlah penelitian telah dilakukan dalam upaya memperbaiki algoritma belajar jaringan syaraf tiruan. Terdapat beberapa algoritma yang diusulkan untuk mempercepat proses pelatihan, diantaranya adalah Fahlman (1988) yang memperkenalkan algoritma Quickprop sebagai hasil modifikasi Backpropagation.

Pada bagian lain, perkembangan algoritma optimasi berlangsung dengan sangat pesat dan ditandai dengan munculnya teknik kecerdasan baru yaitu *swarm intelligence*. Particle Swarm Optimization sebagai bagian dari teknik tersebut telah digunakan dalam banyak kasus optimasi, seperti dalam penjadwalan mesin (Yan, 2009) dan perancangan jalur rangkaian VLSI (Ayob, 2010). Sehingga menjadi hal yang menarik untuk memanfaatkan Particle Swarm Optimization dalam kasus optimasi bobot jaringan.

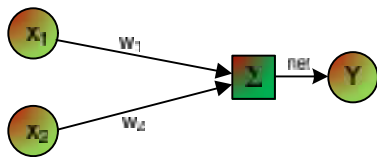
Jaringan Syaraf tiruan

Jaringan syaraf tiruan merupakan representasi buatan dari otak manusia yang selalu men-

coba untuk mensimulasikan proses pembelajaran. Jaringan syaraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi dengan asumsi (Siang, 2005):

- a. Pemrosesan informasi terjadi pada sejumlah elemen sederhana yang disebut dengan neuron
- b. Sinyal dikirimkan di antara neuron tersebut melalui suatu interkoneksi
- c. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal
- d. Untuk menentukan output, tiap neuron menggunakan fungsi aktivasi yang dikenakan pada jumlah input terbobot yang diterima.

Gambar 1. menunjukkan proses komputasi yang terjadi pada jaringan syaraf tiruan.



Gambar 1. Pola Dasar Jaringan Syaraf Tiruan

Output Y menerima input dari neuron x_1 dan x_2 dengan bobot hubungan masing-masing adalah w_1 dan w_2 . Kedua neuron yang ada dijumlahkan menurut Persamaan 2.1.

$$\text{Net} = x_1w_1 + x_2w_2 \quad (2.1)$$

Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $Y = f(\text{net})$. Jika nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan.

Particle Swarm Optimization

Algoritma dari Particle Swarm Optimization (PSO) dikembangkan oleh Dr. Eberhart dan Dr. Kennedy pada tahun 1995. Menurut Kennedy dan Eberhart (2001), terdapat tiga prinsip dasar dalam PSO sebagai sebuah algoritma yang didasarkan pada aspek perilaku sosial kognitif yaitu: evaluasi (*evaluate*), membandingkan (*compare*), dan meniru/mengikuti (*imitate*).

Proses pembelajaran tidak mungkin terjadi kecuali tiap individu melakukan evaluasi kinerja, kemudian membandingkan dengan kelompoknya berdasarkan kriteria yang telah ditentukan. Proses berikutnya adalah belajar dan mengikuti individu yang memiliki nilai terbaik dalam kelompoknya sehingga pengetahuan dioptimalkan melalui interaksi sosial dan bukan hanya bersifat pribadi. Berdasarkan pendekatan dalam pemecahan masalah optimisasi tersebut, PSO digolongkan ke dalam teknik *swarm intelligence*.

Dalam persoalan optimisasi, langkah pertama yang dilakukan dalam pencarian nilai optimal adalah menentukan nilai awal. Nilai awal tersebut diperbaharui secara iteratif menggunakan algoritma tertentu sampai ditemukan solusi yang dianggap memenuhi kriteria. Dalam PSO,

nilai awal tersebut dinyatakan sebagai posisi awal dari partikel. Partikel sendiri merupakan kandidat solusi yang dicari, yang akan terus memodifikasi (memperbaharui) posisinya. Kumpulan kandidat solusi (partikel) ini disebut dengan *swarm*.

Algoritma Particle Swarm Optimization

Algoritma PSO dimulai dengan membangkitkan sejumlah nilai awal secara acak. Jika $\bar{x}_i(t)$ menyatakan posisi dari partikel P_i pada langkah waktu t , maka posisi partikel P_i tersebut akan berubah dengan menambahkan kecepatan $\bar{v}_i(t)$ ke dalam posisi sebelumnya.

Algoritma PSO untuk partikel dengan topologi star dapat dijelaskan sebagai berikut (Engelbrecht, 2002):

- 1) Inisialisasi *swarm*, $P(t)$, dari partikel sedemikian hingga posisi $\bar{x}_i(t)$ dari masing-masing partikel $P_i \in P(t)$ adalah acak
- 2) Hitung performansi (*fitness*) F dari masing-masing partikel pada posisinya saat ini yaitu $F(\bar{x}_i(t))$
- 3) Bandingkan performansi tiap partikel saat ini terhadap performansi terbaik sebelumnya, yaitu $pbest$ (*personal best*).

Jika $F(\bar{x}_i(t)) < pbest_i$ maka:

$$a) pbest_i = F(\bar{x}_i(t)) \quad (2.2)$$

$$b) \bar{x}_{pbest_i} = \bar{x}_i(t) \quad (2.3)$$

- 4) Bandingkan performansi tiap partikel saat ini terhadap performansi terbaik dari seluruh partikel dalam kelompoknya, yaitu $gbest$ (*global best*).

Jika $F(\bar{x}_i(t)) < gbest_i$ maka:

$$a) gbest_i = F(\bar{x}_i(t)) \quad (2.4)$$

$$b) \bar{x}_{gbest_i} = \bar{x}_i(t) \quad (2.5)$$

- 5) Hitung vektor kecepatan tiap partikel:

$$\bar{v}_i(t) = \bar{v}_i(t-1) + \rho_1 (\bar{x}_{pbest_i} - \bar{x}_i(t-1)) + \rho_2 (\bar{x}_{gbest_i} - \bar{x}_i(t-1)) \quad (2.6)$$

Nilai ρ_1 dan ρ_2 didefinisikan sebagai :

$\rho_1 = r_1c_1$ dan $\rho_2 = r_2c_2$, dimana c_1 dan c_2 disebut konstanta akselerasi, sedangkan r adalah suatu bilangan acak

- 6) Perbaharui posisi tiap partikel

$$a) \bar{x}_i(t) = \bar{x}_i(t-1) + \bar{v}_i(t) \quad (2.7)$$

$$b) t = t + 1 \quad (2.8)$$

- 7) Kembali ke langkah 2 hingga tercapai kriteria penghentian

Algoritma Belajar PSO

Proses pelatihan jaringan syaraf tiruan menggunakan algoritma PSO sebagai berikut:

- a. Menentukan jumlah partikel yang digunakan. Panjang partikel menyatakan jumlah bobot-bobot jaringan yang akan dilatih hingga dicapai bobot akhir yang optimum. Jika jaringan mempunyai satu lapisan tersembunyi, maka panjang partikel diberikan oleh: Panjang partikel = (jml neuron input + 1) x jml neuron hidden layer + (jml neuron hidden layer + 1) x jml neuron output
- b. Inisialisasi bobot interkoneksi awal secara acak
- c. Pada tiap iterasi, hitung output jaringan dengan merambatkan sinyal dari input menuju lapisan tersembunyi dan kemudian ke lapisan output dengan menggunakan fungsi aktivasi yang ditentukan.
- d. Menghitung error yang terjadi, yaitu selisih antara target dengan output jaringan. Error ini merupakan nilai *fitness* yang akan dievaluasi pada tiap partikel
- e. Membandingkan dan menentukan nilai dari pbest dan gbest. Pbest adalah nilai error terkecil yang dimiliki suatu partikel, sedangkan gbest adalah nilai error terkecil dari seluruh partikel. Partikel yang memiliki nilai gbest berarti partikel yang bobot-bobotnya menghasilkan error paling kecil
- f. Modifikasi bobot jaringan menurut Persamaan 2.6 dan Persamaan 2.7
- g. Ulangi langkah (c) hingga tercapai kriteria penghentian
- h. Bobot akhir ditentukan dari bobot yang partikelnya mempunyai error terkecil.

3. METODE PENELITIAN

Diagram alir metodologi penelitian ditunjukkan dalam Gambar 2. Sebelum melakukan penelitian, dilakukan pengumpulan dan pengolahan data. Data yang diperlukan adalah data temperatur udara harian minimum dan maksimum. Data kemudian dibagi menjadi data pelatihan dan data pengujian. Pembuatan program jaringan syaraf tiruan terdiri atas pelatihan dengan Backpropagation dan PSO. Setelah program berhasil dijalankan, dilakukan analisis terhadap kedua metode tersebut.

4. HASIL DAN PEMBAHASAN

Parameter Algoritma

Beberapa parameter yang mempengaruhi kinerja algoritma PSO dan Backpropagation sebagai berikut.

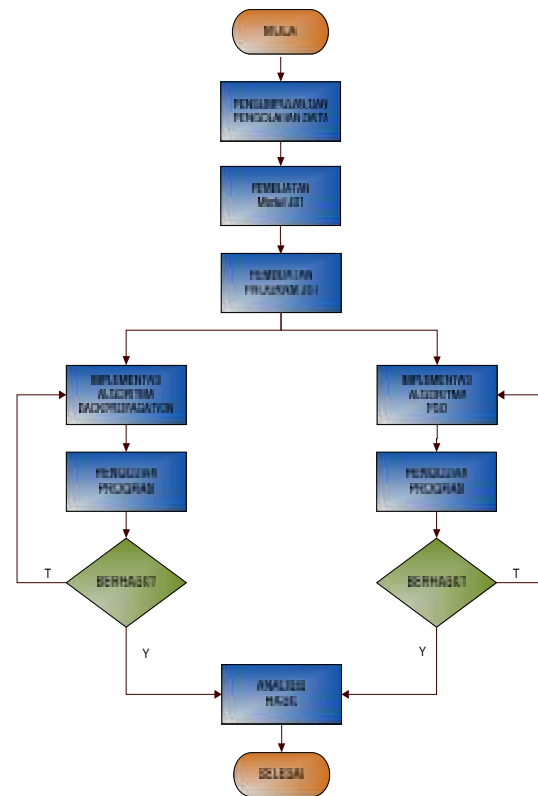
a. Jumlah Partikel

Pencarian nilai optimum dalam PSO dilakukan secara simultan terhadap sejumlah kandidat solusi yang mungkin yang disebut dengan *swarm*. *Swarm* merupakan kumpulan dari indi-

vidu tunggal yang disebut dengan partikel. Individu-individu tunggal dalam *swarm* dianalogikan sebagai suatu set solusi yang mungkin dari permasalahan optimisasi.

Penentuan jumlah partikel pada PSO dilakukan dengan memperhatikan lama waktu komputasi dan probabilitas dalam menemukan solusi optimum. Semakin besar jumlah partikel yang digunakan, probabilitas dalam menemukan nilai optimum juga lebih besar, namun akan membuat perhitungan menjadi lebih lama.

Jumlah partikel yang digunakan umumnya antara 20-50 partikel (Eberhart, 2001). Dalam penelitian ini digunakan 50 buah partikel.



Gambar 2. Diagram Alir Metodologi Penelitian

b. Bobot Inersia dan Konstanta Akselerasi

Bobot inersia (*inertia weight*) digunakan sebagai parameter untuk mengontrol pengaruh dari kecepatan partikel sebelumnya. Jika nilai inersia terlalu besar, kecepatan akan terus meningkat sehingga partikel akan divergen. Jarak partikel terhadap nilai optimumnya akan terus meningkat tiap iterasi.

Eberhart & Shi menyarankan nilai diturunkan secara bertahap dari 0.9 hingga 0.4 (Eberhart, 2001) menurut Persamaan 5.3 untuk menyeimbangkan fungsi eksplorasi (*global search*) dan eksploitasi (*local search*).

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{\text{Iter}_{\max}} \cdot \text{Iter} \quad (5.3)$$

Parameter lain dalam algoritma PSO adalah konstanta akselerasi c_1 dan c_2 . Konstanta c_1 disebut dengan parameter kognitif, sedangkan c_2 disebut parameter sosial. Konstanta akselerasi mempunyai pengaruh terhadap kecepatan konvergensi. Clerc mengenalkan suatu konsep *constriction factor* bagi penentuan c_1 dan c_2 agar menghasilkan konvergensi yang cepat, yaitu dengan memberikan nilai untuk parameter c_1 dan c_2 masing-masing 1.496 (Blum, 2008).

c. Laju Pembelajaran (Learning Rate)

Pemilihan laju pembelajaran biasanya dilakukan secara *trial and error*. Hal yang perlu diperhatikan adalah jika laju pembelajaran terlalu besar akan mengakibatkan algoritma menjadi tidak stabil.

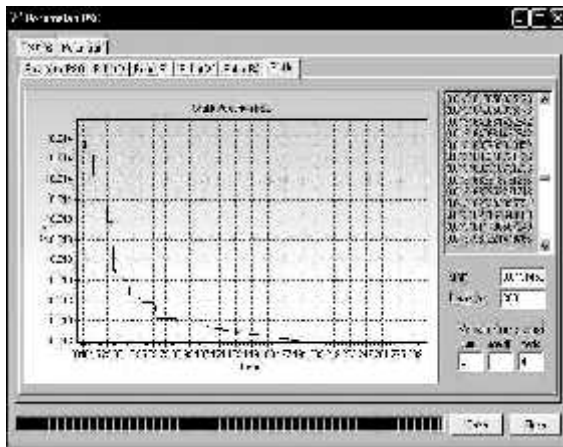
Sebaliknya, jika laju pembelajaran terlalu kecil, algoritma akan konvergen dalam waktu yang lebih lama. Meskipun begitu, laju pembelajaran yang rendah akan mempunyai *step size* yang kecil sehingga lebih menjamin hasil pencarian yang optimum. Pada penelitian ini digunakan laju pembelajaran sebesar 0.01.

Pengujian Prakiraan Temperatur Udara Minimum

Nilai parameter yang digunakan yaitu, inersia minimum dan maksimum masing-masing 0.4 dan 0.9, konstanta akselerasi c_1 dan c_2 adalah 1.496, jumlah partikel 50, dan panjang partikel 31.

a. Proses pelatihan

Proses pelatihan menghasilkan MSE sebesar 0.0131 pada iterasi ke-300 dengan lama waktu komputasi 1 menit 4 detik seperti ditunjukkan dalam Gambar 3.



Gambar 3. Proses Pelatihan Dengan PSO

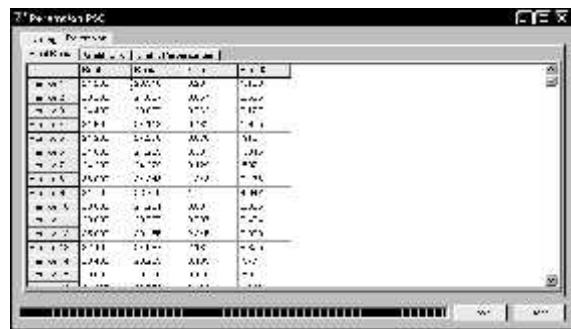
Pengaruh jumlah partikel terhadap lama waktu komputasi diberikan dalam Tabel 1. Dapat disimpulkan bahwa jumlah partikel yang besar menyebabkan waktu komputasi menjadi lebih lama.

Tabel 1. Pengaruh Jumlah Partikel Terhadap Waktu Komputasi

Jumlah Partikel	1	5	10	30	40	50	80	100
Waktu	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
MSE	0.015	0.013	0.012	0.011	0.010	0.009	0.008	0.007

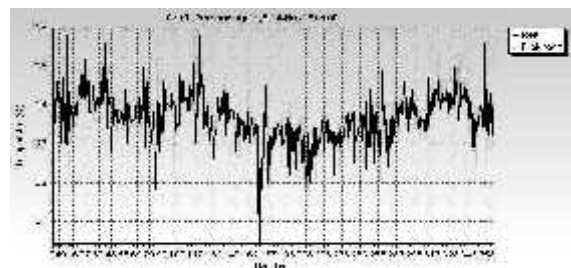
b. Proses Pengujian

Bobot yang dihasilkan pada pelatihan digunakan pada proses pengujian untuk memperkirakan T_{min} harian, kemudian diambil rata-rata kesalahan yang dihasilkan pada proses prakiraan tersebut. Dalam hal ini hanya dilakukan perambatan maju untuk mendapatkan output jaringan. Hasil prakiraan diberikan dalam Gambar 4. dengan rata-rata kesalahan 2.41%.



Gambar 4. Hasil Prakiraan T_{min} Dengan PSO

Grafik perbandingan antara temperatur minimum sesungguhnya (real) dengan hasil prakiraan ditunjukkan dalam Gambar 5.



Gambar 5. Perbandingan T_{min} Real Dengan Hasil Prakiraan

c. Perbandingan Dengan Backpropagation

Perbandingan algoritma PSO (jumlah partikel 50) dengan Backpropagation pada proses pelatihan diberikan dalam Tabel 2. Dari tabel tersebut menunjukkan bahwa pelatihan dengan PSO mencapai tingkat rata-rata error 2.42% pada iterasi ke-150, sedangkan Backpropagation pada iterasi yang dilakukan tersebut baru dapat mencapai 2.58%.

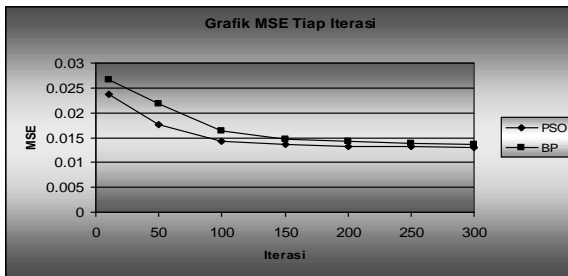
Pada tiap iterasi nilai MSE pelatihan PSO lebih baik dibandingkan dengan Backpropagation. Dengan kata lain, Backpropagation me-

merlukan epoch yang lebih banyak untuk mencapai MSE pelatihan dan rata-rata error prakiraan yang sama dengan PSO.

Tabel 2. Perbandingan PSO dengan Backpropagation Tiap Iterasi Pada Prakiraan Temperatur Minimum

Iterasi	MSE	50	100	150	200	250	300
		Backpropagation	0,0177	0,0142	0,0133	0,0122	0,0122
PSO	MSE	0,0177	0,0142	0,0133	0,0122	0,0122	0,0122
	Rata-rata Error	3,47%	3,40%	3,47%	3,47%	3,47%	3,47%
BP	MSE	0,0177	0,0142	0,0142	0,0142	0,0142	0,0142
	Rata-rata Error	2,73%	1,60%	2,50%	2,50%	2,49%	2,47%

Gambar 6. menunjukkan perbandingan *mean square error*/MSE antara algoritma belajar PSO dan Backpropagation pada proses pelatihan. Dari gambar tersebut dapat disimpulkan bahwa PSO lebih cepat konvergen dibandingkan dengan Backpropagation.



Gambar 6. Grafik Perbandingan Error Pada Proses Pelatihan

Perbandingan waktu komputasi antara PSO dengan Backpropagation diberikan pada Tabel 3. Jika ditinjau dari hal tersebut, Backpropagation memerlukan waktu perhitungan yang lebih singkat dibanding PSO, dimana waktu komputasi PSO sangat tergantung dari jumlah partikel yang digunakan.

Tabel 3. Perbandingan Waktu Komputasi PSO Dengan BP

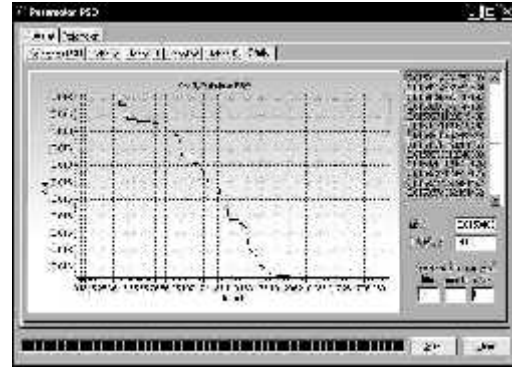
Iterasi	50	100	150	200	250	300
PSO	11 det	29 det	31 det	44 det	54 det	64 det
Backpropagation	1 det	1 det	1 det	1 det	2 det	2 det

Pada proses pengujian, penggunaan Backpropagation memberikan rata-rata kesalahan prakiraan lebih besar yaitu sebesar 2,47%.

Pengujian Prakiraan Temperatur Udara Maksimum

a. Proses Pelatihan

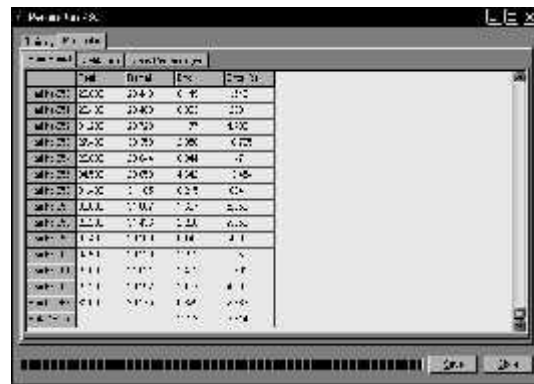
Proses pelatihan dengan algoritma PSO menghasilkan MSE sebesar 0.0158 dengan lama waktu komputasi 1 menit 5 detik jika menggunakan jumlah partikel 50, seperti ditunjukkan dalam Gambar 7.



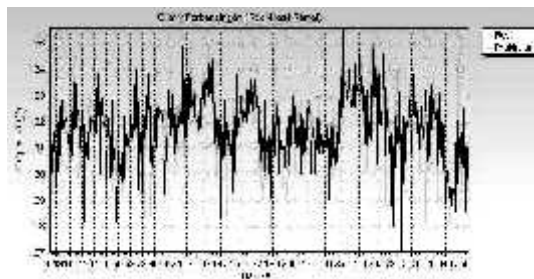
Gambar 7. Proses Pelatihan Dengan PSO

b. Proses Pengujian

Error proses pengujian prakiraan temperatur udara maksimum ditunjukkan dalam Gambar 8 dengan rata-rata error yang dihasilkan adalah sebesar 3.81%. Grafik perbandingan antara temperatur udara maksimum real dengan hasil prakiraan ditunjukkan dalam Gambar 9.



Gambar 8. Hasil Prakiraan T_{maks} Dengan PSO



Gambar 9. Perbandingan T_{maks} Real Dgn Hasil Prakiraan

c. Perbandingan Dengan Backpropagation

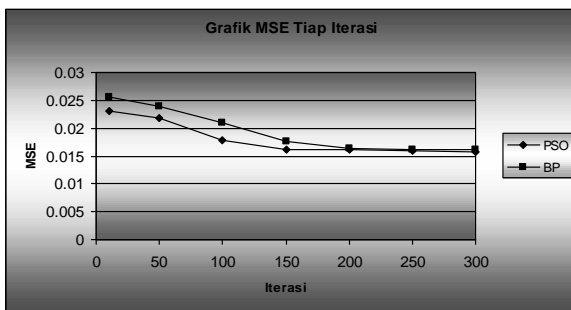
Perbandingan PSO dengan Backpropagation pada proses pelatihan jaringan diberikan dalam Tabel 4. Dari tabel tersebut menunjukkan bahwa pelatihan dengan Backpropagation mencapai tingkat rata-rata dari error prakiraan 3.82% pada iterasi ke-300, sedangkan dengan PSO pada iterasi ke-200. Pada tiap iterasi nilai MSE dan rata-rata error PSO juga lebih baik dibandingkan dengan Backpropagation.

Namun PSO memerlukan waktu komputasi pada proses pelatihan yang lebih lama dibandingkan dengan Backpropagation, dimana waktu komputasi PSO sangat tergantung dari jumlah partikel yang digunakan.

Tabel 4. Perbandingan PSO dgn Backpropagation Tiap Iterasi Pada Prakiraan Temperatur Maksimum

dasar ke-		90	100	150	200	250	300
PSO	MSE	0.0216	0.0176	0.0152	0.0161	0.0159	0.0158
	Prakiraan Error	4.38%	3.78%	3.54%	3.82%	3.82%	3.81%
	Waktu Komputasi	1.31	1.41	1.51	1.61	1.71	1.81
BP	MSE	0.0216	0.0176	0.0152	0.0161	0.0159	0.0158
	Prakiraan Error	4.38%	3.78%	3.54%	3.82%	3.82%	3.81%
	Waktu Komputasi	1.31	1.41	1.51	1.61	1.71	1.81

Gambar 10, menunjukkan perbandingan nilai MSE pada proses pelatihan antara algoritma belajar PSO dan Backpropagation. Seperti halnya pada aplikasi sebelumnya, pada prakiraan temperatur maksimum penggunaan PSO sebagai algoritma belajar akan lebih cepat konvergen dibandingkan dengan Backpropagation.



Gambar 10 Grafik Perbandingan Error Pada Proses Pelatihan

Pada proses pengujian, penggunaan Backpropagation memberikan rata-rata kesalahan prakiraan yang lebih besar yaitu sebesar 3.82%.

5. PENUTUP

Kesimpulan

Dari hasil pengujian, pada penelitian ini dapat ditarik kesimpulan sebagai berikut :

1. Jaringan syaraf tiruan yang dibangun dapat digunakan untuk prakiraan temperatur udara dengan hasil yang cukup baik, dengan rata-rata kesalahan prakiraan adalah sebesar 0.56 °C (2.41%) pada prakiraan temperatur minimum dan 1.20 °C (3.81%) untuk prakiraan temperatur maksimum
2. Pada proses pelatihan, penggunaan algoritma belajar Particle Swarm Optimization

(PSO) memberikan jumlah iterasi yang lebih sedikit pada tingkat kesalahan yang sama jika dibandingkan dengan Backpropagation. Akan tetapi waktu komputasi yang diperlukan menjadi lebih lama

3. Pada proses pengujian jaringan, penggunaan algoritma belajar Particle Swarm Optimization memberikan hasil prakiraan yang lebih baik dengan rata-rata tingkat kesalahan yang lebih kecil dibandingkan dengan Backpropagation

Saran

1. Pengembangan penelitian dapat dilakukan dengan topologi PSO yang berbeda sehingga memberikan proses belajar yang diharapkan lebih baik.
2. Untuk lebih meningkatkan kecepatan konvergensi pada proses pelatihan, algoritma PSO dapat dikombinasikan (hibrid) dengan algoritma optimasi berbasis gradien

6. DAFTAR PUSTAKA

1. Ayob, M., et al, 2010, *A Particle Swarm Optimization Approach for Routing in VLSI*, IEEE, pp. 49-53
2. Blum, C., dan Merkle, D. 2008. *Swarm Intelligence: Introduction and Applications*. Springer-Verlag. Berlin
3. Eberhart, R., dan Shi, Y. 2001. *Particle Swarm Optimization: Development, Applications, and Resources*. IEEE, pp. 81–86
4. Engelbrecht, P. 2002. *Computational Intelligence : An Introduction*. John Wiley & Sons Ltd. England
5. Fahlman, S., 1988, *Faster-Learning Variations on BackPropagation: An Empirical Study*, Proceedings Connectionist Models Summer School
6. Fausett, L. 1994. *Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*. Prentice-Hall Inc. USA
7. Kennedy, J., dan Eberhart, R. 2001. *Swarm Intelligence*. Academic Press. USA
8. Kusumadewi, S. 2004. *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*. Graha Ilmu. Yogyakarta
9. Siang, J. J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Andi. Yogyakarta
10. Yan, P., dan Tang, L. 2009, *PSO Algorithm for a Single Machine Scheduling Problem with Batching in Chemical Industries*, IEEE, pp. 45-49