

Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan
Vol. 08, No. 02, hal. 91- 96, Oktober 2019

Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan

ISSN 2615-6334 (Online)
ISSN 2087-5347 (Print)



Implementasi Paralel PSO dalam Melatih JST untuk Memprediksi Ketinggian Gelombang Laut

Muhamad Radzi Rathomi^{1,*}, Nola Ritha², Ferdi Chahyadi³

^{1,2,3}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Maritim Raja Ali Haji

^{1,2,3}Jl. Politeknik Senggarang, Tanjungpinang 29100

*Corresponding Author: radzi@umrah.ac.id

Abstract— Sea wave height is a major factor in seafaring safety. Prediction technology is needed to estimate the next height that will appear. Prediction can be done by studying the patterns of changes in sea wave height. Artificial Neural Networks can be used to make predictions, and Particle Swarm Optimization is used to train the weights and biases of neural networks in order to make predictions. However, the large amount of data on learning makes the training process time-consuming. This study proposes training processing techniques by dividing the task of processing this data. Processing is assigned with multithread programming techniques. Swarm divided by the number of threads raised and then transferred to each thread to be processed in parallel. The experimental results show that the prediction error of training in parallel is not worse than sequential training, but the speed of training in parallel is faster than sequential training.

Keywords— *Prediction, Training, ANN, PSO, Parallel.*

Intisari—Ketinggian gelombang laut menjadi faktor utama keselamatan pelayaran di laut. Teknologi prediksi diperlukan untuk memperkirakan ketinggian selanjutnya yang akan muncul. Prediksi bisa dilakukan dengan mempelajari pola-pola perubahan tinggi gelombang air laut. Jaringan Syaraf Tiruan dapat digunakan untuk melakukan prediksi, dan Particle Swarm Optimization digunakan untuk melatih bobot dan bias jaringan syaraf agar dapat melakukan prediksi. Namun, banyaknya data pada pembelajaran membuat pemrosesan pelatihan membutuhkan waktu yang lama. Penelitian ini mengusulkan teknik pemrosesan pelatihan dengan membagi tugas pemrosesan data-data tersebut. Pemrosesan ditugaskan dengan teknik pemrograman multithread. Swarm dibagi dengan jumlah thread yang dibangkitkan kemudian ditransfer ke masing-masing thread untuk diproses secara paralel. Hasil percobaan menunjukkan bahwa error prediksi dari pelatihan secara paralel tidak lebih buruk dari pelatihan secara sekuensial, namun kecepatan pelatihan secara paralel lebih cepat dibandingkan dengan pelatihan secara sekuensial

Kata kunci— *Prediksi, Pelatihan, JST, PSO, Paralel.*

I. PENDAHULUAN

Provinsi Kepulauan Riau merupakan wilayah yang terdiri dari lautan dan pulau-pulau yang tersebar. Transportasi laut menjadi salah satu pilihan mobilitas masyarakat yang dapat menghubungkan antar pulau di Kepulauan Riau,

khususnya pulau Bintan. Keadaan gelombang laut juga sangat mempengaruhi perputaran ekonomi yang terjadi di laut. Adanya gelombang laut yang tinggi memunculkan resiko yang besar bagi para nelayan maupun transportasi laut.

Banyak penelitian-penelitian yang mencoba untuk memprediksi keadaan gelombang laut untuk waktu berikutnya, seperti [1] dan [2]. Seiring dengan perkembangan model berbasis data, jaringan saraf tiruan (JST) telah digunakan dalam model peramalan gelombang laut [3]. Proses pembelajaran pada jaringan saraf tiruan dengan menggunakan *backpropagation* terkadang memiliki kendala *overfitting* sehingga tidak dapat menggeneralisasi masalah [4].

Pada penelitian ini, untuk mengatasi kekurangan yang dimiliki jaringan saraf tiruan maka pada proses pelatihan akan menggunakan *Particle Swarm Optimization* (PSO) untuk mendapatkan bobot dengan waktu komputasi yang cepat dan hasil yang optimal [5]. PSO banyak digunakan pada penelitian-penelitian optimasi [6]. Bahkan perbandingan juga dilakukan antara performa PSO dan performa Algoritma genetika [7]. Penelitian ini akan melakukan pelatihan JST menggunakan PSO. Namun, seiring waktu berjalan, jumlah data juga semakin tumbuh besar sehingga diperlukan suatu teknik yang bisa mempercepat pemrosesan.

Salah satu cara mempercepat pemrosesan data yang banyak adalah dengan membagi data tersebut menjadi sub-sub bagian dan memprosesnya secara paralel. Beberapa penelitian melakukan paralelisasi algoritma PSO seperti pada penelitian [8] dan [9]. Pada penelitian-penelitian tersebut belum ada yang mengimplementasikan paralelisasi PSO dalam melatih JST untuk melakukan prediksi ketinggian gelombang. Oleh karena itu, penelitian ini akan mencoba mengimplementasikan hal tersebut.

II. METODE PENELITIAN

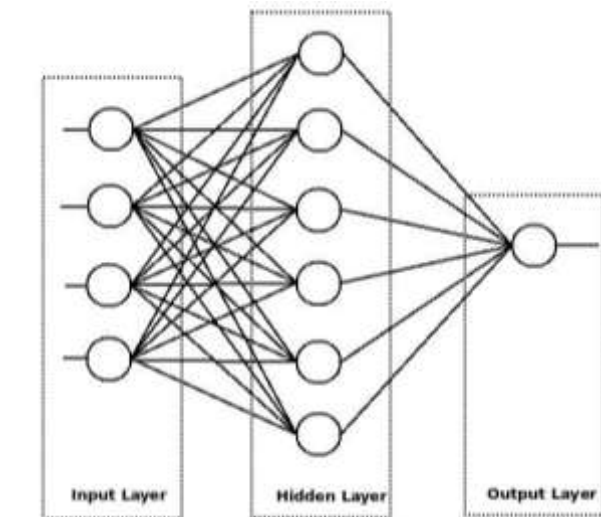
A. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan atau yang juga dikenal dengan istilah Neural Network (NN) adalah suatu sistem pemrosesan informasi yang memiliki kemiripan cara kerja dengan jaringan syaraf biologi. Pada jaringan syaraf biologi, otak manusia terdiri dari sejumlah neuron yang dapat melakukan tugas sederhana. Pada manusia, pemrosesan suatu informasi bersifat adaptif, dimana hubungan antar neuron terjadi secara

dinamis dan memiliki kemampuan untuk mempelajari informasi-informasi yang belum diketahui sebelumnya [10]. Jaringan syaraf tiruan memiliki kemampuan untuk sistem belajar dan melakukan generalisasi sehingga diharapkan sistem tidak hanya dapat mengenali data-data yang sudah pernah diberikan, tetapi juga terhadap data baru [11]. Pemrosesan informasi pada jaringan syaraf tiruan terletak pada suatu neuron yang memiliki sinyal. Setiap neuron masukan dan neuron keluaran memiliki lapisan tersembunyi atau yang biasa disebut dengan *hidden layer*. Penerapan jaringan syaraf tiruan telah banyak digunakan oleh para peneliti sebelumnya [12].

Jaringan Syaraf Tiruan dengan data time series dapat digunakan untuk melakukan peramalan dengan baik. Jaringan syaraf tiruan dikarakteristikan dengan pola koneksi diantara neuron yang disebut dengan arsitektur, menentukan bobot dari koneksi yang disebut *training* atau *learning*, dan fungsi aktivasi [13].

Pada penelitian ini, jaringan syaraf tiruan yang digunakan adalah Feed-forward Neural Network atau disingkat dengan FNN. Data yang digunakan untuk melakukan prediksi adalah data Time Series ketinggian gelombang laut, prediksi dilakukan dengan menggunakan data ketinggian rata-rata pada empat hari sebelumnya untuk memprediksi ketinggian gelombang laut pada hari berikutnya. Oleh karena itu, jumlah input pada JST yang akan dibuat adalah empat. Hidden layer hanya akan dibuat satu lapisan dengan enam hidden node. Output dari jaringan syaraf ini berjumlah satu output yang menghasilkan nilai ketinggian gelombang. Kesalahan hasil jaringan syaraf tiruan pada penelitian ini akan dikalkulasikan menggunakan *Mean Square Error*. Gambar 1 menunjukkan rancangan jaringan syaraf tiruan untuk memprediksi ketinggian gelombang laut.

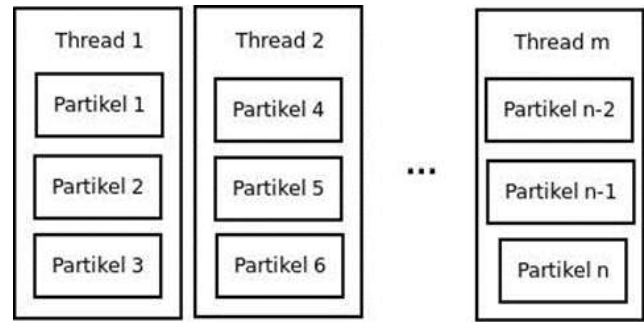


Gambar 1. JST untuk memprediksi ketinggian Gelombang Laut

B. Paralel PSO

Metode pelatihan yang akan diimplementasikan untuk jaringan syaraf tiruan adalah Algoritma *Swarm* atau *Particle Swarm Optimization*. Algoritma *Particle Swarm Optimization* (PSO) merupakan teknik optimasi berdasarkan populasi yang terinspirasi oleh perilaku sosial dari pergerakan burung atau ikan (*bird flocking* atau *fish schooling*) [14]. *Particle Swarm Optimization* digunakan sebagai alat optimasi yang menyediakan prosedur pencarian berbasis populasi dimana masing-masing individu yang dikenal dengan istilah partikel mengubah posisi mereka terhadap waktu [15]. Algoritma *Particle Swarm Optimization* ini mengoptimasi permasalahan dengan cara menggerakkan partikel didalam ruangan permasalahan menggunakan fungsi tertentu. Konsep PSO yang cukup sederhana, implementasi mudah, dan konvergensi cepat membuat *Particle Swarm Optimization* (PSO) dapat diterapkan untuk berbagai aplikasi di berbagai bidang untuk memecahkan masalah optimasi [16].

Optimasi akan dilakukan pada algoritma pelatihan dengan pengimplementasian teknik paralelisasi menggunakan multithread pada bahasa pemrograman java. Paralelisasi dilakukan dengan membagi pemrosesan partikel menjadi sub-sub *swarm*. Gambar 2 menunjukkan pembagian jumlah partikel pada setiap thread.



Gambar 2. Pembagian partikel pada setiap Thread

Pembagian partikel untuk setiap thread akan dibuat sama rata, hal ini bertujuan untuk menyeimbangkan kerja masing-masing thread. Jadi, ketika membangkitkan partikel, jumlah minimal partikel harus sama dengan jumlah thread yang dibangkitkan, jika tidak, algoritma akan langsung masuk ke dalam kriteria berhenti. Algoritma untuk membuat pembagian tersebut sama rata adalah sebagai berikut:

```

subPartikel ← numPartikel / numThread
sisa ← numPartikel modulus numThread
for i ← 0 to sisa:
    subPartikeli ← 1
    
```

Algoritma tersebut akan mengambil jumlah sisa pembagian, kemudian sisa pembagian akan dibagi rata lagi ke thread-thread yang berbeda. Misalnya jumlah partikel ditentukan sebanyak 43, kemudian jumlah thread yang dibangkitkan sebanyak 4, jika dibagikan, sisa pembagiannya adalah 3, sisa ini akan dimasukkan ke thread ke-1, thread ke-2 dan thread ke-3. Sehingga ada 3 thread yang memproses 11 partikel dan 1 thread yang memproses satu partikel.

Paralelisasi ini hanya akan diterapkan pada proses iterasi, sedangkan tahapan inisialisasi dan finalisasi tetap akan diproses secara sekuensial. Pada tahapan inisialisasi, tidak banyak proses yang bisa diproses secara parallel. Jika diimplementasikan maka akan menyebabkan pemborosan sumberdaya komputasi, mengingat penggunaan sumberdaya membutuhkan proses mulai yang memakan waktu. Begitu juga dengan proses finalisasi yang prosesnya lebih sedikit.

III. HASIL DAN PEMBAHASAN

Penelitian ini melakukan dua pengujian, yaitu pengujian terhadap nilai error antara pelatihan yang dilakukan secara paralel dan pengujian terhadap kecepatan dalam menyelesaikan epoch maksimal. Komputer yang digunakan adalah Raspberry Pi Model B dengan sistem operasi Raspbian dengan kelengkapan Java JDK. Dataset yang digunakan adalah data perubahan tinggi gelombang laut perjam yang diperoleh dari Badan Meteorologi Klimatologi dan Geofisika (BMKG) Tanjungpinang. Data tersebut akan dirata-ratakan perhari dan diambil selama tiga bulan. Untuk pelatihan, akan digunakan dataset sebanyak 70% dan untuk pengujian 30%. PSO untuk pengujian ini akan dijalankan dengan Maksimal Epoch 1000, w 0.729, C1 1.49, dan C2 1.49.

A. Pengujian Terhadap Nilai Error

Pada pengujian ini akan dilakukan perubahan jumlah partikel dari 10 hingga 100, nilai error akan dibandingkan antara pelatihan yang dilakukan secara paralel dengan pelatihan yang dilakukan secara sekuensial.

Tabel 1 menunjukkan error setiap pengujian sama antara pelatihan yang dijalankan secara paralel dan pelatihan yang dijalankan secara sekuensial. Berdasarkan hasil ini, paralelisasi PSO untuk melatih jaringan syaraf tiruan tidak menurunkan performa pelatihan. Hasil pengujian yang kurang bervariasi disebabkan oleh ukuran masalah yang kecil.

Tabel 1. Hasil pengujian Mean Squared Error

Jumlah Partikel	Sekuenial		Paralel	
	Pelatihan	Pengujian	Pelatihan	Pengujian
10	0.128	0.068	0.128	0.068
20	0.128	0.068	0.128	0.068
30	0.128	0.068	0.128	0.068
40	0.128	0.068	0.128	0.068
50	0.128	0.068	0.128	0.068
60	0.128	0.068	0.128	0.068
70	0.128	0.068	0.128	0.068
80	0.128	0.068	0.128	0.068
90	0.128	0.068	0.128	0.068

100	0.128	0.068	0.128	0.068
-----	-------	-------	-------	-------

B. Pengujian Kecepatan

Pengujian kecepatan menggunakan parameter yang sama dengan pengujian error. **Tabel 2** menunjukkan kecepatan pemrosesan dalam detik. Terlihat bahwa pengujian secara paralel lebih cepat dibandingkan dengan pengujian secara sekuensial. Hal tersebut dapat diperhatikan dari sisi peningkatan waktu, dimana pengujian sekuensial akan membutuhkan waktu yang lebih lama jika jumlah partikel ditingkatkan. Pembagian jumlah partikel yang sama rata sangat mempengaruhi peningkatan kecepatan ini. Jika satu thread mendapatkan beban yang sangat dominan perbedaannya, maka lama pemrosesan satu thread tersebut menjadi lama keseluruhan algoritma paralel.

Tabel 2. Perbandingan waktu pemrosesan

Jumlah Partikel	Sekuensial (detik)	Paralel (detik)
10	2.26	0.2
20	3.35	0.24
30	4.4	0.24
40	5.81	0.244
50	7.14	0.25
60	8.18	0.26
70	9.25	0.27
80	10.64	0.276
90	11.69	0.28
100	12.84	0.28

IV. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan, penelitian ini telah menunjukkan bahwa paralelisasi Algoritma *Particle Swarm Optimization* untuk pelatihan jaringan syaraf tiruan dapat meningkatkan kecepatan pemrosesan. Peningkatan kecepatan tersebut sangat berpatokan dengan pembagian beban komputasi yang sama rata.

Proses paralelisasi yang benar seharusnya tidak mengganggu kualitas hasil prediksi. Hasil percobaan menunjukkan bahwa error prediksi dari pelatihan secara paralel tidak lebih buruk dari pelatihan secara sekuensial. Hal ini dapat dilihat dari hasil pengujian, yaitu setiap *error* MSE yang dihasilkan Algoritma PSO paralel tidak lebih buruk jika dibandingkan dengan Algoritma PSO yang dijalankan secara sekuensial.

Rata-rata hasil prediksi yang dihasilkan oleh algoritma PSO dengan paralel dan sekuensial sudah mendekati nilai ketinggian sebenarnya, hal ini dapat dilihat dari kesalahan pelatihan sebesar 12% dan kesalahan pengujian sebesar 0.68%

UCAPAN TERIMA KASIH

Penelitian ini didukung oleh Lembaga Penelitian, Pengabdian Masyarakat dan Penjaminan Mutu Universitas Maritim Raja Ali Haji. Terutama DRPM Dikti yang telah memberikan dukungan dana pada skema Penelitian Dosen Pemula.

REFERENSI

- [1] Li, G., Hao, Y.L., Zao, Y., 2009, Research of Neural Network to Tidal Prediction, *IJCCSO*, DOI 10.1109/CSO.2009.347
- [2] Salim, A.M., Dwarakish, G.S., Liju, K.V., Thomas, J., Devi, G., Rajeesh, R., 2015, Weekly Prediction of tides using Neural networks, *Proceeding of 8th International Conference on Asian and Pacific Coast*, pp. 678 – 682
- [3] S. Oh, J. dan Suh, K.D., 2018, Real-time forecasting of wave heights using EOF-wavelet-neural network hybrid model, *Ocean Engineering*, 48-59
- [4] Nugraha, H.G. dan Azhari, SN., 2014, Optimasi Bobot Jaringan Syaraf Tiruan Menggunakan Particle Swarm Optimization, *IJCCS*, Vol. 8, 25-36
- [5] Wartati, D. dan Masrurroh, N.A., 2016, Aplikasi Jaringan Syaraf Tiruan dan Particle swarm Optimization untuk Peramalan Indeks Harga Saham Bursa Efek Indonesia, *Jurnal Tekno sains*, Vol. 6, No. 1, 22-30
- [6] Lalwani, S., Sharma, H., Satapathy, S.C., Deep, K., Bansal, J.C., 2019, A Survey on Parallel Particle Swarm Optimization Algorithm, *AJSE*, <https://doi.org/10.1007/s13369-018-03713-6>
- [7] LoBrutto, O.A., Guillou S.S., Thiebot, J., Gualous, H., 2016, Comparing Particle Swarm Optimization Method and Genetic Algorithm applied to the tidal farm layout Optimization Problem, <https://www.researchgate.net/publication/308967731>
- [8] Prasain, H., Jha, G.K., Thulasiraman, P., Thulasiram, R., 2010, A Parallel Particle Swarm Optimization Algorithm for Option Pricing, *IEEE*, 978-1-4244-6534-7
- [9] Venter, G., Sobieszcanski-Sobieski, J., 2005, A Parallel Particle Swarm Optimization Algorithm Accelerated by Asynchronous Evaluations, *6th World Congress of Structural and Multidisciplinary Optimization*, <https://ntrs.nasa.gov/search.jsp?R=20050182658>
- [10] Fausett, L., 1994, *Fundamental of Neural Network*, Prentice Hall, New York.
- [11] Paman, D.F., Muslim, M.A., dan Dhofir M, 2010, Analisis Implementasi Jaringan Syaraf Adaptif untuk Peramalan Energi Listrik Wilayah Malang *Jurnal Neutrino*, Vol.2, No.2, 117-133.
- [12] Ye, S., 2012, RMB Exchange Rate Forecast Approach Based on BP Neural Network, *Physics Procedia*, 287-293.
- [13] Ritonga, A.S., dan Atmoj, S., 2018, Penembangan Model Jaringan Syaraf Tiruan untuk Memprediksi Jumlah Mahasiswa Baru di PTS Surabaya, *Jurnal Ilmiah Teknologi Informasi Asia*, Vol. 12, No. 1.
- [14] Kennedy, J. dan Eberhart, R.C, 1995, Particle Swarm Optimization, *IEEE International Conference on Neural Networks*, 1942-1948.

- [15] Rosita, A., Purwanto, Y., dan Soelaiman, R., 2012, Implementasi Algoritma Particle Swarm untuk Menyelesaikan Sistem Persamaan Nonlinear, *Jurnal Teknik ITS*, Vol. 1.
- [16] Achyani, Y.E., 2018, Penerapan Metode Particle Swarm Optimization pada Optimasi Prediksi Pemasaran Langsung, *Jurnal Informatika*, Vol. 5, No.1, 1-11.