

KOMBINASI ALGORITMA CUCKOO-SEARCH DAN LEVENBERG-MARQUADT (CS-LM) PADA PROSES PELATIHAN ARTIFICIAL NEURAL NETWORK (ANN)

Bana Handaga¹, Hasyim Asy'ari²

¹Fakultas Ilmu Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta
Jl. A. Yani Tromol Pos 1 Pabelan Kartasura 57102 Telp 0271 717417
handaga.bana@gmail.com

²Jurusan Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Surakarta
Jl. A. Yani Tromol Pos 1 Pabelan Kartasura 57102 Telp 0271 717417
asy_98ari@yahoo.com

Abstrak

Artikel ini menjelaskan tentang sebuah hibrid algorithm yang menggabungkan antara algoritma *cuckoo-search* dengan algoritma *Levenberg-Marquardt* untuk melatih sebuah jaringan syaraf tiruan (*Artificial Neural Network – ANN*). Pelatihan sebuah ANN adalah sebuah pekerjaan optimisasi dengan tujuan untuk menemukan satu set bobot optimum pada jaringan melalui proses pelatihan. Algoritma pelatihan ANN tradisional terjebak pada nilai optimum yang bersifat lokal (*local minima*), sedangkan teknik pencarian nilai optimum yang bersifat global (*global minima*) memerlukan waktu yang lama atau berkerja sangat lambat. Selanjutnya telah dikembangkan model hybrid yang mengkombinasikan antara algoritma pencarian global-optimum dan algoritma lokal-optimum untuk melatih ANN. Pada penelitian ini, dilakukan sebuah penggabungan antara algoritma *cuckoo-search* dengan algoritma *Levenberg-Marquardt* (CS-LM) untuk melatih sebuah ANN. Hasil menunjukkan bahwa algoritma gabungan CS-LM memiliki performa yang lebih baik dibanding algoritma CS dan LM jika diterapkan secara individual.

Kata kunci: *Pelatihan Artificial Neural Network (ANN), Cuckoo-Search, Levenberg-Marquadt, kombinasi algoritma, multilayer perceptron (MLP).*

Pendahuluan

Karakteristik sebuah *Artificial Neural Network* (ANN) adalah dapat beradaptasi, mampu belajar dari contoh, dan mampu meniru perilaku secara umum yang dapat diterapkan untuk penyelesaian masalah dalam bidang klasifikasi pola, pendekatan fungsi, optimisasi, pencocokan pola dan memori asosiatif (Dayhoff, J. 1990; K. Mehrotra, et. all, 1997). Diantara berbagai model *neural network*, *multilayer feedforward neural network* atau *multilayer perceptron* (MLP) merupakan model yang lebih sering digunakan, karena sudah dikenal memiliki kemampuan pendekatan yang baik dan bersifat universal (S. Haykin, 1999),.

Keberhasilan *neural network* sangat bergantung pada arsitektur, algoritma pelatihan, dan pemilihan fitur yang digunakan dalam proses pelatihan. Semua itu menyebabkan masalah optimisasi dalam perancangan ANN menjadi masalah yang sulit (X. Yao, 1999). Dari kebanyakan pendekatan yang digunakan dalam perancangan ANN, masalah topologi dan fungsi transfer relatif sudah teratasi, selanjutnya permasalahan tertuju pada masalah pemilihan atau pengaturan nilai bobot (*weight*) dan *bias* pada ANN (D. Rumelhart et. al, 1986). Beberapa algoritma optimisasi untuk pengaturan bobot dan bias seperti *Ant Colony optimization* (C. Blum and K. Socha, 2005), *tabu search* (R. Sexton, 1998), *simulated annealing* dan *genetic algorithm* (R. Sexton, 1999) yang digunakan pada proses pelatihan ANN menggunakan topologi yang sudah tetap. Proses pembelajaran ANN yang menghasilkan konfigurasi nilai bobot dan bias yang optimum diasosiasikan terhadap kesalahan (*error*) terkecil yang terdapat pada bagian keluaran (output).

Sehubungan dengan pelatihan MLP, algoritma yang sering digunakan dalam pelatihan adalah algoritma *back-propagation* (BP) dan algoritma *Levenberg-Marquadt* (LM), dimana operasi dari keduanya berdasarkan pada gradient (*gradient-based*). Algoritma *back-propagation* mencapai konvergensi dalam tingkat turunan pertama (*first-order derivative*), sedangkan algoritma *Levenberg-Marquadt* mencapai konvergensi pada tingkat turunan yang kedua (*second-order derivative*). Diantara beberapa metode konvensional yang ada, para peneliti lebih memilih algoritma *Levenberg-Marquadt* karena algoritma tersebut memiliki kecepatan mencapai konvergensi dan akurasi yang lebih baik. Disisi lain, algoritma pelatihan yang beroperasi berdasarkan pada turunan (*derivative*) beresiko mencapai nilai minimum yang bersifat local (*local minima*). Untuk mengatasi permasalahan ini, beberapa teknik

pencarian nilai minimal yang bersifat global, yang dapat menghindari dari nilai minimum yang bersifat lokal, digunakan untuk mengatur bobot pada MLP, seperti *Evolutionary Algorithm* (EA), *simulated annealing* (SA), *tabu search* (TS), *ant colony optimization* (ACO), *particle swarm optimization* (PSO), *artificial bee colony* (ABC), dan *Cuckoo Search* (CS). (C. Blum and K. Socha, 2005; T. Back and H. P. Schwefel, 1993; B. Verma and R. Ghosh, 2002; D. Karaboga and C. Ozturk, 2009; X.-S. Yang, S. Deb 2009; C. Ozturk and D. Karaboga, 2011)

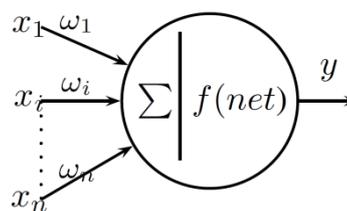
Algoritma *evolutionary* dan algoritma berbasis populasi yang lain memiliki performansi yang konsisten jika digunakan pada pelatihan MLP. Selanjutnya dengan mengambil keuntungan dari teknik pencarian nilai minimum yang bersifat global (*global minima*) dan kecepatan mencapai konvergensi pada teknik pelatihan konvensional, beberapa model algoritma yang mengkombinasikan kedua teknik tersebut telah digunakan dalam pelatihan ANN. Diantaranya adalah algoritma genetik dikombinasikan dengan metode pencarian local minima berbasis gradient BP dan LM (E. Alba and J. Chicano, 2004). Algoritma *simulated annealing* dikombinasikan dengan metode pencarian *local minima* berbasis gradient (N. Treadgold and T. Gedeon, 1998). Algoritma *tabu search* dikombinasikan dengan metode pencarian local berbasis gradient (T. Ludermit, et. all, 2006). Teknik *particle swarm optimization* (PSO) dikombinasi dengan algoritma pencarian *local minima* berbasis gradient (M. Carvalho and T. Ludermit, 2007), dan kombinasi antara teknik PSO dengan BP (L. Wang, et. all, 2007). Ada juga peneliti yang mengkombinasikan antara *adaptive* PSO dengan BP untuk melatih MLP (J. Zhang, et. all, 2007).

Penelitian mengenai penggunaan algoritma *Cuckoo Search* (CS) dalam pelatihan ANN telah dilakukan oleh (X.-S. Yang, S. Deb 2009; X.-S. Yang, S. Deb, 2010). Algoritma CS memiliki kemampuan yang sangat baik dalam menemukan *global minima*, sedangkan algoritma *Levenberg-Marquadt* (K. Levenberg, 1944) memiliki kemampuan yang baik untuk menemukan *local minima* (Hagan, M.T., and M. Menhaj, 1994). Motivasi dari penelitian ini adalah mengkombinasikan antara algoritma CS dengan algoritma LM untuk pelatihan MLP untuk mendapatkan keluaran dengan nilai kesalahan yang sekecil mungkin. Kombinasi algoritma ini merupakan sebuah kombinasi algoritma yang baru yang kita sebut dengan nama algoritma CS-LM.

Pelatihan ANN di terangkan pada bagian-2. Algoritma Cuckoo-Search (CS) dan versi kombinasi CS-LM di terangkan pada bagian-3. Pada bagian-4, dibahas tentang hasil ujicoba. Kesimpulan dan penelitian kedepan diberikan pada bagian-5.

Pelatihan ANN tipe Feed-Forward

Sebuah ANN terdiri atas element-element pemroses (lihat gambar 1), biasa disebut sebagai neuron-neuron atau node-node, yang saling berhubungan antara satu dengan yang lainnya (X. Yao, 1999). Dalam model ANN Feed-Forward, seperti ditunjukkan pada gambar 2, setiap node menerima sinyal dari node-node yang berada pada lapisan (layer) sebelumnya, dan pada setiap sinyal-sinyal tersebut dilakukan operasi perkalian dengan sebuah angka tertentu (disebut bobot), dimana bobot ini berbeda untuk masing-masing sinyal. Sinyal-sinyal input yang sudah dikalikan dengan bobot dijumlahkan, dan kemudian dilewatkan pada sebuah fungsi pembatas yang akan menghasilkan output berupa sebuah angka yang memiliki rentang nilai yang sudah ditetapkan. Selanjutnya, output dari pembatas dikirimkan ke semua node-node yang terdapat pada lapisan berikutnya. Sinyal input yang diterapkan pada bagian input di lapisan yang pertama, akan merambat masuk ke dalam jaringan (network), dan nilai output yang dihasilkan pada node ke-*i* dapat diterangkan dengan menggunakan persamaan (1).

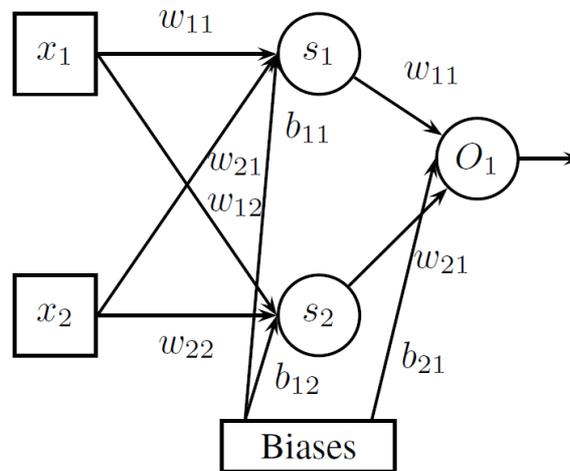


Gambar 1. Unit satuan pemroses pada ANN

$$y_i = f_i \left(\sum_{j=1}^n w_{ij} x_j + b_i \right) \tag{1}$$

y_i adalah output dari node, x_j adalah input node ke- j , w_{ij} adalah bobot penghubung antara node dan input x_j , b_i adalah merupakan sebuah bias (threshold) pada node, sedangkan f_i adalah fungsi transfer pada node. Biasanya fungsi transfer yang digunakan pada node berasal dari fungsi non-linear seperti fungsi heaviside, fungsi sigmoid, fungsi Gaussian dan lain-lain. Pada penelitian ini, digunakan fungsi transfer berupa fungsi sigmoid logarithmic (persamaan-2) pada lapisan tersembunyi (hidden) dan lapisan output.

$$y = f(net) = \frac{1}{1 + e^{-net}} \tag{2}$$



Gambar 2. Model ANN tipe multilayer feed-forward (MLP)

Tujuan sebuah optimisasi adalah untuk meminimisasi sebuah fungsi obyektif dengan cara mengoptimumkan bobot-bobot pada jaringan. Pada algoritma Evolutionary, ide terbesarnya adalah menginterpretasikan matrik-matrik bobot pada ANN sebagai individu, untuk merubah bobot. Pada penelitian ini, mean-square-error (MSE), diberikan pada persamaan-3, dipilih sebagai fungsi kesalahan yang dihasilkan oleh jaringan (ANN) dan proses adaptasi dilakukan untuk meminimisasi nilai MSE.

$$E(\vec{w}(t)) = \frac{1}{N} \sum_{j=1}^N \sum_{k=1}^K (d_k - o_k)^2 \tag{3}$$

dimana, $E(\vec{w}(t))$ adalah nilai kesalahan (*error*) pada iterasi ke- t ; $\vec{w}(t)$ adalah nilai bobot-bobot pada penghubung input pada iterasi ke- t , d_k merupakan nilai output yang diinginkan, sedangkan o_k adalah nilai output yang dihasilkan oleh ANN. K adalah jumlah output node dan N adalah jumlah data yang diterapkan pada input.

Algoritma Pelatihan

A. Algoritma Cuckoo-Search (CS)

Cuckoo atau adalah sejenis burung yang biasa disebut dengan burung tekukur, algoritma CS dibuat berdasarkan perilaku dari burung ini. Terdapat tiga aturan utama dalam algoritma CS (X.S. Yang and S. Deb, 2009, 2010), yaitu (1) Setiap burung tekukur akan meletakkan satu telur pada setiap waktu, dan telur itu diletakkan pada sebuah sarang yang dipilih secara acak. (2) Sarang yang terbaik dengan telur yang berkualitas tinggi akan melahirkan generasi berikutnya. (3) Jumlah rumah sarang adalah terbatas, telur yang diletakkan oleh tekukur akan ditemukan oleh burung pemilik sarang dengan probabilitas sebesar $p_a \in [0,1]$. Pada kasus ini burung pemilik sarang dapat membuang telur tersebut atau merusak sarang dimana telur itu berada, dan membangun sebuah sarang yang baru. Pada kasus tersebut penyederhanaan dapat dilakukan dengan cara mengambil sebagian nilai p_a dari sejumlah n sarang yang akan digantikan dengan sarang yang baru.

Kualitas (*fitness*) sebuah solusi dapat diambil dari sebuah nilai yang berbanding lurus dengan nilai fungsi obyektif. Bentuk kualitas solusi yang lain dapat di definisikan dengan cara yang serupa dengan algoritma genetika. Pada kasus ini setiap telur akan merepresentasikan sebuah solusi yang baru, tujuannya adalah menggunakan sebuah telur baru yang memiliki potensi (solusi) yang lebih baik untuk menjadi tekukur, untuk menggantikan telur (solusi) yang kurang baik yang terdapat dalam sarang. Algoritma ini dapat dikembangkan untuk kasus yang lebih kompleks dimana setiap sarang memiliki beberapa telur yang merepresentasikan satu set solusi. Pada penelitian ini menggunakan pendekatan yang sederhana yaitu bahwa setiap sarang hanya memiliki satu telur.

Berdasarkan pada tiga aturan diatas, langkah-langkah dasar algoritma CS dapat diringkas sebagai pseudo-code seperti ditunjukkan pada gambar-3. Pada saat membuat solusi baru $x^{(t+1)}$ untuk sebuah tekukur i , dilakukan sebuah ‘Levy flight’ yang direpresentasikan oleh persamaan-3 berikut

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda) \quad (4)$$

dimana $\alpha > 0$ adalah step-size yang berhubungan dengan masalah yang terkait. Pada kebanyakan kasus, dapat digunakan nilai $\alpha = 1$. Persamaan-3 di atas adalah merupakan persamaan stokastik dalam kasus 'random walk'. Secara umum sebuah 'random-walk' adalah sebuah 'Markov chain' yang memiliki status atau lokasi selanjutnya tergantung pada status/lokasi yang sekarang ($x_i^{(t)}$), dan probabilitas transisi ($\alpha \oplus Levy(\lambda)$). Symbol \oplus berarti sebuah operasi perkalian 'entrywise'. Operasi ini serupa dengan operasi yang digunakan pada algoritma PSO, tetapi operasi 'random walk' melalui 'Levy flight' yang digunakan di sini lebih efisien dalam mengeksplorasi ruang pencarian ketika jumlah tahapan menjadi lebih panjang.

```

begin
  fungsi obyektif  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_d)^T$ 
  Buat  $n$  populasi rumah sarang awal  $\mathbf{x}_i$ 
    ( $i = 1, 2, \dots, n$ )
  while ( $t < \text{MaxGeneration}$ ) or (stop criterion)
    Dapatkan sebuah tekukur secara acak
      dengan menggunakan Levy flights, dan
    Evaluasi kualitasnya (fitness)  $F_i$ 
    Pilih satu sarang diantara  $n$  sarang
      secara acak (katakana, sarang  $j$ ).
    if ( $F_i > F_j$ ),
      Gantikan sarang  $j$  dengan dengan
        solusi yang baru;
    end
    Buang sebagian sarang yang buruk (sebagian  $p_a$ )
      dan buat sarang yang baru;
    Simpan solusi yang terbaik;
    Susun rangkin berdasar kualitas solusi dan
      cari solusi yang terbaik;
  end while
  Pemrosesan selanjutnya dan visualisasi.
end

```

Gambar 3. Pseudo code algoritma CS

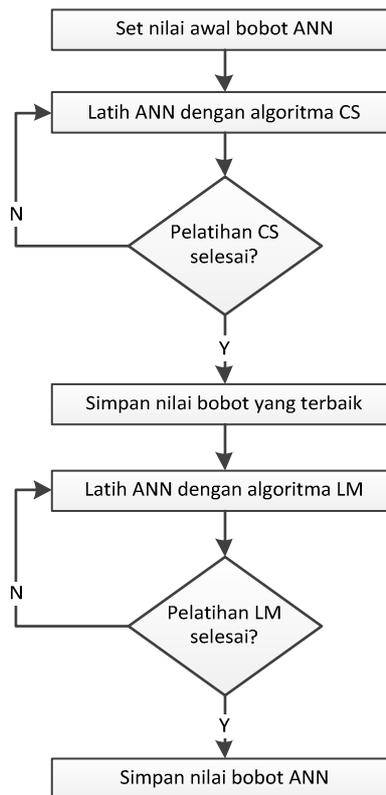
'Levy flight' akan menyediakan sebuah 'random walk' dengan 'random step length' ditentukan dari sebuah distribusi Levy dalam persamaan-5 berikut ini.

$$Levy \square u = t^{-\lambda}, \quad (1 < \lambda < 3) \quad (5)$$

B. Kombinasi CS dan LM

Algoritma Cuckoo-Search (CS) memiliki kemampuan yang baik untuk menemukan global-optimum sedangkan algoritma Levenberg-Marquadt (LM) memiliki kemampuan yang baik dalam menemukan lokal-minimal. Kombinasi algoritma dalam penelitian ini, merupakan pengkombinasian algoritma yang baru yaitu kombinasi antara algoritma CS dan LM. Pada kombinasi algoritma ini, pertama kali digunakan algoritma CS untuk mencari nilai optimum. Proses pelatihan selanjutnya dilakukan dengan algoritma LM. Algoritma LM merupakan algoritma yang menginterpolasi antara metode Newton dan 'gradient descent', yang menggunakan ekspresi orde-kedua (second-order) sebagai pendekatan untuk kesalahan yang dihasilkan oleh ANN.

Diagram alir untuk model CS-LM ditunjukkan pada Gambar-4. Dalam algoritma kombinasi CS-LM, algoritma CS beroperasi seperti pada (X.S. Yang and S. Deb, 2009; 2010). Pada tahap pertama algoritma CS menyelesaikan proses pelatihannya, kemudian dilanjutkan pelatihan dengan menggunakan algoritma LM dengan menggunakan bobot-bobot yang diperoleh dari proses pelatihan sebelumnya, dengan algoritma CS, dan melatih ANN untuk 100 epoch berikutnya.



Gambar 4. Diagram alir algoritma CS-LM

Metode Penelitian

Performa algoritma CS-LM untuk pelatihan ANN di uji cobakan untuk masalah Xor, Decoder-Encoder, dan 3 Bit Parity. Pada setiap masalah digunakan ANN tipe feed-forward yang terdiri atas tiga lapis neuron, yaitu satu lapis input, satu lapis hidden (tersembunyi) dan satu lapis output. Jumlah neuron (node) pada lapisan tersembunyi terdiri atas tiga sampai dengan enam neuron. Bias dan fungsi aktivasi tipe 'logsig' digunakan pada setiap node. Eksperimen pada model algoritma kombinasi CS-LM diulang 10 kali untuk setiap kasus yang berawal dengan pembuatan sarang tekukur pada algoritma CS. Pada ujicoba ini sebagai target utama adalah nilai kesalahan terkecil, dalam setiap kasus target kesalahan terkecil adalah 1×10^{-12} . Tiga variabel yang diamati pada percobaan ini adalah jumlah epoch, waktu, yang diperlukan untuk mencapai target tersebut. Pada percobaan ini digunakan sebuah laptop dengan mikroprosesor tipe Intel® Pentium® CPU 987 @1.50GHz, dengan memory sebesar 6 GB.

A. Kasus Exclusive-OR

Ujicoba yang pertama menggunakan fungsi boolean exclusive-OR (XOR) yang merupakan satu kasus klasifikasi yang sulit, yaitu memetakan dua input binary kepada satu output binary (0 0;0 1;1 0;1 1)→(0;1;1;0). Dalam simulasi ini digunakan tiga struktur ANN, yaitu pertama ANN tipe feedforward dengan struktur 2-4-1 dengan 12 buah bobot penghubung dan 5 bias (XOR17). Kedua struktur 2-5-1 dengan 19 buah bobot penghubung dan 6 bias, sehingga jumlah parameter menjadi 21 (XOR21). Struktur ketiga adalah struktur 2-6-1 dengan sembilan bobot penghubung dan empat bias sehingga total parameter berjumlah tigabelas (XOR25). Range nilai parameter untuk ketiga struktur adalah [-100,100] untuk struktur XOR17, [-100,100] untuk struktur XOR21, dan [-100,100] untuk struktur XOR25. Dua input dalam persamaan XOR akan menghasilkan sebuah output sesuai dengan nilai seperti ditunjukkan pada Tabel-1.

Tabel 1. Tabel kebenaran XOR

Input-1	Input-2	Output
0	0	0
0	1	1
1	0	1
1	1	0

B. Kasus 3-Bit Parity

Kasus kedua adalah kasus 3-bit parity, pada kasus ini akan mengambil nilai modulus 2 dari penjumlahan tiga input. ANN memiliki 12 bobot penghubung dan empat bias, sehingga secara keseluruhan ANN Memiliki 16 parameter. Pada kasus ini dipilih nilai parameter antara [-100,100]. Tiga input pada persamaan 3-Bit Parity akan menghasilkan sebuah output seperti ditunjukkan pada Tabel-2.

Tabel 2. Tabel Kebenaran 3-Bit Parity

Input-1	Input-2	Input-3	Output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

C. Kasus 4-Bit Encoder-Decoder

Kasus ketiga adalah kasus 4-bit encoder-decoder. Pada kasus ini, pada bagian input ANN diterapkan 4 pola input yang berbeda, masing-masing pola memiliki satu bit on dan tiga bit off. Target outut adalah sebuah duplikasi dari input seperti ditunjukkan pada Tabel-3, (0 0 0 1;0 0 1 0;0 1 0 0;1 0 0 0)→(0 0 0 1;0 0 1 0;0 1 0 0;1 0 0 0). Pada kasus ini digunakan ANN tipe feed-forward dengan struktur 4-2-4, total terdiri atas 22 parameter terdiri atas 18 bobot penghubung dan 6 bias. Pada kasus ini nilai parameter dibatasi antara [-10,10].

Tabel 3. Tabel kebenaran 4-Bit Encoder-Decoder

Inp-1	Inp-2	Inp-3	Inp-4	Out-1	Out-2	Out-3	Out-4
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
1	0	0	0	1	0	0	0

Hasil Eksperimen

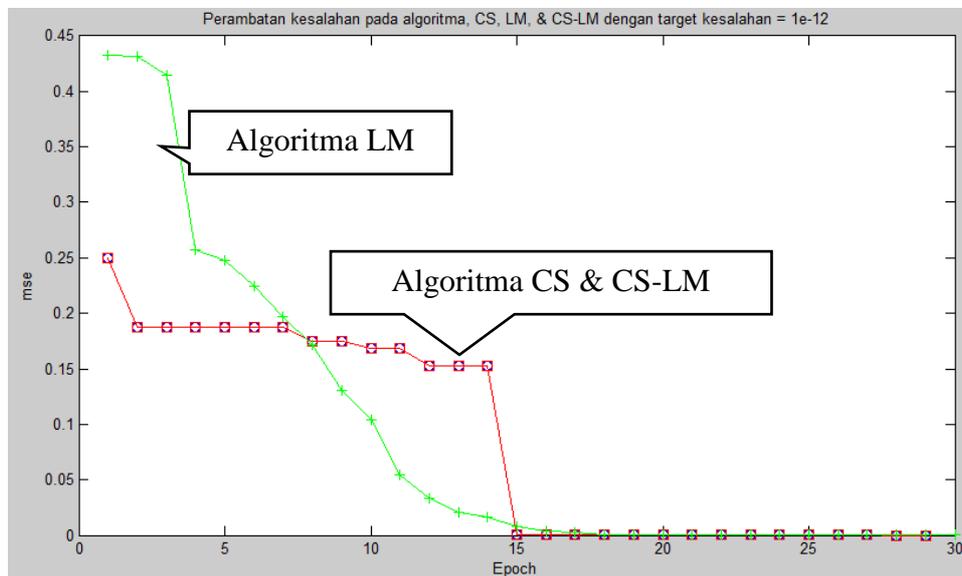
Hasil ujicoba algoritma secara statistic untuk kasus XOR6, XOR9, XOR13, 3-Bit Parity dan 4-Bit Encoder-Decoder ditunjukkan ada Tabel-4. Jelas bahwa dari Tabel-4 tersebut algoritma CS-LM memperoleh hasil yang terbaik dibanding dengan algoritma CS, dan LM jika diterapkan secara individual. Hasil ini seperti yang diharapkan, karena ANN dilatih dengan menggunakan set bobot penghubung terbaik yang dihasilkan dari pelatihan dengan algoritma CS, dan hasil yang diperoleh dari pelatihan algoritma CS lebih baik dari hasil yang diperoleh dari pelatihan dengan menggunakan algoritma LM. Selanjutnya, hasil dari pelatihan algoritma kombinasi CS-LM paling tidak sebanding dengan hasil yang diperoleh dari pelatihan dengan menggunakan algoritma CS. Terlihat bahwa hasil dari pelatihan dengan algoritma CS-LM memiliki kualitas yang paling baik dilihat dari nilai kesalahan. Algoritma CS-LM juga sangat handal karena memiliki standar deviasi yang sangat rendah.

Kesimpulan

Pada penelitian ini, dikembangkan sebuah kombinasi algoritma berdasar pada algoritma CS, digunakan untuk pelatihan sebuah ANN tipe feed-forwad, dan diujicobakan pada kasus XOR, 3-Bit Parity,dan 4-Bit Encoder-Decoder. Algoritma CS dikombinasikan dengan algoritma LM, pertama algoritma CS diguakan untuk melatih ANN dan diteruskan dengan pelatihan dengan menggunakan algoritma LM, dengan menggunakan set bobot-bobot penghubung terbaik yang dihasilkan dari pelatihan sebelumnya. Hasil eksperimen menunjukkan bahwa kombinasi algoritma CS-LM memiliki performa yang lebih baik dibanding performa darikedua algoritma jika diterapkan secara individual. Penelitian selanjutnya akan dipelajari imlementasi CS-LM pada kasus klasifikasi.

Tabel 4. Perbandingan jumlah epoch dan waktu yang diperlukan untuk mencapai target kesalahan sebesar 1×10^{-12} antara algoritma CS, LM, dan CS-LM

Kasus		LM	CS	CS-LM	
				CS	LM
XOR17	Time (second)	1.4	1.44	0.5	0.21
	Epoch	29	1400	750	13
XOR21	Time (second)	1.02	1.02	0.37	0.35
	Epoch	26	700	500	12
XOR25	Time (second)	0.51	0.27	0.16	0.18
	Epoch	32	250	200	12
3-Bit Parity	Time (second)	1.51	2.57	1.51	0.22
	Epoch	64	2250	1600	12
Encoder-Decoder	Time (second)	0.36	4.65	0.62	0.17
	Epoch	18	4050	750	12



Gambar 5. Perambatan Kesalahan selama proses pelatihan dengan target kesalahan sebesar 1×10^{-12}

Daftar Pustaka

B. Verma and R. Ghosh (2002), "A novel evolutionary neural learning algorithm, evolutionary computation," in *Proceedings of CEC'02*, May 12-17, pp. 1884-1889.

C. Blum and K. Socha (2005), "Training feed-forward neural networks with ant colony optimization: An application to pattern classification," pp. 233-238.

C. Blum and K. Socha (2005), "Training feed-forward neural networks with ant colony optimization: An application to pattern classification," pp. 233-238.

D. Karaboga and C. Ozturk (2009), "Neural networks training by artificial bee colony algorithm on pattern classification," *Neural Network World*, vol. 19(3), pp. 279-292.

D. Karaboga and C. Ozturk (2011), "A novel clustering approach: Artificial bee colony (abc) algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 652-657.

D. Karaboga, B. Akay, and C. Ozturk (2007), *Modeling Decisions for Artificial Intelligence*, ser. LNCS. Springer-Verlag, vol. 4617/2007, ch. Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks, pp. 318-329.

D. Rumelhart, G. Hinton, and R. Williams (1986), "Learning representations by backpropagation errors," *Nature*, vol. 323, pp. 533-536.

E. Alba and J. Chicano (2004), *Training Neural Networks with GA Hybrid Algorithms*, ser. Proc. of Gecco, LNCS. Springer-Verlag, pp. 852-863.

- Hagan, M.T., and M. Menhaj (1994), "Training feed-forward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, pp. 989–993.
- J. Dayhoff (1990), *Neural-Network Architectures: An Introduction*. New York: Van Nostrand Reinhold.
- J. Zhang, J. Zhang, T. Lok, and M. Lyu (2007), "A hybrid particle swarm optimization backpropagation algorithm for feedforward neural network training," *Applied Mathematics and Computation*, vol. 185, pp. 1026–1037.
- K. Mehrotra, C. Mohan and S. Ranka (1997), *Elements of Artificial Neural Networks*. Cambridge, MA: MIT Press
- Kenneth Levenberg (1944). "A Method for the Solution of Certain Non-Linear Problems in Least Squares". *Quarterly of Applied Mathematics* 2: 164–168.
- L. Wang, Y. Zeng, C. Gui, and H. Wang (2007), "Application of artificial neural network supported by bp and particle swarm optimization algorithm for evaluating the criticality class of spare parts," in *Third International Conference on Natural Computation (ICNC 2007)*, Haikou, China, August 24-27.
- M. Carvalho and T. Ludermir (2007), "Hybrid Training of Feed-Forward Neural Networks with Particle Swarm Optimization", ser. *LNCS. Springer-Verlag*, vol. 4233, pp. 1061–1070.
- N. Treadgold and T. Gedeon (1998), "Simulated annealing and weight decay in adaptive learning: the sarprop algorithm," *IEEE Transactions on Neural Networks*, vol. 9, pp. 662–668.
- R. Sexton, B. Alidaee, R. Dorsey, and J. Johnson (1998), "Global optimization for artificial neural networks: a tabu search application," *European Journal of Operational Research*, vol. 106, pp. 570–584.
- R. Sexton, R. Dorsey, and J. Johnson (1999), "Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing," *European Journal of Operational Research*, vol. 114, pp. 589–601.
- S. Haykin (1999), *Neural Networks a Comprehensive Foundation*. Prentice Hall, New Jersey.
- T. Back and H. P. Schwefel (1993), "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 1–23.
- T. Ludermir, A. Yamazaki, and C. Zanchetin (2006), "An optimization methodology for neural network weights and architectures," *IEEE Transactions on Neural Networks*, vol. 17(5), pp. 1452–1460.
- X. Yao (1999), "Evolving artificial neural networks," in *Proceedings of the IEEE*, vol. 87(9), pp. 1423–1447.
- X.S. Yang and S. Deb (2009), "Cuckoo search via Levy flights", in: *Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, December 2009, India, IEEE Publications, USA, pp. 210-214.
- X.S. Yang and S. Deb (2010), "Engineering optimization by cuckoo search", *Int. J. Mathematical Modelling and Numerical Optimisation*, Vol. 1, No. 4, 330-343.