

Optimasi *Training Neural Network* Menggunakan *Hybrid Adaptive Mutation* PSO-BP

Salnan Ratih Asriningtias, Harry Soekotjo Dachlan, dan Erni Yudaningtyas

Abstract—Optimization of training neural network using particle swarm optimization (PSO) and genetic algorithm (GA) is a solution backpropagation's problem. PSO often trapped in premature convergent (convergent at local optimum) and GA takes a long time to achieve convergent and crossover makes worse the results. In this research adaptive mutation particle swarm optimization and backpropagation (AMPPO-BP) is used for training the neural network of the iris plant, breast cancer, wine, glass identification and pima indian diabetes. The addition of PSO with adaptive mutation to prevent premature convergent and BP to increase the efficiency of local searching. AMPPO-BP training results will be compared with the GA and BP. The test results showed AMPPO-BP is able to optimize the process of training the neural network. AMPPO-BP more rapidly achieve the minimum error (global minimum), fast convergent and have the ability memorization and generalization with more accurate results than the other methods.

Index Terms—Adaptive Mutation, Backpropagation, Particle Swarm Optimization, Training Neural Network.

Abstrak—Optimasi *Training neural network* menggunakan Particle Swarm Optimization (PSO) dan Genetic Algorithm (GA) merupakan solusi untuk mengatasi kelemahan dari *backpropagation* (BP). PSO masih sering terjadi *premature convergent* (konvergen pada solusi optimum lokal), sedangkan GA membutuhkan waktu yang lama untuk mencapai konvergen dan *crossover* memungkinkan memperoleh hasil yang lebih buruk. Pada penelitian ini metode *hybrid adaptive mutation particle swarm optimization* dan *backpropagation* (AMPPO-BP) digunakan untuk *training neural network* pada *iris plant*, *breast cancer*, *wine*, *glass identification* dan *pima indian diabetes*. Penambahan PSO dengan *adaptive mutation* untuk mencegah *premature convergent* dan penggabungan dengan BP untuk meningkatkan efisiensi pencarian (*local searching*). Hasil *training* AMPPO-BP akan dibandingkan dengan GA dan BP. Hasil pengujian menunjukkan AMPPO-BP mampu mengoptimalkan proses *training neural network*. AMPPO-BP lebih cepat mencapai nilai eror minimum (*fast convergent*) yang bersifat *global minimum* dan memiliki kemampuan

memorisasi dan generalisasi yang optimum dengan hasil yang lebih akurat dibandingkan dengan metode yang.

Kata Kunci—Adaptive Mutation, Backpropagation, Particle Swarm Optimization, Training Neural Network.

I. PENDAHULUAN

NEURAL NETWORK adalah proses pengolahan informasi yang terinspirasi dari sistem kerja jaringan syaraf biologis manusia. *Neural network* terdiri dari *input layer*, *hidden layer* dan *output layer*, dimana masing-masing *layer* memiliki sejumlah *unit* yang saling terhubung antar *layer* dan memiliki bobot.

Hasil *output* dan kesimpulan *neural network* didasarkan pada pengalaman saat melakukan proses pembelajaran (*training*). Proses *training neural network* diawali dengan memberikan nilai bobot awal. Informasi yang sudah diketahui hasilnya dimasukkan ke dalam *unit* pada lapisan *input*. Bobot-bobot ini digunakan untuk mengingat pola informasi yang telah diberikan. Pengaturan bobot diatur secara terus menerus sampai diperoleh hasil yang diharapkan. Tujuan *training neural network* adalah untuk mencapai kemampuan memanggil kembali secara sempurna sebuah pola yang telah dipelajari (memorisasi) dan menghasilkan nilai *output* yang bisa diterima terhadap pola-pola yang serupa (tidak sama) yang disebut sebagai generalisasi [1].

Hal yang mempengaruhi kinerja *training neural network* adalah penentuan arsitektur *neural network* (jumlah *layer* dan *unit*) dan algoritma pembelajaran. Tidak ada ketentuan yang pasti dalam menentukan arsitektur *neural network*, tetapi ukuran arsitektur yang terlalu kecil berakibat *neural network* tidak mampu belajar dan sebaliknya ukuran arsitektur yang terlalu besar akan bersifat lemah dalam *generalisasi* dan memakan banyak waktu *training* [2].

Algoritma pembelajaran yang umum digunakan adalah *backpropagation* (BP), tetapi algoritma ini memiliki banyak kelemahan. Kelemahan-kelemahan BP diantaranya hanya bagus dalam aplikasi tertentu [3], sering terjebak pada lokal minimum, membutuhkan waktu yang lama untuk mencapai konvergen dan memiliki kemampuan yang rendah dalam belajar sehingga menghasilkan output yang tidak akurat [4].

Banyak penelitian yang telah dikembangkan untuk memperbaiki kelemahan BP diantaranya menggunakan metode *Particle Swarm Optimization* (PSO) dan

Salnan Ratih Asriningtias adalah mahasiswa Program Studi Magister Teknik Elektro, Universitas Brawijaya, Malang, Indonesia (085646582021; email salnanratih@gmail.com).

Harry Soekotjo Dachlan adalah dosen Program Studi Magister Teknik Elektro, Universitas Brawijaya, Malang, Indonesia (email : harysd@brawijaya.ac.id).

Erni Yudaningtyas adalah dosen Program Studi Magister Teknik Elektro, Universitas Brawijaya, Malang, Indonesia (email : erni_yudaningtyas@yahoo.co.id).

Genetic Algorithm (GA). Training neural network menggunakan PSO membutuhkan sedikit parameter dan membutuhkan waktu yang lebih cepat untuk mencapai konvergen [5], tetapi lemah dalam pencarian lokal dan sering terjadi *premature convergent* (konvergen pada solusi optimum lokal) karena memiliki ruang pencarian yang terbatas [6]. *Training neural network* menggunakan GA lebih optimal dibandingkan dengan PSO karena adanya faktor *mutation* dan *crossover* yang menyebabkan ruang pencarian solusi GA lebih besar untuk menghasilkan solusi baru [7], tetapi GA membutuhkan waktu yang lama untuk mencapai konvergen dan proses *crossover* pada GA memungkinkan hasil yang lebih buruk [8].

Fokus dalam penelitian ini adalah optimasi *training neural network* menggunakan data klasifikasi yaitu *iris plant, breast cancer, glass identification, wine* dan *pima indian diabetes*. Metode yang diusulkan adalah *hybrid adaptive mutation particle swarm optimization* dan *backpropagation* (AMPSO-BP). *Particle swarm optimization* (PSO) digunakan untuk meningkatkan *convergent rate* dan mendapatkan solusi optimum global. *Adaptive mutation* pada PSO berfungsi untuk memperlebar ruang pencarian PSO untuk mencegah *premature convergent*. *Backpropagation* (BP) digunakan untuk meningkatkan efisiensi pencarian lokal.

II. LANDASAN TEORI

A. Backpropagation (BP)

BP adalah metode berbasis penurunan gradien untuk meminimalkan kuadrat eror hasil keluaran [9]. BP dapat meningkatkan efisiensi pencarian dan bersifat *local optimization*.

BP melakukan dua tahap perhitungan. Perhitungan maju untuk menghitung eror antara nilai output dan target dan perhitungan mundur yang mempropagasikan balik eror tersebut dari *output* ke *input* untuk memperbaiki bobot-bobot pada semua *unit*. Metode ini hanya melakukan perubahan bobot berdasarkan fungsi eror, maka kombinasi bobot yang mampu meminimalkan fungsi eror dianggap sebagai solusi permasalahan. Fungsi aktivasi yang digunakan dalam BP adalah fungsi sigmoid biner yang memiliki interval 0 sampai 1.

B. Particle Swarm Optimization (PSO)

PSO adalah teknik berbasis populasi (sejumlah partikel) dengan mengatur posisi dan kecepatan yang mengacu pada partikel yang optimum untuk mencapai solusi [10]. PSO dimulai dengan membangkitkan partikel (solusi) secara acak dan setiap partikel dievaluasi dengan menggunakan fungsi *fitness*. Partikel akan terbang mengikuti partikel yang optimum dari satu posisi ke posisi yang baru dengan cara menambahkan kecepatan (v) ke posisi (x) untuk mendapatkan posisi baru (x').

Perubahan posisi dan kecepatan masing-masing partikel dipengaruhi oleh posisi terbaik yang diperoleh semua partikel ($gbest$) dan posisi terbaik masing-masing

partikel ($pbest$). Persamaan yang digunakan untuk memperbarui kecepatan dan posisi dapat dilihat pada persamaan (1) dan (2):

$$v'_{id} = \omega * v_{id} + c_1 * r * (p_{id} - x_{id}) + c_2 * r * (g_{id} - x_{id}) \quad (1)$$

$$x'_{id} = x_{id} + v_{id} \quad (2)$$

dimana i adalah partikel ke- i dan d adalah dimensi ke- d . v adalah kecepatan partikel. ω adalah *inertia weight*. c_1 adalah *learning rate* untuk komponen *cognition* (kecerdasan individu) dan c_2 adalah *learning rate* untuk komponen *social* (kecerdasan sosial antar individu). p adalah $pbest$ dan g adalah $gbest$. r adalah bilangan acak dalam interval (0,1).

C. Adaptive Mutation pada PSO

PSO merupakan teknik berbasis populasi dengan mengatur posisi dan kecepatan yang mengacu pada partikel yang terbaik untuk mencapai solusi. Posisi selanjutnya yang akan ditempuh partikel dipengaruhi oleh posisi terbaik partikel sebelumnya dan posisi terbaik oleh semua partikel. Jika posisi terbaik tidak terjadi perubahan di *local optimum*, maka partikel akan cepat konvergen di posisi terbaiknya. Hal ini menyebabkan terjadinya konvergen di *local optimum* yang disebut sebagai *premature convergent*. *Premature convergent* dapat dicegah dengan mengubah posisi terbaik saat ini dengan posisi baru menggunakan proses mutasi.

Mutasi pada PSO dilakukan dengan menambahkan angka random dengan range (0,1) pada $gbest$. Penggunaan mutasi pada PSO masih belum mampu mencegah *local optimum*, karena terlalu kecil menjangkau ruang pencarian secara menyeluruh. Solusi baru untuk mengatasi terbatasnya jangkauan partikel yang dimutasi adalah dengan menambahkan ukuran mutasi yang terkait dengan ukuran ruang pencarian yang disebut *adaptive mutation*. Persamaan *adaptive mutation* pada PSO dapat dilihat pada persamaan (3) [11].

$$gbest_j(t+1) = gbest_j(t) + [b_j(t) - a_j(t)] * rand() \quad (3)$$

Dimana $gbest$ adalah vektor $gbest$ partikel. j adalah dimensi ke- j partikel. $a_j(t)$ adalah nilai minimum dimensi ke- j partikel. $b_j(t)$ adalah nilai maksimum dimensi ke- j partikel.

III. METODE PENELITIAN

A. Data

Data digunakan untuk proses *training* dan pengujian metode *training neural network*. Data yang digunakan adalah data skunder berupa dataset dengan kategori permasalahan klasifikasi yang diambil dari [12]. Deskripsi dari data-data tersebut diantaranya:

- *Iris plant*: dikenal sebagai literatur database untuk pengenalan pola. Data ini terdiri dari 3 kelas dimana masing-masing kelas terdiri dari 50 data. Masing-masing kelas merupakan tipe *iris plant* yaitu *iris setosa, iris versicolour* dan *iris virginica*. Pembagian data ke dalam 3 kelas didasarkan 4 jenis atribut.

- *Breast cancer*: database yang berisi diagnosa penyakit kanker payudara dari *university of Wisconsin Hospital, Medison* oleh Dr. William H. Wolberg. Data ini terdiri dari 2 kelas (*benign* dan *malignant*) yang merupakan jenis dari *breast cancer*. Total jumlah data adalah 699 dengan 10 jenis atribut.
- *Wine*: data hasil dari analisis kimia dari *wine* yang berkembang di wilayah italia tetapi dengan 3 pengolahan yang berbeda. Hasil analisis menunjukkan terdapat 13 unsur yang ditemukan dalam masing-masing tiga jenis wine yang berbeda. Total jumlah data adalah 178.
- *Glass identification*: database yang berisi klasifikasi tentang tipe *glass*. Data ini mendukung investigasi kriminal. Data ini dibagi ke dalam 7 kelas dengan 10 jenis atribut. Total jumlah data adalah 214.
- *Pima indian diabetes*: database yang berisi data pasien yang terkena penyakit diabetes. Semua pasien berjenis kelamin perempuan dengan usia 21 tahun yang merupakan keturunan *Pima Indian*. Pasien dibagi ke dalam 2 kelas yaitu positif (terkena diabetes) dan negatif (tidak terkena diabetes) yang didasarkan pada 8 jenis atribut dengan jumlah data sebanyak 768.

Data hanya digunakan 70% untuk proses *training neural network* dan sisanya digunakan untuk proses pengujian atau aplikasi. Normalisasi data dilakukan untuk mengoptimalkan proses *training neural network*. Dengan adanya normalisasi data, semua data akan diubah nilainya dalam kisaran 0 dan 1.

B. AMPSO-BP untuk Training Neural Network

Langkah pertama dalam proses *training neural network* adalah penentuan arsitektur *neural network* yang disesuaikan dengan jenis data yang digunakan. Arsitektur meliputi jumlah *layer* dan jumlah *unit* pada masing-masing *layer* serta penentuan jumlah bobot dan bias pada *neural network*. Jumlah *layer* yang akan digunakan adalah 3 *layer* yang terdiri dari 1 *input layer*, 1 *hidden layer* dan 1 *output layer*.

Jumlah unit pada input layer dan output layer disesuaikan dengan permasalahan untuk masing-masing jenis data. Jumlah unit pada hidden layer ditentukan dengan persamaan (4)[13].

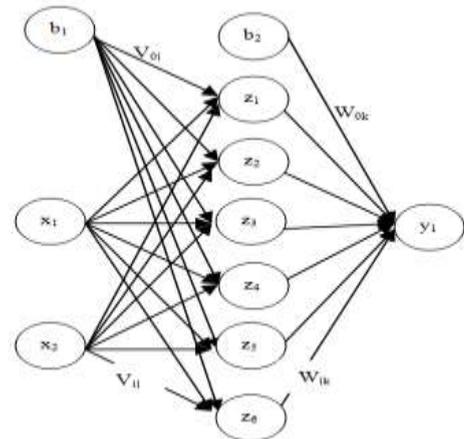
$$H = 3M \tag{4}$$

Dimana H adalah jumlah *unit* pada *hidden layer* dan M adalah jumlah *unit* pada *input layer*. Sebagai contoh diketahui 2 *unit input* dan 1 *unit output* maka jumlah *unit hidden* adalah 6 *unit*. Dalam arsitektur terdapat bobot dan bias. Perhitungan jumlah bobot dan bias ditentukan dengan persamaan (5)[14].

$$w = mxn + lxn + nxq + lxq = (m + 1)n + (n + 1)q \tag{5}$$

Dimana w adalah jumlah bobot dan bias, m adalah jumlah *unit input*, n adalah jumlah *unit hidden* dan q adalah jumlah *unit output*. Penambahan nilai 1 adalah penambahan 1 *unit* untuk bias. Sesuai persamaan (5) maka ada penambahan 1 *unit* pada *input layer* dan

hidden layer, sehingga arsitektur *neural network* dapat dilihat dalam Gambar 1.



Gambar 1. Arsitektur *Neural Network*. Arsitektur *neural network* terdiri atas $x = \text{input unit}$, $z = \text{hidden unit}$, $y = \text{output unit}$, b_1 dan b_2 adalah bias, v_{ij} = bobot antara *layer i* (input) dan *layer j* (hidden), w_{jk} = bobot antara *layer j* (hidden) dan *layer k* (output)

Solusi yang diperoleh dari *training neural network* adalah berupa bobot yang nantinya akan digunakan dalam aplikasi. Penentuan nilai bobot dalam *training neural network* menggunakan AMPSO-BP memiliki langkah yang berbeda dengan menggunakan BP. Dalam AMPSO-BP solusi direpresentasikan ke dalam partikel dimana 1 partikel merupakan 1 solusi dan 1 partikel memiliki dimensi. Bobot yang menjadi solusi *training neural network* akan direpresentasikan ke dalam dimensi partikel. Struktur dimensi partikel dalam *training neural network* yang didasarkan pada persamaan (5) dan Gambar 1. dapat dilihat dalam Gambar 2.

Partikel (x)	v_{ij} (mm)						v_{0j} (mm)				w_{jk} (mm)						w_{0k} (mm)	
x_1	v_{11}	v_{12}	...	v_{15}	v_{16}	v_{17}	v_{01}	...	v_{06}	w_{11}	...	w_{16}	w_{17}	...	w_{18}	w_{01}	w_{02}	

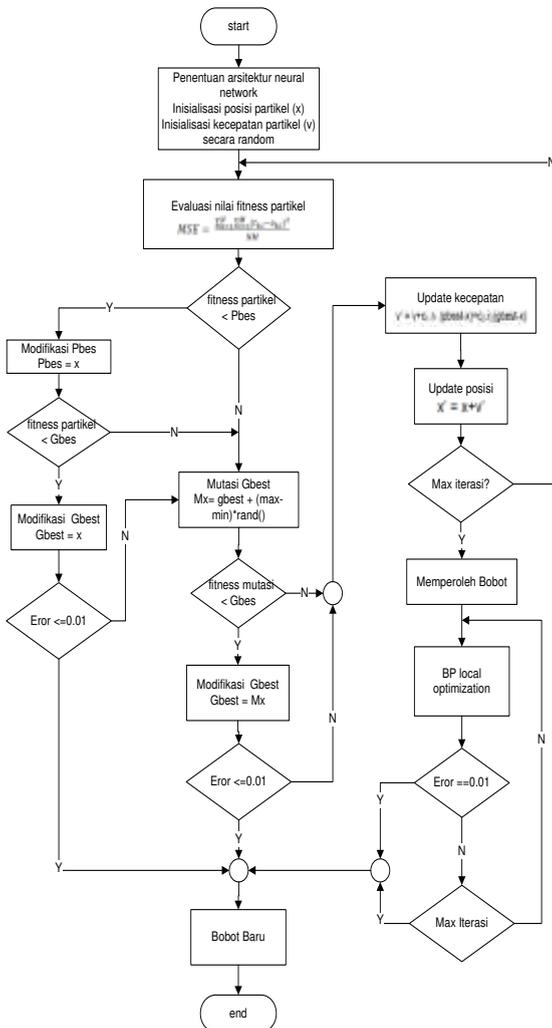
Gambar 2. Struktur dimensi partikel. Dimensi partikel terdiri dari v_{ij} = bobot antara input layer dan hidden layer, v_{0j} = bias pada hidden layer, w_{jk} = bobot antara hidden layer dan output layer dan w_{0k} = bias pada output layer

Dalam proses *training*, jumlah partikel yang digunakan lebih dari 1 partikel dan setiap partikel membawa solusi. Untuk menentukan partikel mana yang memiliki solusi terbaik, setiap partikel memiliki nilai *fitness* yang dievaluasi menggunakan fungsi *fitness*. Nilai *fitness* adalah ukuran kualitas suatu partikel sedangkan fungsi *fitness* adalah fungsi yang digunakan untuk evaluasi partikel untuk mendapatkan nilai *fitness*. Fungsi *fitness* dalam *training neural network* adalah fungsi untuk menghitung nilai *error* atau selisih nilai antara nilai target dan nilai *input*. Fungsi yang digunakan adalah *mean squared error* (MSE) yang dapat dilihat dalam persamaan (6).

$$MSE = \frac{\sum_{k=1}^N \sum_{l=1}^M (t_{k,l} - o_{k,l})^2}{NM} \tag{6}$$

Dimana N adalah jumlah pola *input*, M adalah jumlah *unit input*, t adalah nilai target dan o adalah nilai *output*. *Training neural network* merupakan proses

perubahan bobot (dimensi) dengan memberikan nilai input dan membandingkan nilai output dengan nilai target. Perubahan bobot dipengaruhi oleh perubahan posisi dan kecepatan masing-masing partikel yang sesuai dengan persamaan (1) dan (2). Pada setiap generasi, posisi dan kecepatan partikel dipengaruhi oleh $pbest$ dan $gbest$ partikel. Perubahan dilakukan secara berkelanjutan hingga mencapai kriteria penghentian tertentu. Detail proses *training neural network* menggunakan AMPSO-BP dapat dilihat dalam Gambar 3.



Gambar 3. Detail Proses *Training Neural Network* Menggunakan AMPSO-BP

Terdapat beberapa macam kriteria penghentian untuk proses *training* yang bisa digunakan, diantaranya [15]:

- Proses *training* berhenti dengan nilai *fitness* atau *error* tertentu.
- Proses *training* berhenti dalam jumlah iterasi tertentu.
- Proses *training* berhenti jika mengalami konvergen yaitu terjadi kemiripan nilai untuk semua partikel atau dalam beberapa iterasi tidak terjadi perubahan nilai.
- Proses *training* berhenti dengan batasan waktu tertentu.

Hasil dari *training neural network* berupa bobot yang

direpresentasikan oleh $gbest$. Untuk mencegah *premature convergent* dilakukan proses mutasi (*adaptive mutation*) pada $gbest$ yang sesuai dengan persamaan (3). *Adaptive mutation* merupakan proses mutasi yang menggunakan ukuran ruang pencarian yaitu pada $pbest$ partikel. Proses mutasi pada $gbest$ hanya dilakukan pada dimensi tertentu berdasarkan probabilitas mutasi (Pm) yang telah ditentukan.

C. Inisialisasi Parameter Training Neural Network

Parameter yang digunakan dalam AMPSO-BP dapat dilihat dalam Tabel 1

TABEL I
INISIALISASI PARAMETER

Metode	Parameter
PSO	$\theta = 0.8$ $v = \text{rand}(-2,2)$ bobot (w) = $\text{rand}(0,1)$ jumlah partikel (t) = 20 $c_1 = c_2 = 2$ $r_1 = r_2 = \text{rand}(0,1)$ $V_{\max} = 2$
BP	$Pm = 0.02$ $\alpha = 0.5$

IV. HASIL DAN PEMBAHASAN

A. Perbandingan Nilai Error Training

Nilai *error training neural network* untuk semua data dengan semua metode dalam 10 iterasi dapat dilihat dalam Tabel 2.

TABEL II
ERROR TRAINING

Data	AMPSO-BP	GA	BP
Iris Plant	9.4×10^{-14}	11.119	0.3307
Breast Cancer	0.0048	78.170	1.4×10^{-5}
Glass	0.0067	6.5494	0.1297
Identification			
Wine	0.1888	12.709	0.2352
Pima Indian	0.0057	65.579	0.0063
Diabetes			

Berdasarkan Tabel 2. Nilai *error training neural network* untuk AMPSO-BP paling baik jika dibandingkan dengan metode yang lain untuk keseluruhan data.

B. Perbandingan Waktu Training

Waktu *training neural network* untuk semua data dengan semua metode dalam 10 iterasi dapat dilihat dalam Tabel 3.

Berdasarkan Tabel 3. waktu proses training berbeda-beda untuk masing-masing data dan metode. Hal ini disebabkan proses training untuk masing-masing metode berbeda. Berdasarkan proses *training* terdapat tiga jenis metode diantaranya metode *hybrid* yaitu *global optimization* dan *local optimization* (AMPSO-BP), metode *global optimization* (GA), dan metode *local optimization* (BP). Metode *hybrid* lebih lama jika dibandingkan dengan metode *global optimization*. Dalam jumlah iterasi maksimum yang sama metode

global optimization lebih cepat dibandingkan dengan *local optimization*, karena proses perubahan bobot pada *local optimization* dilakukan di setiap data sedangkan pada metode *global optimization* dilakukan di semua data. Selain proses *training* lama waktu *training* dipengaruhi jumlah data dan kompleksitas arsitektur *neural network*. Semakin banyak jumlah data *training* maka proses *training* membutuhkan waktu lebih lama dan semakin kompleks arsitektur *neural network* maka proses *training* juga membutuhkan waktu lebih lama.

TABEL III
WAKTU TRAINING

Data	AMPSO-BP	GA	BP
Iris Plant	57 m 50 s	20 m 35 s	131 m 34 s
Breast Cancer	994 m 15 s	262 m 32 s	59 s
Glass Identification	1303 m 16 s	248 m 59 s	1377 m
Wine	385 m 5 s	275 m 13 s	682 m 33 s
Pima Indian	312 m 6 s	202 m 50 s	11 s
Diabetes			

C. Perbandingan Nilai CEP (Classification Error Percentage)

Perbandingan nilai CEP *training neural network* untuk 10 iterasi dapat dilihat dalam Tabel 4. Berdasarkan Tabel 4, rata-rata nilai CEP untuk AMPSO-BP lebih baik dibandingkan dengan metode yang lain, sehingga AMPSO-BP memiliki kemampuan memorisasi dan generasilisasi lebih baik. Nilai *error* yang lebih minimum ketika *training neural network* belum tentu memberikan nilai CEP yang lebih baik. Hal ini disebabkan adanya faktor perbedaan proses *training* untuk jenis metode yang berbeda.

TABEL IV
NILAI CEP

Data	AMPSO-BP	GA	BP
Iris Plant	82	100	100
Breast Cancer	34	66	66
Glass Identification	73	100	100
Wine	66	100	100
Pima Indian	100	100	100
Diabetes			

Berdasarkan jenis metode *training* terdapat dua jenis nilai *error* yaitu *error* global yang merupakan hasil *training global optimization* (PSO dan GA) dan *error* lokal yang merupakan hasil *training local optimization* (BP). BP memiliki nilai *error* minimum, tetapi hanya bersifat lokal minimum, berbeda dengan GA dan PSO memiliki nilai *error* minimum yang bersifat global minimum. Nilai *error* minimum yang bersifat global minimum memberikan nilai CEP yang lebih optimal dibandingkan dengan nilai *error* minimum yang bersifat lokal minimum. Nilai *error* yang diperoleh PSO dan GA memiliki sifat global minimum, tetapi PSO lebih cepat mencapai nilai *error* minimum (cepat konvergen) jika dibandingkan dengan GA.

D. Perbandingan Jumlah Iterasi Terhadap Error, Waktu dan CEP

Nilai *error training neural network* untuk data *iris plant* dengan jumlah iterasi yang berbeda dapat dilihat dalam Tabel 5.

TABEL V
ITERASI TERHADAP EROR TRAINING

Iterasi	AMPSO-BP	GA	BP
10	9.4×10^{-14}	11.119	0.3307
30	0.3333	8.8295	0.3300
50	0.0042	12.834	0.3297

Proses *training* yang melibatkan proses *global optimization* (AMPSO-BP dan GA), semakin banyak iterasi belum tentu memperoleh nilai *error* semakin kecil. Hal ini disebabkan proses *global optimization* melibatkan proses yang bersifat *random* atau acak. Setiap kali melakukan proses *training*, hasil *training* pertama dengan selanjutnya kemungkinan berbeda walaupun dalam jumlah iterasi yang sama. Berbeda dengan BP, dengan proses *training* yang berbeda, semakin banyak iterasi maka nilai *error* semakin kecil. Dapat disimpulkan bahwa proses metode *local optimization* (BP) bersifat lebih stabil dibandingkan dengan *global optimization*.

Waktu *training neural network* untuk data *iris plant* dengan jumlah iterasi yang berbeda dapat dilihat dalam Tabel 6.

TABEL VI
ITERASI TERHADAP WAKTU TRAINING

Iterasi	AMPSO-BP	GA	BP
10	57 m 50 s	50 m 35 s	131 m 34 s
30	291 m 51 s	133 m 35 s	383 m 11 s
50	545 m	273 m 5 s	681 m 16 s

Berdasarkan Tabel 6, untuk semua jenis metode, semakin banyak iterasi maka waktu yang dibutuhkan untuk proses *training* semakin lama.

Nilai CEP *training neural network* untuk data *iris plant* dengan jumlah iterasi yang berbeda dapat dilihat dalam Tabel 7.

TABEL VII
ITERASI TERHADAP NILAI CEP

Iterasi	AMPSO-BP	GA	BP
10	82	100	100
30	60	100	100
50	70	100	100

Berdasarkan Tabel 7, semakin banyak iterasi nilai CEP yang diperoleh belum tentu semakin lebih baik. Hal ini disebabkan nilai CEP dipengaruhi *error* global yang diperoleh dan nilai *error* global akan berubah-ubah untuk setiap proses disebabkan adanya proses *random*.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian dan analisis maka dapat diambil kesimpulan antara lain AMPSO-BP mampu

mengoptimalkan proses *training neural network* yaitu lebih cepat mencapai nilai *error* minimum yang bersifat global minimum dan *fast convergent* serta memiliki kemampuan memorisasi dan generalisasi yang optimum dengan hasil yang lebih akurat.

AMPSO-BP memiliki nilai *error training* paling baik jika dibandingkan dengan GA dan BP untuk semua jenis data. Dalam perhitungan nilai CEP, AMPSO-BP lebih baik dibandingkan dengan GA dan BP. Pengukuran waktu masing-masing metode berbeda-beda. Hal ini bergantung pada kecepatan dalam mencapai nilai *error* minimum, proses dari masing-masing metode, jumlah data *training* serta arsitektur *neural network*. Peningkatan jumlah iterasi tidak selalu memberikan penurunan nilai *error* untuk metode yang bersifat *global optimization* (PSO dan GA) akan tetapi mampu menurunkan nilai *error* pada metode yang bersifat *local optimization* (BP), selain itu peningkatan jumlah iterasi akan memberikan peningkatan waktu dalam proses *training*.

B. Saran

Saran yang mungkin perlu dilakukan dalam pengembangan metode AMPSO-BP untuk *training neural network* adalah penentuan arsitektur dengan melakukan uji rumus empiris memakan waktu terlalu lama karena terlalu banyak rumus empiris yang tersedia dan tidak semua rumus empiris bagus di semua jenis data. Alangkah baiknya jika dalam menentukan arsitektur bisa secara dinamis menyesuaikan jenis data tanpa harus melakukan uji rumus empiris.

Mutasi yang hanya dilakukan pada *gbest* membutuhkan waktu yang lama untuk mencapai solusi optimum, oleh sebab itu perlu penambahan proses mutasi tidak hanya *gbest* akan tetapi dilakukan pula pada *pbest* dengan disertai optimasi parameter mutasi.

Proses *training* metode AMPSO-BP dengan alat uji yang telah dibuat yang melibatkan proses *multithread* membutuhkan waktu yang lama dengan jumlah data

training yang sangat banyak disertai arsitektur yang kompleks. Hal ini dikarenakan tidak terdapat fungsi optimasi memori untuk menampung proses *thread* yang terlalu banyak. Oleh sebab itu perlu dilakukan optimasi pada sisi hardware (optimasi pemakaian memori) dalam pembuatan alat uji metode.

DAFTAR PUSTAKA

- [1] D. Puspitaningrum , “Pengantar Jaringan Syaraf Tiruan,” Yogyakarta: Andi, 2006.
- [2] Z.G. Chee, T.A. Chiang, and Z.H.Chie, “Feed Forward Neural Network : A Comparison Between Genetic Algorithm and Backpropagation Learning Algorithm,” ICIC International, 2011, vol 7, pp 5839-5850, ISSN:1349-4198.
- [3] F. Zhao, Z. Ren, and D. Yu, “Application of an Improvement Particle Swarm Optimization Algorithm for Neural Network,” IEEE, 2005, pp 1693-1698.
- [4] X.Cheng, D. Wang, K. Xie, and J. Zhang, “PSO Algorithm Combined with Neural Network Training Study,” IEEE, 2009, vol 9.
- [5] E.Andres, S.F. Marco, S.A. Jorge, O. Manuel, P.hector, C.Martin, B. Rosarion, and Rico, “Comparison of PSO and DE for Training Neural Network,” IEEE, 2011, vol 11, pp 83-87.
- [6] Suyanto, “Evolutionary Computation,” Bandung : Informatika, 2008.
- [7] N. Zhang and P.K. Bahera, “Solar Radiation Based on Particle Swarm Optimization and Evolutionary Using Recurrent and Neural Network,” IEEE, 2013, vol 13.
- [8] F. Ahmad, N.A.M. Isa, M.K. Osman, and Z.Hussain, “Performance Comparison of Gradient Descent and Genetic Algorithm Based Artificial Neural Network Training,” IEEE, 2010, vol 10, pp 604-609.
- [9] T. Sutojo, E. Mulyanto, And V. Suhartono, “Kecerdasan Buatan,” Yogyakarta:Andi, 2010.
- [10] Suyanto, “Algoritma Optimasi Deterministik dan Probabilistik,” Yogyakarta:Graha Ilmu, 2010.
- [11] J. Tang and X. Zhao, , “Particle Swarm Optimization with Adaptive Mutation,” IEEE, 2009,pp 234-237, doi:10.1109/ICIE.2009.53
- [12] UCI Machine Learning Repository, 2014, Dikutip 18 Mei 2014 dari <http://archive.ics.uci.edu/ml/datasets>.
- [13] Wiharto, Y.S. Pagulnadi, and M.A. Nugroho , “Analisis Penggunaan Algoritma Genetika untuk Perbaikan Jaringan Syaraf Tiruan Radial Basis Function,” SENTIKA, 2013,pp 181-188, ISSN:2089-9815.
- [14] L. Tarrasenko , “A Guide to Neural Computing Application,” New York:Elsevier, 1998.
- [15] W. Kosink, “Advances in Evolutionary Algorithm,” United Kingdom:Intech, 2008