

Classification of Wine Types Based on Composition Using Backpropagation And Particle Swarm Optimization

Kelvin Herwanda Tandrio, Kartini Aprilia Pratiwi Nuzry, Yovi Prasetyo Ardi, Heri Prasetyo

Department of Informatics
Sebelas Maret University (UNS)
Surakarta, Indonesia

kelvin_ht@student.uns.ac.id, kartiniaprilia21@student.uns.ac.id, yoviprasetyo@student.uns.ac.id,
heri.prasetyo@staff.uns.ac.id

Abstract— This paper presents a technique for classifying types of wine using Neural Network Back Propagation (NNBP). In this research also developed Neural Network technique with Particle Swarm Optimization (PSO) algorithm to calculate neural weight and bias. In the experimental results, the use of PSO-based NNBP or Neural Network Particle Swarm Optimization (NNPSO) yielded better results than the classic NNBP method. This clearly shows that the classification of types of wine with NNPSO can classify all types of wine. Meanwhile, NNBP techniques can only classify one type of wine. With the Mean Square Error (MSE) limit of 0.2, the accuracy obtained with NNPSO is about 75% - 90% and NNBP around 25 - 45%.

Keywords—Particle Swarm Optimization, Backpropagation, Classification, Wine

I. PENDAHULUAN

Anggur atau wine merupakan minuman beralkohol yang terbuat dari fermentasi anggur atau buah – buahan yang lain. Minuman ini dapat berfermentasi tanpa tambahan gula, asam, enzim, air atau nutrisi [1]. *Wine* mempunyai kadar alkohol sebesar 8 – 15%. Kualitas *wine* dapat ditentukan berdasarkan komposisi bahan baku, proses fermentasi, dan beberapa perubahan yang terjadi secara alami maupun disengaja selama proses fermentasi. Minuman ini juga melibatkan proses biokimiawi yang kompleks dan peran sejumlah enzim dari beberapa mikroorganisme yang secara alami hidup dalam buah – buahan yang berperan sebagai pembentuk etanol [2].

Wine banyak dikonsumsi oleh warga masyarakat. Minuman ini mempunyai banyak jenis. Karena *wine* sangat banyak jenis, masyarakat mengalami kesulitan dalam menentukan jenis *wine* yang dikonsumsi karena setiap *wine* memiliki kandungan yang berbeda. Oleh karena itu, peneliti melakukan klasifikasi jenis *wine* dilihat dari kadar kandungan dalam minuman ini. Proses klasifikasi dapat dilakukan dengan berbagai metode seperti k-NN, Naive Bayes, Artificial Neural Network, dan lain – lain. Selain itu, Teknik klasifikasi juga menggunakan metode *Evolutionary Computation*. Salah satunya adalah *Particle Swarm Optimization*. Teknik ini merupakan perkembangan algoritma untuk klasifikasi data yang lebih efisien.

Penelitian klasifikasi data sudah dilakukan oleh beberapa peneliti. Salah satu contoh adalah prediksi curah hujan menggunakan model jaringan *Backpropagation* yang dilakukan oleh Vamsidhar [3]. Model ini dapat memprediksi curah hujan dengan menggunakan tiga paramter. Ketiga parameter tersebut adalah tekanan, kelembaban, dan titik embun. Hasil akurasi yang diperoleh adalah 94,28%. Contoh penelitian lain adalah klasifikasi deteksi penyakit jantung dengan sembilan parameter menggunakan Naive Bayes berbasis Particle Swarm

Optimizarion yang dilakukan oleh Widiastuti [4]. Penelitian ini akan membandingkan hasil akurasi antara metode Naive Bayes dengan Naive Bayes berbasis Particle Swarm Optimization (PSO). Hasil akurasi yang diperoleh adalah 82,14% untuk Naive Bayes dan 92,86% untuk Naive Bayes berbasis PSO. Pengembangan klasifikasi data yang lain adalah penggunaan algoritma PSO untuk optimasi nilai *Center Radial Basis Probabilistic Neural Network* (RBPNN) pada klasifikasi data kanker payudara yang dilakukan oleh Nurmalasari [5]. Penelitian ini akan menentukan nilai *Center Radial* menggunakan algoritma PSO yang digunakan untuk proses kalsifikasi dengan metode RBPNN. Hasil akurasi yang diperoleh adalah 89,163% untuk data *training* dan 88,707% untuk data *testing*.

Dalam penelitian ini akan mengklasifikasikan jenis minuman *wine* berdasarkan kandungan atau komposisi yang dimilikinya. Proses klasifikasi ini menggunakan metode jaringan *Backpropagation* (NNBP) dan jaringan *Backpropagation* yang dimodifikasi dalam bentuk *Particle Swarm Optimzation* (NNPSO). Kedua metode ini akan membandingkan hasil akurasi pada data pengujian.

II. LANDASAN TEORI

A. Wine

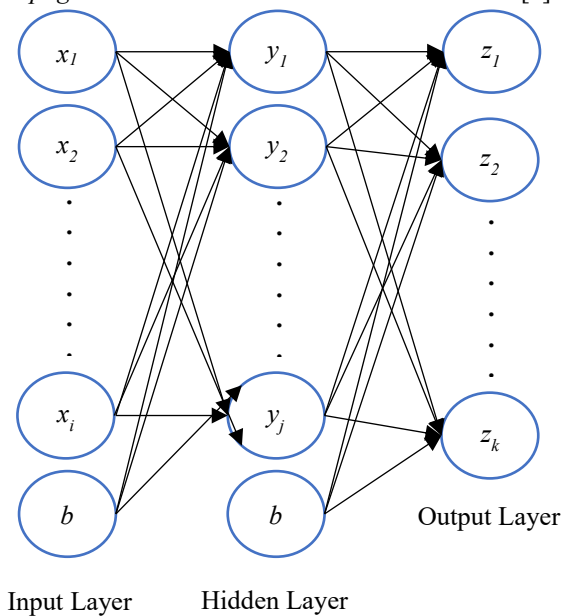
Wine merupakan minuman hasil fermentasi dari anggur yang sudah dikenal sejak sekitar 6000 tahun SM. Minuman ini berasal dari daerah Mesopotamia yang kemudian menyebar di berbagai negara termasuk negara Indonesia. Hal ini terbukti pada tahun 2004 jumlah produksi *wine* di Indonesia sebesar 2-3 miliar botol per tahun [2].

Pada umumnya, *wine* terbuat dari anggur yang difermentasikan. Selain itu, minuman *wine* juga dapat dibuat dari buah – buahan lain (seperti apel, stroberi, dan lain – lain)

dan bahan lainnya (seperti singkong, tape, dan lain – lain). Jenis *wine* yang terbuat dari sari buah biasanya mengikuti nama buah yang digunakan, seperti *wine* apel, *wine* berry dan secara umum disebut *fruity wine*, sedangkan *wine* yang terbuat dari bahan pangan yang mengandung pati seperti beras, singkong, maka lebih dikenal dengan istilah *Sake*. Minuman *wine* yang terbuat dari jahe dikenal dengan nama *Brandy*[6].

B. Backpropagation (NNBP)

Algoritma *backpropagation* merupakan metode sistematis pada model jaringan syaraf tiruan yang terdapat *hidden layer* dan berjalan secara maju dan mundur saat proses pembelajaran. Algoritma ini juga merupakan *learning supervised algorithm* karena menggunakan *gradien descent* untuk melakukan perubahan nilai bobot dalam model jaringan ini [7]. Penggunaan algoritma *gradien descent* sebagai teknik optimasi untuk menentukan nilai *Mean Squard Error* minimum dan menyesuaikan nilai bobot di *hidden layer* dan *output layer*. Selain itu, teknik ini juga menentukan algoritma pelatihan *backpropagation* terbaik dalam hal akurasi hasil akhir [8].



Gambar 1 Model Arsitektur Backpropagation

Perkembangan model jaringan runtun balik atau *backpropagation* menjadi perkembangan jaringan baru dalam menyelesaikan masalah yang lebih rumit. Hal ini dilihat bahwa model jaringan ini terdapat perubahan bobot dengan arah maju dan mundur menggunakan perubahan *gradien descent*. Perubahan ini biasanya digunakan pada fungsi keluaran sigmoid [9]. Dilihat dari model jaringan gambar 1, penentuan nilai *hidden layer* dan *output layer* dapat dibentuk pada persamaan (1) dan (2) [10].

$$y_{net_j} = \sum_{i=1}^n x_i \cdot w_{ij} + b_j \quad (1)$$

$$z_{net_k} = \sum_{j=1}^n y_j \cdot v_{jk} + b_k \quad (2)$$

Bentuk fungsi sigmoid yang digunakan pada umumnya menggunakan persamaan

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

Karena menggunakan fungsi sigmoid, maka kedua layar tersebut dibuat dalam persamaan (4) dan (5).

$$y_j = f(y_{net_j}) = \frac{1}{1+e^{-\theta(y_{net_j})}} \quad (4)$$

$$z_k = f(z_{net_k}) = \frac{1}{1+e^{-\theta(z_{net_k})}} \quad (5)$$

Dalam model NNBP, karena menggunakan teknik differensial, maka bentuk persamaan fungsi sigmoid diperoleh persamaan :

$$y' = f'(x) = \theta f[1 - f(x)] \quad (6)$$

Maka diperoleh faktor kesalahan pada jaringan keluaran (δ_k) dan jaringan tersembunyi (δ_j) dengan persamaan :

$$\delta_k = (t_k - z_k) \cdot z_k \cdot (1 - z_k) \quad (7)$$

$$\delta_j = \left(\sum_{k=1}^m \delta_k \cdot v_{jk} \right) \cdot y_j \cdot (1 - y_j) \quad (8)$$

Perubahan nilai bobot dalam model ini menggunakan nilai *learning state* (α). Maka, persamaan yang diperoleh untuk perubahan bobot x_{ij} dan y_{jk} adalah

$$v_{jk}(\text{baru}) = v_{jk}(\text{lama}) + \alpha \delta_k z_k \quad (9)$$

$$w_{ij}(\text{baru}) = w_{ij}(\text{lama}) + \alpha \delta_j y_j \quad (10)$$

C. Particle Swam Optimization

Algoritma *Particle Swarm Optimization* (PSO) merupakan teknik optimasi stokastik berbasis populasi yang diperkenalkan oleh Dr. Eberhart dan Dr. Kennedy pada tahun 1995. Algoritma ini terbentuk dari simulasi tingkah laku sosial pada kawanan burung yang terbang sangat cepat tanpa bertabrakan satu sama lain. Burung memiliki kecerdasan yang luar biasa karena mampu mengatur antara jarak dengan kecepatan terbang. Oleh karena itu, algoritma ini dapat digunakan untuk menyelesaikan masalah beberapa optimasi seperti optimasi tidak dibatasi, optimasi terbatas, optimasi multi-tujuan, dan lain – lain [11].

Algoritma PSO adalah salah satu dari teknik *Evolutionary Computation* yang dimana populasi pada n -dimensi akan menentukan lokasi yang terbaik (paling optimal) dalam suatu kasus. Selain itu, algoritma optimasi ini memiliki komputasi yang lebih efisien dibandingkan dengan algoritma optimasi lainnya [12]. Algoritma ini adalah salah satu dari teknik komputasi evolusioner yang mana populasi pada PSO didasarkan pada penelusuran algoritma dan diawali dengan suatu populasi yang random yang disebut dengan *particle*. Populasi dibangkitkan secara acak dengan batasan nilai terkecil dan terbesar. Bentuk algoritma PSO ini dapat dilihat pada gambar 2 [12] [13] [14].

Bentuk formulasi matematika yang menggambarkan posisi dan kecepatan partikel suatu n -dimensi dapat dibentuk pada persamaan (11) dan (12) [14]

$$X_i(t) = x_{i1}(t), x_{i2}(t), x_{i3}(t), \dots, x_{iN}(t) \quad (11)$$

$$V_i(t) = v_{i1}(t), v_{i2}(t), v_{i3}(t), \dots, v_{iN}(t) \quad (12)$$

Dengan persamaan ini dapat menentukan nilai fitness, nilai *minimization function* (*local best*), dan optimum values

(*global best*) dengan model persamaan yang dibentuk [14]. Bentuk algoritma PSO dapat dilihat pada gambar 2 [15].

```
// PSO'S ALGORITHM
for n particle
  inialitation particle
end
repeat
  for n particle
    determine fitness every particle
    if ( fitness[ i + 1 ] < fitness[ i ] )
      fitness[ i + 1 ] = fitness[ i ]
    end
  end
  Choose all particle for get best fitness
  for n particle
     $v_{i+1} = w \cdot v_i + \phi_1 r_1 (p_i - x_i) + \phi_2 r_2 (p_g - x_i)$ 
     $x_{i+1} = x_i + v_{i+1}$ 
  end
End
until ( toleran = true )
```

Gambar 2 Bentuk Source Code Algoritma PSO

Setiap partikel akan memperoleh nilai dari fungsi *fitness* yang dibentuk. Algoritma PSO akan menentukan posisi partikel p_i terbaik (p_{best}) antara posisi partikel sekarang dengan posisi partikel sebelumnya dilihat dari nilai *fitness*. Seluruh partikel yang sudah memperoleh nilai *fitness* akan ditentukan nilai global p_g terbaik (g_{best}) pada setiap iterasi yang dilakukan. Iterasi dalam algoritma ini akan menentukan nilai *fitness* terbaik untuk memperoleh hasil yang lebih optimal lagi [15].

Dalam algoritma PSO terdapat batas posisi dan kecepatan partikel. Batas posisi partikel dimulai dari x_{min} sampai x_{max} dan batas kecepatan partikel dimulai dari v_{min} sampai v_{max} . Algoritma ini dimulai dari posisi dan kecepatan awal partikel dengan persamaan :

$$x(0) = x_{min} + r(x_{max} - x_{min}) \quad (13)$$

$$v(0) = r \frac{(x_{max} - x_{min})}{20} \quad (14)$$

dimana r merupakan bilangan random.

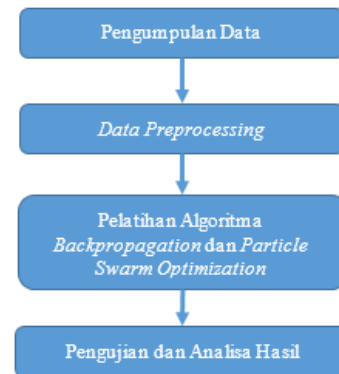
Perubahan posisi dan kecepatan partikel dalam algoritma PSO memiliki batas penentuan agar kedua nilai tersebut tidak melebihi batas yang sudah ditentukan. Ketentuan perubahan posisi dan kecepatan partikel dapat dilihat pada persamaan (15) dan (16) [17].

$$x(t+1) = \begin{cases} x_{min} & , \text{ if } x(t+1) < x_{min} \\ x_{max} & , \text{ if } x(t+1) > x_{max} \end{cases} \quad (15)$$

$$v(t+1) = \begin{cases} v_{max} & , \text{ if } v(t+1) > v_{max} \\ v(t+1) & , \text{ if } v(t+1) \leq v_{max} \end{cases} \quad (16)$$

III. METODOLOGI PENELITIAN

Adapun tahap – tahap penelitian ini dapat digambarkan pada gambar 3.



Gambar 3 Diagram Alur Metodologi Penelitian

Dari tahap – tahap penelitian di atas dapat diuraikan sebagai berikut :

3.1. Pengumpulan Data

Data *wine* yang digunakan dalam penelitian ini adalah data dari UCI Repository. Data ini memiliki 13 atribut yaitu *alcohol, malic acid, ash, alcalinity, magnesium, total phenols, flavanoids, nonflavanoid, proathocya, color intensity, hue, od280 of diluted wings* dan *proline*. Data yang diteliti terdiri dari 50 data tipe wine 1, 70 data tipe wine 2, dan 40 data tipe wine 3. Ketiga data tersebut akan diambil 5 data tipe wine I, 7 data tipe wine II, dan 4 data tipe wine III digunakan untuk data pengujian. Sedangkan, sisanya digunakan untuk data pelatihan dengan kedua metode ini.

3.2. Data Preprocessing

Pada bagian ini akan dilakukan *preprocessing* pada data *wine*. Tiap atribut data *wine* akan dibagi 1000. Hal ini bertujuan agar nilai yang diperoleh tidak mendekati nol pada saat perhitungan fungsi sigmoid dalam menentukan nilai *hidden layer* dan *output layer*. Nilai atribut pada data *wine* digunakan sebagai *input layer* pada model NNBP dan NNPSO.

3.3. Pelatihan Algoritma Backpropagation (NNBP) dan Particle Swarm Optimization (NNPSO)

Pelatihan data *training* yang sudah di-*preprocessing* diimplementasikan dalam bahasa pemrograman Java. Proses ini akan dibatasi dengan jumlah iterasi sebanyak 1000 iterasi dan besar MSE adalah 0,2.

Pelatihan data dengan metode NNBP dapat ditentukan dengan menentukan tiga nilai *output layer* pada model jaringan *backpropagation* dengan nilai bobot secara *random*. Dalam metode ini, nilai *learning state* yang digunakan adalah $\alpha = 0,1$. Hasil tersebut akan dihitung nilai *Mean Square Error* (MSE) dengan persamaan [16] :

$$MSE = \frac{1}{n} \sum_{k=1}^n (z_t - z_k)^2 \quad (17)$$

Perhitungan MSE ini digunakan untuk mengetahui apakah nilai tersebut sudah mencapai nilai toleransi MSE atau belum. Jika sudah mencapai nilai toleransi, maka proses *training* berhenti.

Tetapi, jika belum, maka nilai bobot dalam model jaringan NNBP akan *update* dengan menggunakan gradien differensial. Perubahan nilai bobot pada x_{ij} (nilai bobot dari *input layer* ke *hidden layer*) dan y_{jk} (nilai bobot dari *hidden layer* ke *output layer*) menggunakan persamaan 5 dan 6. Setelah itu, mengulang kembali untuk menentukan nilai *output layer* dan MSE sampai mencapai nilai toleransi.

Dalam penelitian ini, algoritma NNBP akan dimodifikasi ke dalam bentuk algoritma *Particle Swarm Optimization* (model NNPSO). Model ini akan menentukan nilai MSE minimum dan proses pembaharuan nilai bobot menggunakan teknik PSO. Dalam teknik ini akan dibuat beberapa partikel yang akan ditentukan nilai fitness terbaik pada tiap iterasi. Pada penelitian ini, banyak partikel yang dipakai adalah 1000 partikel dengan batas *weight* [-20, 20] dan *velocity* [-5, 5]. Teknik ini juga menggunakan *learning rate* untuk kecerdasan individu (φ_1) dan hubungan sosial antarindividu (φ_2). Nilai *learning rate* yang digunakan adalah $\varphi_1 = \varphi_2 = 2$. Pada teknik ini, proses iterasi akan berhenti ketika sudah memenuhi batas toleransi MSE berdasarkan nilai *fitness* terbaik yang diperoleh.

3.4. Pengujian dan Analisa Hasil

Nilai bobot dalam *neural network* dari proses pelatihan data akan diuji dengan data *testing*. Proses pengujian data akan

dihitung akurasi dengan nilai rata – rata *cross validation*. Pengujian ini akan diuji sebanyak 10 *cross validation*. Perhitungan akurasi akan diperoleh dengan persamaan :

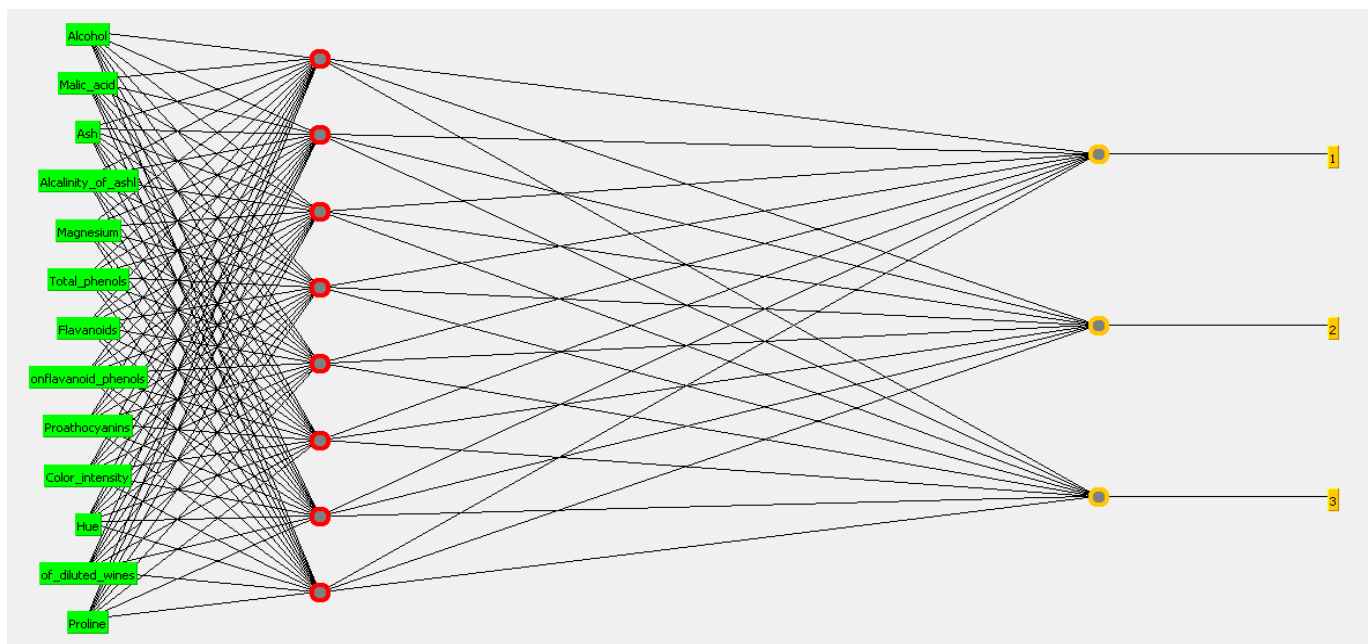
$$accuracy = \frac{\sum_{n=1}^{10} Cross_n}{10} \quad (18)$$

Penentuan akurasi ini juga menggunakan *hidden layer* dengan jumlah dari 1 sampai 10 *hidden layer* yang akan diuji sebanyak 5 kali. Pengujian data *wine testing* sebanyak 16 data yang terdiri dari 5 data jenis I, 7 data jenis II, dan 4 data jenis III. Pengujian data dengan kedua metode dibatasi 1000 iterasi.

IV. HASIL DAN PEMBAHASAN

4.1. Model Neural Network

Proses klasifikasi jenis *wine* dapat dibuat model *neural network* seperti pada gambar 4. Model tersebut masing – masing punya nilai bobot dan bias sendiri pada jaringan *input layer* ke *hidden layer* dan *hidden layer* ke *output layer*. Penentuan bobot dan bias dengan algoritma *Backpropagation* (NNBP) dapat ditentukan berdasarkan nilai MSE pada *output layer*. Sedangkan teknik NNPSO akan dilihat berdasarkan nilai fitness (nilai MSE) yang sudah mencapai targetnya.



Gambar 4 Model Jaringan *Backpropagation* Dalam Proses Klasifikasi Data *Wine*

4.2. Hasil Pelatihan

Hasil pelatihan data *wine* dengan teknik NNBP dan NNPSO memiliki hasil yang berbeda. Pelatihan dengan teknik NNPSO lebih baik daripada teknik NNBP. Hal ini dapat dilihat dari segi kestabilan dalam penentuan optimal dan tingkat akurasi yang diperoleh. Pada teknik NNPSO, proses pelatihan lebih stabil daripada teknik NNBP. Pada gambar 5 menunjukkan proses iterasi antara teknik NNBP dengan NNPSO pada *neural network* dengan dua *hidden layer* pada *cross* ke 4. Iterasi pada teknik NNBP terlihat tidak konvergen atau divergen untuk menuju nilai

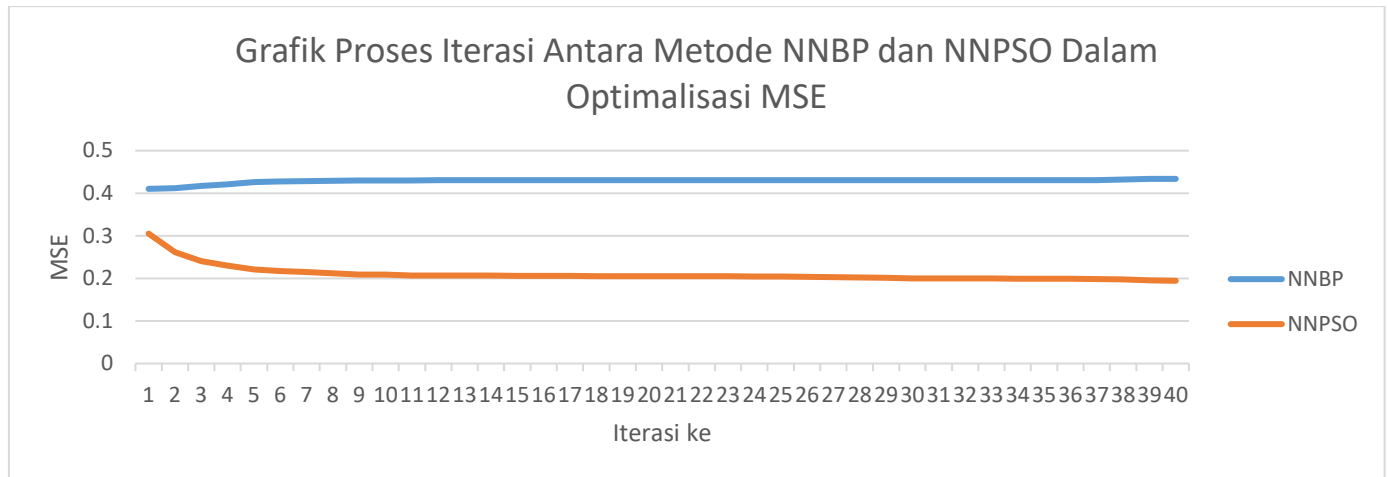
batas MSE yang ditentukan. Sedangkan, iterasi dengan NNPSO terlihat konvergen, stabil, dan dapat mencapai batas iterasi MSE dengan cepat. Teknik NNPSO membutuhkan sekitar 10 – 30 iterasi untuk mencapai batas toleransi. Sedangkan, teknik NNBP membutuhkan 100 sampai 1000 untuk mencapainya.

4.3. Hasil Uji Data

Hasil pengujian data *wine* dengan teknik NNBP dan NNPSO dapat dilihat pada tabel 1 dan 2. Hasil pengujian menunjukkan nilai akurasi pada teknik NNPSO lebih baik dari teknik NNBP.

Besar akurasi pada teknik NNBP sekitar 25 – 45% sedangkan akurasi pada teknik NNPSO sekitar 75 – 100%. Pada tabel 1 dan 2, nilai akurasi yang terbaik adalah teknik NNPSO dengan 9 jumlah *hidden layer* dengan nilai rata – rata akurasi sebesar 84%. Pada teknik NNPSO memiliki kestabilan yang tinggi dalam proses iterasi. Hal ini dapat dilihat pada gambar 8. Iterasi teknik NNBP tidak stabil dan divergen. Hal ini dikarenakan teknik

NNBP menggunakan teknik *gradient descent* dalam pembaharuan bobot dalam model jaringan. Tetapi sebaliknya, teknik NNPSO lebih melihat kondisi nilai MSE sehingga proses iterasi dengan teknik ini efisien, stabil, dan konvergen. Maka, teknik NNPSO juga mengoptimalkan nilai MSE dengan cepat dan mencapai batas toleransi yang ditentukan.



Gambar 5 Proses Iterasi Antara Metode NNBP dan NNPSO Dalam Optimalisasi *Measn Square Error*

Tabel 1 : Hasil Data *Wine Testing* Dengan Algoritma NNBP

Cross ke-	1	2	3	4	5	6	7	8	9	10
1	37,5%	41,25%	37,5%	42,5%	31,25%	35%	31,25%	31,25%	33,75%	33,75%
2	28,75%	43,75%	37,5%	45%	38,75%	27,5%	37,5%	38,75%	35%	37,5%
3	33,75%	36,25%	33,75%	36,25%	40%	40%	51,25%	28,75%	33,75%	31,25%
4	36,25%	33,75%	30%	41,25%	35%	37,5%	32,5%	33,75%	33,75%	45%
5	36,25%	35%	36,25%	40%	41,25%	38,75%	35%	30%	45%	40%
6	30%	38,75%	41,25%	35%	35%	36,25%	41,25%	30%	33,75%	32,5%
7	37,5%	33,75%	35%	40%	31,25%	35%	36,25%	38,75%	32,5%	35%
8	31,25%	35%	33,75%	36,25%	43,75%	41,25%	36,25%	46,25%	38,75%	38,75%
9	33,75%	38,75%	33,75%	36,25%	36,25%	42,5%	38,75%	46,25%	35%	41,25%
10	33,75%	46,25%	46,25%	38,75%	41,25%	38,75%	46,25%	37,5%	43,75%	31,25%
Average	33,88%	38,25%	36,50%	39,13%	37,38%	37,25%	38,63%	36,13%	36,50%	36,63%

Tabel 2 : Hasil Data *Wine Testing* Dengan Algoritma NNPSO

Cross ke-	1	2	3	4	5	6	7	8	9	10
1	75%	77,5%	80%	78,75%	75%	78,75%	77,5%	82,5%	90%	82,5%
2	75%	75%	75%	81,25%	75%	93,75%	75%	75%	81,25%	75%
3	75%	75%	75%	75%	75%	78,75%	77,5%	82,25%	93,75%	75%
4	75%	75%	75%	87,5%	75%	87,5%	75%	87,5%	77,5%	75%
5	75%	75%	75%	78,75%	75%	75%	80%	78,75%	78,75%	80%
6	75%	75%	75%	75%	75%	75%	87,5%	75%	75%	75%
7	75%	76,25%	76,25%	76,25%	75%	75%	75%	78,75%	82,5%	76,25%
8	75%	76,25%	76,25%	76,25%	77,5%	78,75%	82,5%	77,5%	80%	87,5%
9	75%	75%	82,25%	76,25%	75%	75%	75%	86,25%	87,5%	75%
10	75%	75%	75%	76,25%	76,25%	80%	82,5%	75%	96,25%	81,25%
Average	75%	76%	76%	78%	75%	80%	79%	80%	84%	78%

V. KESIMPULAN

Berdasarkan penelitian dapat mengklasifikasikan tipe *wine* dengan teknik NNPSO dan NNBP. Jika dibandingkan, teknik NNPSO diperoleh lebih baik dari teknik NNBP. Hasil penelitian ini dapat dilihat pada tabel 1 dan 2. Akurasi yang diperoleh dengan teknik NNBP sekitar 25-45%. Setelah menggabungkan dengan teknik PSO, maka akurasi yang diperoleh sekitar 75-100%. Selain itu, teknik NNPSO dapat melatih data training dengan cepat, sangat stabil, dan konvergen daripada teknik NNBP. Hal ini dilihat bahwa dengan teknik NNPSO, nilai bobot yang diperoleh akan ditentukan berdasarkan nilai fitness global yang diperoleh. Sedangkan teknik NNBP melihat dari nilai *delta gradien* sampai nilai MSE terpenuhi dilihat dari nilai output layer. Oleh karena itu, waktu yang dibutuhkan pada teknik NNBP lama untuk mencapai optimal.

DAFTAR PUSTAKA

- [1] Ariyanto, H. D., Hidayatulloh, F., & Murwono, J. (2013). Pengaruh penambahan gula terhadap produktivitas alkohol dalam pembuatan wine berbahan apel buang (Reject) dengan menggunakan Nopkor MZ. 11. *Jurnal Teknologi Kimia dan Industri*, 2(4), 226-232.
- [2] Hawusiwa, E. S., Wardani, A. K., & Ningtyas, D. W. (2014). PENGARUH KONSENTRASI PASTA SINGKONG (Manihot esculenta) DAN LAMA FERMENTASI PADA PROSES PEMBUATAN MINUMAN WINE SINGKONG [IN PRESS JANUARI 2015]. *Jurnal Pangan dan Agroindustri*, 3(1), 147-155.
- [3] Vamsidhar, E., Varma, K. V. S. R. P., Rao, P. S., & Satapati, R. (2010). Prediction of rainfall using backpropagation neural network model. *International Journal on Computer Science and Engineering*, 2(4), 1119-1121.
- [4] Widiastuti, N. A., Santosa, S., & Supriyanto, C. (2014). Algoritma Klasifikasi data mining naïve bayes berbasis Particle Swarm Optimization untuk deteksi penyakit jantung. *Jurnal Pseudocode*, 1(1), 11-14..
- [5] Nurmalasari, E. (2017). ALGORITMA PARTICLE SWARM OPTIMIZATION (PSO) UNTUK OPTIMASI NILAI CENTER RADIAL BASIS PROBABILISTIC NEURAL NETWORK (RBPNN) PADA KLASIFIKASI DATA BREAST CANCER. *Jurnal Elektronik Nasional Teknologi dan Ilmu Komputer*, 1(02)..
- [6] NANIK, Rollyana Ajeng Ovihapsany Akhmad Mustofa, et al. KARAKTERISTIK MINUMAN BERALKOHOL DENGAN VARIASI KADAR EKSTRAK BUAH BIT (Beta vulgaris L.) DAN LAMA FERMENTASI. *JITIPARI*, 2018, 5.3.
- [7] Hardinata, J. T., Zarlis, M., Nababan, E. B., Hartama, D., & Sembiring, R. W. (2017, December). Modification Of Learning Rate With Lvq Model Improvement In Learning Backpropagation. In *Journal of Physics: Conference Series* (Vol. 930, No. 1, p. 012025). IOP Publishing.
- [8] Chai, S. S., Veenendaal, B., West, G., & Walker, J. P. (2008). Backpropagation neural network for soil moisture retrieval using NAFE'05 data: a comparison of different training algorithms. *Int Archives Photogramm, Remote Sens Spatial Inf Sci (China)*, 37, 1345.
- [9] Widodo, P. P., & Handayanto, R. T. (2012). Penerapan Soft Computing Dengan Matlab. *Bandung: Rekayasa Sains*.
- [10] Siang, J. J. (2005). Jaringan syaraf tiruan dan pemrogramannya menggunakan Matlab. *Penerbit Andi, Yogyakarta*.
- [11] Jamous, R. A., Seidy, E., Tharwat, A. A., & Bayoum, B. I. (2015). Modifications of Particle Swarm Optimization Techniques and Its Application on Stock Market: A Survey. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 6(3)..
- [12] Lee, K. E., bin Abdul Aziz, I., & bin Jaafar, J. Adaptive Multilayered Particle Swarm Optimized Neural Network (AMPSONN) for Pipeline Corrosion Prediction.
- [13] Suyanto, 2014. Algoritma Optimasi Deterministik atau Probabilitik. Penerbit : Graha Ilmu. Yogyakarta
- [14] García-Gonzalo, E., & Fernández-Martínez, J. L. (2012). A brief historical review of particle swarm optimization (PSO). *Journal of Bioinformatics and Intelligent Control*, 1(1), 3-16
- [15] Eliantara, Felia., Imam Cholissodin dan Indriati. 2016. Optimasi Pemenuhan Kebutuhan Gizi Keluarga Menggunakan Particle Swarm Optimization.
- [16] Moghaddam, M. G., Ahmad, F. B. H., Basri, M., & Rahman, M. B. A. (2010). Artificial neural network modeling studies to predict the yield of enzymatic synthesis of betulinic acid ester. *Electronic Journal of Biotechnology*, 13(3), 3-4.
- [17] Bonyadi, M. R., & Michalewicz, Z. (2016). Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 20(3), 370-385.