

Particle Swarm Optimization Untuk Optimasi Bobot Extreme Learning Machine Dalam Memprediksi Produksi Gula Kristal Putih Pabrik Gula Candi Baru-Sidoarjo

Eka Yuni Darmayanti¹, Budi Darma Setiawan², Fitra Abdurrachman Bachtiar³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹ekayunidarmayanti23@gmail.com, ²s.budidarma@ub.ac.id, ³fitra.bachtiar@ub.ac.id

Abstrak

Permintaan gula akan meningkat seiring dengan peningkatan jumlah penduduk, pendapatan masyarakat, dan pertumbuhan industri pengolahan makanan dan minuman. Oleh karena itu agar proses produksi gula selalu meningkat sesuai dengan kebutuhan gula itu sendiri, maka diperlukannya perencanaan produksi. Peramalan yang akurat dapat membantu perusahaan dalam mengambil keputusan untuk menentukan jumlah gula yang akan diproduksi, bahan yang dibutuhkan dan menentukan harga barang. Salah satu metode yang dapat digunakan untuk melakukan prediksi adalah algoritme *Extreme Learning Machine*. Akan tetapi metode tersebut dalam pemilihan *input weight* dan bias dipilih secara acak, hal ini dapat menyebabkan hasil yang didapat dalam perhitungan kurang maksimal. Diperlukannya kombinasi dengan algoritme *Particle Swarm Optimization* yang dapat melakukan optimasi nilai *input weight* dan bias secara optimal. Penelitian ini menggunakan 45 data giling produksi gula dengan 5 fitur. Berdasarkan penelitian yang telah dilakukan, didapatkan parameter yang optimal yaitu jumlah ukuran populasi 50, perbandingan data training 80% (36 data), jumlah hidden neuron 10, bobot inersia 0.5, dan iterasi maksimal 250. dari parameter tersebut didapatkan nilai rata-rata MAPE sebesar 0.59%. Dari hasil rata-rata MAPE yang didapat, menunjukkan bahwa penambahan algoritme PSO pada ELM dapat menentukan nilai *input weight* dan bias yang optimal.

Kata kunci: prediksi, optimasi, *Extreme Learning Machine*, *Particle Swarm Optimization*

Abstract

Sugar demand will increase in line with the increase in population, income, and growth in food and beverage processing industry. Therefore, in order for the sugar production process is always increasing in accordance with needs of the sugar itself, hence need for production planning. Accurate forecasting can help companies in taking decisions to determine the amount of sugar to be produced, the materials needed and determine the price of the goods. One method that can be used to do the prediction algorithm is Extreme Learning Machine. But that method in a selection of input and weight bias is chosen randomly, this can lead to the results obtained in the calculation less maximum. This need for a combination of Particle Swarm Optimization algorithms that can perform optimization the input value weight and bias optimally. This research uses data 45 milled sugar production with 5 features. Based on the research that has been performed, the obtained optimal parameters, namely the number of population size 50, 80% training data comparison (36), the number of hidden neurons 10, weighs of inertia 0.5, and a maximum of iterations 250. The parameter value is obtained from the average MAPE of 0.59%. From the average MAPE results obtained, shows that the addition of the PSO algorithm on ELM can determine the value of the input of weight and optimal bias.

Keywords: prediction, optimization, *Extreme Learning Machin*), *Particle Swarm Optimization*

1. PENDAHULUAN

Dalam kegiatan memproduksi gula, PT Pabrik Gula Candi Baru sudah melakukan perencanaan produksi untuk dapat

mengoptimalkan strategi produksi. Perencanaan produksi adalah suatu kegiatan untuk menetapkan produk yang diproduksi diantaranya jumlah yang dibutuhkan, kapan produk harus selesai dan sumber-sumber yang dibutuhkan.

Namun perencanaan produksi ini masih dilakukan secara manual yang memungkinkan terjadinya kesalahan dalam perhitungannya (Siwi, et al., 2016). Peramalan yang akurat dan efektif dapat membantu pengambil keputusan dalam perusahaan untuk menentukan jumlah barang yang akan diproduksi, bahan baku yang dibutuhkan serta menentukan harga terhadap barang jadi sehingga perusahaan memiliki tingkat inventory rendah (Sun, et al., 2008).

ELM merupakan jaringan syaraf tiruan *feedforward* dengan satu *hidden layer* atau lebih dikenal dengan istilah *single hidden layer feedforward neural network* (SLNFs). Metode ELM sendiri mempunyai kelebihan dalam *learning speed*, serta mempunyai tingkat akurasi yang lebih baik dibandingkan dengan metode konvensional lainnya. Sehingga dengan diterapkannya metode ELM diharapkan mampu menghasilkan ramalan yang lebih efektif (Agustina, et al., 2010).

Penelitian yang dilakukan oleh Irwin Dwi Agustina (2010) yang berjudul “Penerapan Metode *Extreme Learning Machine* Untuk Peramalan Permintaan” dalam penelitian ini, peneliti melakukan peramalan permintaan terhadap perencanaan produksi toko “Cak Cuk Shop” pada penelitian ini menghasilkan tingkat kesalahan menggunakan MAPE yaitu sebesar 0.0042% pada produk kaos dan 0.0095% pada produk pin. Ada penelitian lain yang dilakukan oleh Iga Permata Siwi (2016) penelitian yang berjudul “Peramalan Produksi Gula Pasir Menggunakan ELM pada PG Candi Baru Sidoarjo” pada penelitian ini tingkat error yang juga dihitung dengan nilai MAPE sebesar 0,74 %.

Selain memiliki keuntungan ELM sendiri memiliki kelemahan yaitu, jumlah dari *hidden nodes* yang ada ditentukan dengan cara *try and error*, sehingga tidak dapat diketahui jumlah dari *hidden nodes* yang tepat untuk mendapatkan hasil peramalan yang tepat dengan menggunakan metode ELM. Selain permasalahan *hidden nodes* dalam melakukan pemilihan *input weight*, bias dalam metode ELM nilai tersebut dipilih secara acak atau *random*, maka hal ini dapat menyebabkan hasil yang didapat dalam perhitungan juga kurang maksimal (Handika, I P.S (2016)).

Penelitian yang menggunakan *Particle Swarm Optimization* (PSO) sebagai metode optimasi adalah penelitian dari Zhao, Zhang dan Han (2014). Pada penelitian ini peneliti

menggunakan metode ELM lalu mengembangkannya dengan *Adaptive Growth of Hidden Nodes* (AG) dan PSO. Pada penelitian ini peneliti menggunakan PSO untuk menghitung *input weight*, *output weight* dan bias yang selanjutnya digunakan pada AG-ELM. Peneliti menyimpulkan bahwa pengembangan PSO-AG-ELM ini memiliki tingkat akurasi yang tinggi jika dibandingkan dengan AG-ELM dan ELM biasanya. Penelitian yang juga menggunakan *hybrid algorithm* adalah penelitian dari Nur Afifah Sugianto (2017). Peneliti menggunakan metode ELM sebagai metode klasifikasi untuk pemilihan keminatan pada mahasiswa Informatika. Akan tetapi pada metode ELM tidak memiliki kemampuan untuk melakukan seleksi fitur sehingga dikombinasikan metode ELM dengan metode PSO sebagai metode untuk menyeleksi fitur yang digunakan agar dapat menyeleksi fitur dengan otomatis dan optimal. Penelitian ini menghasilkan nilai akurasi sebesar 94.44% dibandingkan dengan hanya menggunakan metode ELM biasa yakni sebesar 66,67%.

Oleh karena itu diperlukan metode optimasi untuk menghitung input weight dan bias yang akan digunakan pada peramalan dengan metode ELM. Berdasarkan permasalahan yang ada, dengan menggunakan *hybrid algorithm* ELM dan PSO dapat menghasilkan hasil yang lebih optimal dengan pembelajaran yang cepat. Sehingga pada penelitian kali ini peneliti mengusulkan penelitian tentang optimasi bobot *Extreme Learning Machine* dengan *Particle Swarm Optimization* dalam memprediksi produksi gula kristal putih agar dapat memberikan prediksi yang optimal dengan tingkat *error* yang rendah

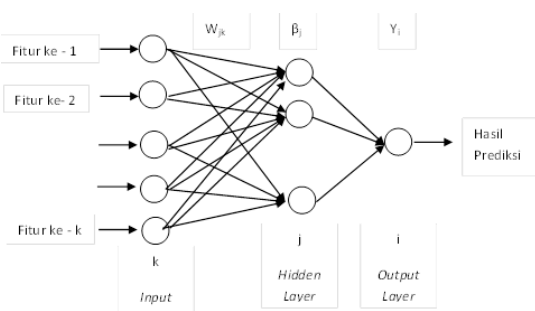
2. EXTREME LEARNING MACHINE (ELM)

Extreme Learning Machine merupakan metode pembelajaran baru dari jaringan syaraf tiruan. Metode ini pertama kali dikenalkan oleh Huang (2004). Metode ELM sendiri merupakan jaringan syaraf tiruan *feedforward* dengan *single hidden layer* atau dengan kata lain *Single Hidden Layer Feedforward Neural Networks* (SLFNs). Huang et al (2004) mengemukakan dua alasan mengapa JST *feedforward* lain mempunyai *learning speed* rendah, yaitu:

1. Menggunakan *slow gradient based learning algorithm* untuk melakukan *training*.

- Semua parameter pada jaringan ditentukan secara *iterative* dengan menggunakan metode pembelajaran tersebut.

Pada ELM parameter-parameter seperti input weight dan hidden bias dipilih secara random, sehingga ELM memiliki *learning speed* yang cepat dan mampu menghasilkan generalisasi dengan performa yang baik. Berikut adalah struktur dari metode ELM pada data (x_i, y_i) (Cholissodin, et al., 2017).



Gambar 1 Arsitektur ELM

Langkah – langkah proses pelatihan (*testing*) ELM adalah sebagai berikut:

- Inisialisasi *Input weight* (W_{jk}) dan Bias (b) secara acak.
- Menghitung Matriks Keluaran *Hidden Layer*

$$H = \frac{1}{(1 + \exp(-(X \cdot W^T + \text{ones}(N_{\text{train}}, 1) * b))} \quad (1)$$

Keterangan:

- H = Keluaran Hidden Layer
- Exp = Eksponensial
- X = Matriks Data Masukan (*Input*)
- W^T = Matriks *Transpose Input Weight*
- $\text{Ones}(N_{\text{train}}, 1)$ = Matriks yang bernilai 1 dengan ukuran baris sama dengan jumlah *training* dan 1 kolom
- B = Bias

- Menghitung Matriks *Moore-Penrose Pseudo Inverse*

$$H^+ = (H^T \times H)^{-1} \times H^T \quad (2)$$

Keterangan:

- H^+ = matriks *Moore – Penrose Pseudo Inverse*
- H = matriks keluaran *hidden layer*
- H^T = transpose matriks keluaran *hidden layer*

- Menghitung Output Weight ($\hat{\beta}$)

$$(\hat{\beta}) = H^+ \times Y \quad (3)$$

Keterangan:

- $\hat{\beta}$ = bobot keluaran *hidden layer*
- H^+ = matriks *Moore – Penrose Pseudo Inverse*
- Y = matriks data *output* atau target

- Menghitung hasil prediksi (\hat{Y})

$$(\hat{Y}) = H \times \hat{\beta} \quad (4)$$

Keterangan:

- \hat{Y} = hasil prediksi
- H = matriks keluaran *hidden layer*
- $\hat{\beta}$ = matriks bobot keluaran *hidden layer*

Langkah – langkah proses pengujian (*training*)

- Menggunakan *input weight* (W_{jk}) dan *output weight* ($\hat{\beta}$) yang didapatkan pada proses *training*
- Menghitung matriks keluaran *hidden layer* (H) menggunakan Persamaan 2.1 seperti pada proses *training*.
- Menghitung hasil prediksi (\hat{Y}) menggunakan persamaan 4 seperti pada proses *training*.
- Menghitung nilai evaluasi menggunakan MAPE.

3. PARTICLE SWARM OPTIMIZATION (PSO)

Algoritma PSO pertama kali diusulkan oleh J. Kennedy dan R. C Eberhart (Kennedy, 1995). *Particle Swarm Optimization* (PSO) merupakan sebuah metode optimasi yang didasarkan pada perilaku sebuah kawanan serangga misalnya semut, rayap, lebah atau burung. Suatu partikel dalam ruang memiliki posisi dan setiap posisi dalam ruang pencarian merupakan alternatif solusi yang dapat dievaluasi menggunakan fungsi objektif. Setiap partikel dapat menyesuaikan posisi dan kecepatan masing-masing dengan cara setiap partikel menyampaikan informasi terbaiknya kepada partikel yang lain. Oleh karena itu, setiap partikel memiliki kecenderungan untuk terbang menuju posisi yang dianggap terbaik Partikel sendiri adalah setiap individu dalam suatu kelompok (Handika, et al., 2016).

Langkah – langkah pada *Particle Swarm Optimization* (PSO) dapat diuraikan seperti berikut:

- Proses Inisialisasi
 - Inisialisasi Kecepatan Awal Partikel

Pada iterasi awal atau iterasi ke-0, nilai kecepatan awal semua partikel di atur dengan memberi nilai 0.

- b. Inisialisasi Posisi Awal Partikel
Pada iterasi awal atau iterasi ke-0, posisi awal partikel dibangkitkan secara acak atau *random*.
 - c. Menghitung Nilai *Fitness*
 - d. Inisialisasi Nilai *Pbest* dan *Gbest*
2. Update kecepatan Partikel

$$v_{i,j}^{t+1} = w \cdot v_{i,j}^t + c_1 \cdot r_1 (Pbest_{i,j}^t - x_{i,j}^t) + c_2 \cdot r_2 (Gbest_{g,j}^t - x_{i,j}^t) \quad (5)$$

Keterangan:

$v_{i,j}^{t+1}$ = kecepatan partikel *i* dimensi *j* pada iterasi

$v_{i,j}^t$ = kecepatan partikel *i* dimensi *j* pada iterasi *t* (*t* sebelumnya)

w = bobot inersia

c1 = konstanta kecepatan 1

c2 = konstanta kecepatan 2

r1, r2 = nilai acak $\in [0,1]$

$Pbest_{i,j}^t$ = posisi terbaik dari partikel *i* dimensi *j* pada iterasi *t*

$Gbest_{g,j}^t$ = global optimal dari partikel *g* dimensi *j* pada iterasi *t*

$x_{i,j}^t$ = posisi partikel *i* dimensi *j* pada iterasi *t*

3. Update Posisi Partikel

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad (6)$$

4. Update Nilai *Pbest* dan *Gbest*

Untuk mendapat nilai *Pbest* dapat dilakukan dengan cara membandingkan antara nilai *Pbest* pada iterasi sebelumnya dengan hasil dari perhitungan *update* posisi. Nilai *fitness* yang lebih rendah dari keduanya akan menjadi nilai *Pbest* yang baru. Sedangkan untuk mendapatkan nilai *Gbest* terbaru, dapat dilakukan perhitungan dengan cara melihat nilai *Pbest* yang memiliki nilai *fitness* yang paling rendah. Nilai *fitness* yang digunakan dalam penelitian ini didapat dari nilai MAPE pada perhitungan ELM sehingga dipilihlah nilai *fitness* terendah.

5. Mengulang langka ke 2-4 sampai kondisi berhenti terpenuhi (*Termination condition*) terpenuhi. Ada beberapa kondisi berhenti yang digunakan, diantaranya adalah:

- a. Ketika iterasi sudah mencapai *Maximum*
- b. Iterasi berhenti tetapi tidak ada perubahan yang signifikan
- c. Ketika telah mencapai waktu *Maximum*

4. *Extreme Learning Machine* (ELM) & *Particle Swarm Optimization* (PSO)

Pada umumnya *hybrid algorithm* PSO-ELM sering kali digunakan untuk 2 tujuan, yaitu seleksi fitur dan mendapatkan bias terbaik (Sugianto, et al., 2017). Akan tetapi pada penelitian kali ini, metode PSO yang peneliti gunakan adalah untuk mendapatkan nilai *input weight* dan bias yang optimal sehingga dapat menghasilkan peramalan yang lebih optimal juga. Pengkodean PSO yang peneliti gunakan adalah pengkodean *real-code* PSO.

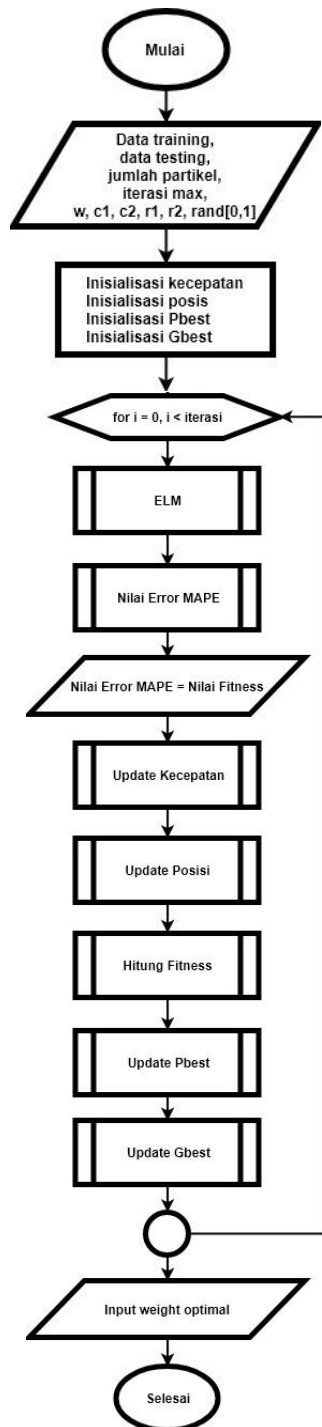
Langkah – langkah pengerjaan ELM dengan menggunakan optimasi PSO secara umum seperti berikut:

1. Menentukan panjang dimensi *particle* dan banyaknya iterasi maksimum yang akan digunakan. Panjang *particle* didapat dari fitur dikalikan jumlah *hidden neuron* dan ditambah dengan bias.
2. Inisialisasi kecepatan *particle*. Seperti pada langkah – langkah PSO pada proses inisialisasi diatas, bahwa pada iterasi ke-0, kecepatan *particle* bernilai 0.
3. Inisialisasi posisi *particle* secara acak dengan panjang dimensi *particle* seperti yang sudah dijelaskan pada langkah 1 diatas.
4. Melakukan proses *training* pada ELM dan melakukan proses *testing* seperti yang sudah dipaparkan dengan nilai input weight yang didapat dari PSO untuk mendapat nilai MAPE yang nantinya digunakan sebagai nilai *fitness* pada posisi partikel, *update* *Pbest* dan *Gbest*.
5. Melakukan langkah-langkah pengerjaan PSO seperti yang sudah dijelaskan pada bab nomor 3 hingga kondisi berhenti dan mendapatkan nilai *Gbest* dengan *fitness* yang paling rendah.

5. PERANCANGAN DAN IMPLEMENTASI

Siklus dari metode *hybrid algorithm* antara ELM dan PSO adalah suatu penyelesaian dalam menyelesaikan masalah yakni untuk mencari

nilai *input weight* dan bias yang optimal. Langkah-langkah *hybrid algorithm* ELM dan PSO adalah sebagai berikut:



Gambar 2 Diagram Alir ELM-PSO

Langkah-langkah yang dilakukan dalam implementasi PSO untuk optimasi bobot ELM dalam memprediksi produksi gula adalah sebagai berikut:

1. Inisialisasi parameter PSO yaitu, kecepatan awal, posisi awal partikel, pbest, gbest, jumlah ukuran populasi, w, c1, c2,

r1, r2, rand[0,1] . Pada penelitian ini inisialisasi partikel dilakukan secara *random* dengan range [-1,1] untuk *input weight* dan [0,1] untuk bias (PSO *real code*) dengan panjang partikel didapat dari perkalian antara fitur dengan jumlah *hidden neuron* dan ditambah dengan bias. Contoh representasi partikel pada penelitian ini ditunjukkan pada tabel 1.

Table 1 Representasi Partikel

No	P	1	2	...	18	Fitness
1	X1(0)	0.6	0.1	...	0.74	3.25

Pada tabel diatas dapat diketahui panjang partikel 18 yang terdiri dari 15 *input weight* dan 3 bias.

2. Inisialisasi parameter ELM yaitu data *training*, data *testing*, *input weight* dan bias. Inisialisasi *input weight* dan bias didapat dari posisi awal pada PSO secara *random* dengan *range* [-1,1] dan [0,1].
3. Menghitung nilai evaluasi dengan *Mean Absolute Percentage Error* (MAPE) dari hasil prediksi yang nantinya akan digunakan sebagai nilai *fitness* pada PSO menggunakan persamaan 7.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \left(\frac{y_i - \hat{y}_i}{y_i} \times 100 \right) \right| \quad (7)$$

Keterangan:

- n = jumlah data
- \hat{y}_i = nilai hasil ramalan
- y_i = nilai aktual

4. Melakukkann proses *update* kecepatan, *update* posisi, *update Pbest*, dan *update Gbest* untuk mendapat nilai yang optimal.

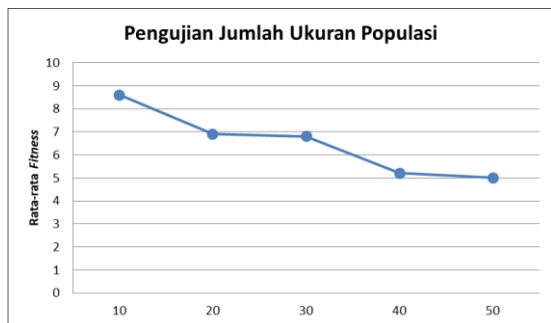
6. PENGUJIAN DAN ANALISIS HASIL

6.1 Pengujian Jumlah Ukuran Populasi

Tujuan dari pengujian jumlah ukuran populasi adalah untuk mengetahui pengaruh jumlah ukuran populasi terbaik untuk menghasilkan nilai *error* yang optimal. Banyaknya ukuran populasi yang diuji dimulai dari 5 dan kelipatannya sampai 100. Parameter yang digunakan pada pengujian ini yaitu:

- a. Jumlah Iterasi : 100
- b. Bobot Inersia : 0.5
- c. Nilai Konstanta 1 (c1) : 1
- d. Nilai Konstanta 2 (c2) : 1

- e. Jumlah *Hidden Neuron* : 3
- f. Banyak data *training* dan data *testing* : 80% : 20%



Gambar 1 Penguujian Ukuran Populasi

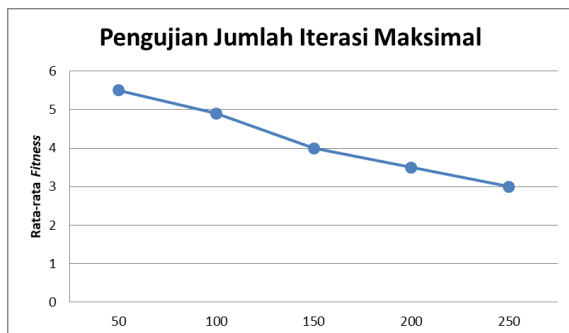
Berdasarkan gambar 3 hasil penguujian jumlah ukuran populasi di atas, ukuran populasi 50 mengasilkan rata-rata nilai *fitness* terkecil, yaitu 5.0. Dari gambar diatas dapat disimpulkan apabila jumlah ukuran populasi semakin besar maka nilai *fitness* semakin kecil, meskipun ada peningkatan pada beberapa titik dikarenakan nilai yang *random*.

6.2 Penguujian Jumlah Iterasi Maksimal

Penguujian jumlah iterasi maksimum bertujuan untuk mengetahui jumlah iterasi terbaik untuk mendapatkan hasil prediksi yang optimal. Penguujian jumlah iterasi dimulai dari angka kelipatan 50 hingga 250.

Parameter yang digunakan pada penguujian ini yaitu:

- a. Banyak data *training* dan *testing* : 80% : 20%
- b. Bobot Inersia : 0.5
- c. Nilai Konstanta 1 (*c1*) : 1
- d. Nilai Konstanta 2 (*c2*) : 1
- e. Jumlah *Hidden Neuron* : 3
- f. Jumlah ukuran populasi : 50



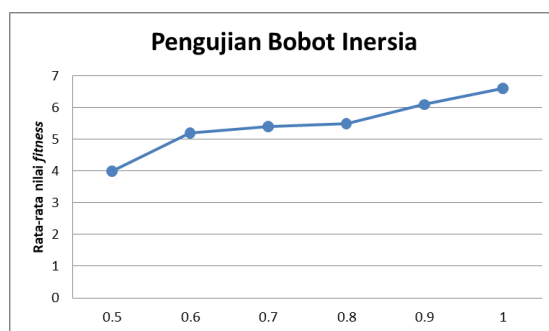
Gambar 2 Penguujian Jumlah Data *Training*

Dari gambar 4 dapat disimpulkan bahwa nilai rata-rata *fitness* juga dipengaruhi oleh banyaknya jumlah iterasi. Semakin banyak jumlah iterasi maka nilai *fitness* akan semakin menurun, hal ini dikarenakan banyaknya iterasi membuat pencarian solusi yang optimal lebih banyak.

6.3 Penguujian Bobot Inersia

Penguujian bobot inersia dilakukan untuk mengetahui bobot terbaik guna mendapatkan hasil prediksi yang optimal. Penguujian bobot inersia dimulai dari nilai 0.5 sampai 1. Parameter yang digunakan pada penguujian ini yaitu:

- a. Nilai Konstanta 1 (*c1*) : 1
- b. Nilai Konstanta 2 (*c2*) : 1
- c. Banyak data *training*: 80%
- d. Jumlah ukuran populasi : 50
- e. Jumlah *Hidden Neuron* : 3
- f. Jumlah Iterasi : 250



Gambar 5 Penguujian Bobot Inersia

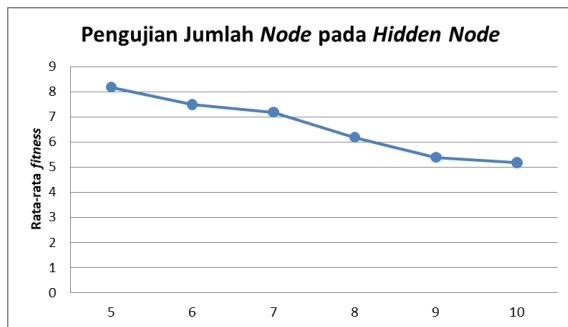
Berdasarkan Gambar 5 hasil penguujian bobot inersia menunjukkan bahwa, semakin besar nilai bobot inersia maka rata-rata *fitness* yang dihasilkan akan semakin besar pula. Peningkatan nilai rata-rata *fitness* dikarenakan apabila bobot inersia semakin besar maka nilai kecepatan partikel yang dihasilkanpun juga besar sehingga kecepatan pada partikel menjadi lebih lambat pada awal pencarian solusi. Oleh sebab itu, apabila bobot inersia yang digunakan semakin kecil maka kecepatan partikel akan semakin cepat sehingga kesempatan eksploitasi partikel semakin kecil dan partikel akan lebih cepat melakukan eksplorasi (Sugianto, et al., 2017).

6.4 Penguujian Jumlah *Hidden Neuron*

Penguujian jumlah *hidden neuron* bertujuan untuk mengetahui jumlah node terbaik pada *hidden neuron* untuk mendapatkan hasil prediksi

yang optimal. Pengujian jumlah hidden neuron dimulai dari angka 5 sampai 10. Parameter yang digunakan pada pengujian ini yaitu:

- a. Nilai Konstanta 1 (c1) : 1
- b. Nilai Konstanta 2 (c2) : 1
- c. Banyak data training: 80%
- d. Jumlah ukuran populasi : 50
- e. Bobot Inersia : 0.5
- f. Jumlah Iterasi : 250



Gambar 6 Pengujian Jumlah Node pada Hidden Neuron

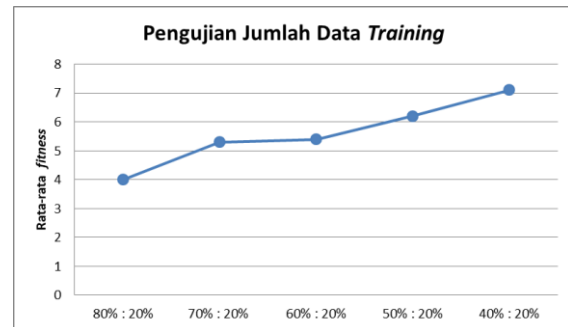
Berdasarkan gambar 6 hasil pengujian jumlah *node* pada *hidden neuron* menunjukkan bahwa semakin banyak jumlah *node* pada *hidden neuron* maka nilai dari rata-rata fitness akan semakin besar. Dari pengujian yang dilakukan, didapatkan nilai rata-rata fitness terkecil terdapat pada jumlah node 10 yakni 5.2. Fungsi dari *node* sendiri dalam metode ELM adalah sebagai penghubung yang terbentuk antara *input layer* dan *output layer*. Sehingga dapat disimpulkan bahwa *node* pada *hidden neuron* dengan jumlah 7 menghasilkan nilai yang optimal.

6.5 Pengujian Jumlah Data

Pengujian jumlah data training dilakukan untuk mengetahui pengaruh banyaknya data training terhadap proses perhitungan algoritma ELM. Pengujian ini terdiri dari 5 jenis perbandingan data, yaitu 80%, 70%, 60%, 50%, dan 40% untuk data *training* dan data *testing* dengan tujuan untuk mengetahui pengaruh dari jumlah data *training* terhadap jumlah data *testing* yang mana data testing sebesar 20% dari keseluruhan data (database) selain yang digunakan untuk data *training*. Parameter yang digunakan pada pengujian ini yaitu:

- a. Nilai Konstanta 1 (c1) : 1
- b. Nilai Konstanta 2 (c2) : 1
- c. Bobot Inersia : 0.5

- d. Jumlah ukuran populasi : 50
- e. Jumlah Hidden Neuron : 10
- f. Jumlah Iterasi : 250



Gambar 7 Pengujian Jumlah Data Training

Berdasarkan gambar 7 rata-rata nilai *fitness* akan bertambah besar apabila jumlah data *training* semakin kecil. Hal tersebut dibuktikan pada perbandingan data 80% : 20% dengan rata-rata nilai *fitness* terkecil yang memiliki nilai 4.0. Hal ini dikarenakan metode ELM merupakan metode pelatihan dimana semakin banyak nilai data training yang digunakan maka semakin banyak data yang dilatih pada proses pelatihan sehingga memiliki banyak pertimbangan untuk menghasilkan optimasi yang baik.

6.6 Pengujian Prediksi Extreme Learning Machine (ELM)

Pada pengujian prediksi ELM bertujuan untuk mencari nilai MAPE dengan menggunakan *input weight* yang sudah optimal. Nilai input weight yang digunakan adalah [0.99, -0.4, -0.04, 0.73, -0.73, -0.42, -0.33, -0.18, 0.11, 0.23, -0.85, -0.94, -0.62, -0.28, 0.18, -0.9, 0.68, -0.06, 0.55, -0.5, 0.15, 0.5, -0.37, -0.35, -0.56, 0.17, -0.31, -0.14, -0.27, -0.19, -0.41, -0.72, -0.19, -0.73, -0.02, 0.27, 0.13, -0.58, 0.89, 0.56, -0.01, 0.37, -0.28, 0.51, 0.43, -0.24, -0.38, 0.84, -0.9, -0.1, 0.68, -0.57, -0.41, -0.56, 0.43, -0.71, 0.08, -0.36, -0.69, -0.87]. Data yang digunakan untuk pengujian ini diambil secara random dari data testing yang menghasilkan nilai MAPE sebesar 0.59%.

6.7 Hasil Pengujian dan Analisis Global

Pada penelitian ini telah dilakukan 6 kali percobaan terkait dengan metode yang digunakan. Pengujian pertama terkait dengan jumlah ukuran populasi menunjukkan apabila jumlah ukuran populasi semakin besar maka nilai rata-rata *fitness* semakin kecil. Dari pengujian tersebut didapat jumlah ukuran

populasi sebanyak 50 dengan rata-rata nilai *fitness* 5.01%. Pengujian kedua terkait dengan jumlah iterasi maksimal, dikarenakan banyaknya iterasi membuat pencarian solusi optimal lebih banyak. Dari pengujian ini didapatkan iterasi maksimal adalah 250 dengan rata-rata nilai *fitness* 3.06%. Pengujian ketiga terkait dengan pengaruh bobot inersia terhadap nilai *fitness*, bahwa semakin besar bobot inersia yang digunakan maka perubahan kecepatan yang dihasilkan juga besar yang artinya kecepatan partikel akan menjadi lambat. Sedangkan bobot inersia sendiri merupakan pengontrol dari perubahan kecepatan partikel dalam mencari solusi yang optimal sehingga menyebabkan partikel mempunyai kesempatan untuk melakukan eksploitasi lebih dalam dan menghasilkan solusi yang paling optimal. Dalam pengujian ini dihasilkan nilai bobot inersia yang optimal adalah 0.5 dengan nilai rata-rata *fitness* 4.08%.

Pengujian keempat terkait dengan banyaknya jumlah hidden neuron yang menunjukkan semakin banyak jumlah hidden neuron maka nilai *fitness* akan semakin kecil. Dari pengujian ini didapatkan jumlah hidden neuron yang paling baik berjumlah 10 dengan nilai rata-rata *fitness* 5.21%. Pengujian kelima terkait dengan pengaruh banyaknya perbandingan data training dan data testing yang digunakan. Dari hasil pengujian menunjukkan bahwa penggunaan data training yang banyak bisa memberikan hasil yang baik. Perbandingan data training dan data testing dari hasil pengujian adalah 36 data training dan 9 data testing dengan nilai *fitness* 4.03 %. Pengujian terakhir atau pengujian keenam terkait dengan pengujian prediksi dimana dalam pengujian ini menguji nilai input weight yang optimal. Dalam pengujian ini data diambil secara random yang menghasilkan nilai MAPE sebesar 0.59%.

7 KESIMPULAN

Berdasarkan perancangan, implementasi dan hasil pengujian *Particle Swarm Optimization* dalam Memprediksi Produksi Gula Kristal Putih dengan *Extreme Learning Machine*, didapatkan kesimpulan sebagai berikut:

1. Parameter-parameter terbaik yang didapat dari hasil pengujian adalah jumlah ukuran populasi 50, perbandingan data *training* 80% dengan jumlah data *training* sebanyak 36 data, jumlah *hidden neuron* 10, bobot

inersia 0.5, dan iterasi maksimal yakni 250. Dari hasil perhitungan parameter tersebut didapatkan nilai *Mean Absolute Percentage Error*(MAPE)-nya 1.24% dengan jumlah partikel dengan nilai [0.99, -0.4, -0.04, 0.73, -0.73, -0.42, -0.33, -0.18, 0.11, 0.23, -0.85, -0.94, -0.62, -0.28, 0.18, -0.9, 0.68, -0.06, 0.55, -0.5, 0.15, 0.5, -0.37, -0.35, -0.56, 0.17, -0.31, -0.14, -0.27, -0.19, -0.41, -0.72, -0.19, -0.73, -0.02, 0.27, 0.13, -0.58, 0.89, 0.56, -0.01, 0.37, -0.28, 0.51, 0.43, -0.24, -0.38, 0.84, -0.9, -0.1, 0.68, -0.57, -0.41, -0.56, 0.43, -0.71, 0.08, -0.36, -0.69, -0.87].

2. Tingkat kesalahan prediksi pada ELM yang dioptimasi dengan PSO dengan nilai evaluasi menggunakan MAPE menghasilkan nilai sebesar 0.59% dengan tingkat prediksi yang baik.
3. Tingkat kesalahan antara ELM biasa dengan ELM-PSO lebih kecil tingkat kesalahan ELM-PSO daripada ELM biasa. Hal ini menunjukkan bahwa metode optimasi PSO mampu meningkatkan nilai *error* untuk permasalahan prediksi produksi.

DAFTAR PUSTAKA

- Agustina, Irwin Dwi. 2010. *Penerapan Metode Extreme Learning Machine untuk Peramalan Permintaan*. Institut Teknologi Sepuluh Nopember.
- Cholissodin, I., 2016. Analisis Big Data-Semester Ganjil 2017-2018.
- Cholissodin, I., 2017. Buku Ajar Swarm Intelligence.
- Cholissodin, Imam. 2016. *Optimasi Kandungan Gizi Susu Kambing Peranakan Etawa (PE) Menggunakan ELM – PSO Di UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari – Malang*. Universitas Brawijaya.
- Handika, I Putu S. 2016. *Perbandingan Metode Extreme Learning Machine dan Particle Swarm Optimization Extreme Learning Machine untuk Peramalan Jumlah Penjualan Barang*. Teknologi Elektro, Vol.15, No.1, Januari-Juni.
- Muhamad, H., Cahyo, A.P., Sugianto, N.A., Surtiningsih, L., 2016. *Optimasi Naïve Bayes Classifier Dengan Menggunakan Particle Swarm Optimization Pada Data Iris*. Jurnal Teknologi Informasi dan Ilmu

Komputer. Universitas Brawijaya.

- Napitulu, Dewi Agustina. 2013. *Faktor-Faktor Yang Mempengaruhi Produksi Gula Dalam Negeri dan Proyeksi Produksi dan Konsumsi Gula Di Indonesia*. Universitas Atma Jaya Yogyakarta.
- Subagyo, Pangestu. 1986. *Forecasting Konsep dan Aplikasi*. BPEE UGM: Yogyakarta.
- Sugiyanto, Catur. 2007. *Permintaan Gula Di Indonesia*. Universitas Gajah Mada.
- Sun, Z.L., Choi, T.M., Au, K.F., dan Yu, Y. 2008. *Sales Forecasting using Extreme Learning Machine with Application in Fashion Retailing*. *Elsevier Decision Support Systems* 46 (2008) 411-419.
- Siwi, Iga P. 2016. *Peramalan Produksi Gula Pasir Menggunakan Extreme Learning Machine (ELM) Pada PG Candi Baru Sidoarjo*. Universitas Brawijaya.
- S, Karpachlevi. 2014. *Classification of ECG Signals Using Hybrid Particle Swarm Optimization in Extreme Learning Machine*. *Journal of Applied Sciences and Engineering Research*, Vol. 3, Issue 5.
- Sugianto, Nur Afifah. 2017. *Klasifikasi Keminatan Menggunakan Algoritme Extreme Learning Machine dan Particle Swarm Optimization Untuk Seleksi Fitur (Studi Kasus: Program Studi Teknik Informatika FILKOM UB)*. Universitas Brawijaya.
- Vincent Gaspersz. 2002. *Production Planning and Inventory Control*. PT. Gramedia Pustaka Utama: Jakarta
- Zhao, Min-Ru, Zhang, Jian-Ming, Han, Feri. 2014. *An Improved Extreme Learning Machine with Adaptive Growth of Hidden Nodes based on Particle Swarm Optimization*. *International Joint Conference on Neural Networks (IJCNN)*. Beijing, China. July 6-11 2014.