

## Klasifikasi Data Cardiotocography dengan Integrasi Metode Neural Network dan Particle Swarm Optimization

Khalid<sup>1</sup>, Bagus Setya Rintyarna<sup>1,2</sup>, Agus Zainal Arifin<sup>1</sup>

<sup>1</sup>Laboratorium Vision and Image processing, Jurusan Teknik Informatika ITS Surabaya

<sup>2</sup>Jurusan Teknik Elektro Universitas Muhammadiyah Jember

khalid10@mhs.if.its.ac.id, bagus10@mhs.if.its.ac.id, agusza@cs.its.ac.id

### Abstrak

*Backpropagation (BP)* adalah sebuah metode yang digunakan dalam training *Neural Network (NN)* untuk menentukan parameter bobot yang sesuai. Proses penentuan parameter bobot dengan menggunakan metode *backpropagation* sangat dipengaruhi oleh pemilihan nilai *learning rate (LR)*-nya. Penggunaan nilai *learning rate* yang kurang optimal berdampak pada waktu komputasi yang lama atau akurasi klasifikasi yang rendah. Penelitian ini mengusulkan algoritma *Particle Swarm Optimization (PSO)* dalam training *Neural Network* untuk optimasi penentuan nilai bobot *Neural Network* dalam klasifikasi data *Cardiotocography*. *Principal Component Analysis (PCA)* diimplementasikan untuk reduksi fitur data *Cardiotocography*. Berdasarkan hasil uji coba, implementasi *Principal Component Analysis* mampu meningkatkan akurasi klasifikasi sebesar rerata 0.04%. Sedangkan optimasi *Particle Swarm Optimization* pada proses training *Neural Network* menghasilkan peningkatan kecepatan komputasi sebesar rerata 6 kali pada berbagai jumlah *Neuron* dan nilai *learning rate* yang berbeda dengan nilai perbedaan akurasi klasifikasi yang tidak signifikan.

### 1. Pendahuluan

*Cardiotocography* adalah sebuah metode yang digunakan untuk melakukan evaluasi kondisi janin selama masa kehamilan [1]. Metode klasifikasi yang diusulkan bertujuan untuk memprediksi atau mengelompokkan data *cardiotocography* ke dalam 3 kelas yaitu : Normal, Suspect, dan Pathologic.

Nidhal et al., mengusulkan sebuah metode optimasi untuk menentukan *fetal heart rate baseline* pada dataset *cardiotocography* [2]. Metode yang diusulkan berupa sebuah algoritma berdasarkan digital *cardiotocography* yang diimplementasikan di matlab untuk meng-estimasi nilai *Fetal Heart Rate (FHR) baseline* yang mengindikasikan kondisi kesehatan janin. Metode diusulkan untuk memberikan solusi penentuan nilai *FHR baseline* yang selama ini dilakukan secara manual oleh

*obstetricians* selama proses analisis data *cardiotocography*.

Klasifikasi data *cardiotocography* dengan menggunakan *NN* diusulkan oleh Magenes et al., [3]. Sebuah *Multilayer Perceptron (MLP) network* dengan 8 neuron, 2 tansigmoid hidden layers dan sebuah *output layer* dengan 3 *neuron* diimplementasikan untuk melakukan klasifikasi. *MLP* ditraining dengan menggunakan algoritma *adaptive backpropagation*. Uji coba menunjukkan *error* sebesar 5.5%.

Jezewski et al., [4] melakukan optimasi algoritma *NN* dengan mengimplementasikan *Radial Basis Function (RBF)* dan *MLP*. Perbandingan hasil klasifikasi dievaluasi dengan menghitung beberapa parameter antara lain : *Sensitivity*, *Specifity*, *Positive Predictive Value (PPV)* dan *Negative Predictive Value (NPV)*. Eksperimen menunjukkan bahwa hasil klasifikasi berbanding lurus dengan jumlah *neuron* [4].

Kesulitan Metode *Backpropagation* dalam proses training *NN* terletak pada proses penentuan parameter bobot-nya. Penentuan nilai parameter bobot pada *NN* dipengaruhi pemilihan jumlah *hidden layer* dan nilai *learning rate* yang sesuai. Nilai *learning rate* yang tidak sesuai berdampak pada hasil klasifikasi yang kurang optimal. Apabila nilai *learning rate* terlalu besar, *network* akan mencapai konvergen dalam waktu yang singkat tetapi *error* klasifikasi menjadi besar. Sebaliknya bila *learning rate* terlalu rendah, hasil klasifikasi memiliki akurasi yang baik tetapi proses training membutuhkan waktu yang lama [5].

Penelitian ini mengusulkan algoritma *PSO* untuk melakukan optimasi dalam proses training *NN*. Sebelum dilakukan proses klasifikasi pada data *cardiotocography*, terlebih dahulu diimplementasikan *PCA* untuk memilih fitur data yang memiliki bobot *Principal Component* yang tinggi. Metode yang diusulkan diharapkan mampu meningkatkan akurasi sekaligus kecepatan komputasi pada klasifikasi data *cardiotocography*.

### 2. Dataset Cardiotocography

*Dataset Cardiotocography* didapatkan dari Doppler Ultrasound Transducer dan Pressure Transducer.

**Tabel 1** Deskripsi Dataset Cardiotocography

Fitur	Kelas			Jumlah Data
	Normal	Suspected	Pathologic	
21	1655	295	176	2126

Dataset merekam perubahan fetal heart rate dan hubungan temporal-nya dengan kontraksi uterine janin dalam kandungan. Analisa pola dataset memungkinkan dilakukannya prediksi dini kondisi janin [1].

Dalam penelitian ini, *dataset Cardiotocography* yang digunakan diunduh dari website Center for Machine Learning and Intelligent System, University of California Irvine Machine Learning Repository (<http://archive.ics.uci.edu/ml/>). Dataset disediakan oleh J. Bernardes, Faculdade de Medicina, Universidade do Porto, Porto, Portugal. Deskripsi dataset bisa dilihat dalam Tabel 1.

**3. Integrasi PCA, NN, dan PSO**

Diagram dari metode yang di gunakan dalam penelitian ini secara keseluruhan bisa digambarkan dalam blok diagram di Gambar 1.

**3.1 Principal Component Analysis**

PCA pertama kali diperkenalkan oleh Pearson (1901) dan dikembangkan oleh Hotelling (1933) dan merupakan sebuah metode yang sangat bermanfaat dalam analisis multivariate [7]. PCA mengidentifikasi pola dalam data dengan memberikan bobot persamaan dan perbedaan antar fitur dalam data tersebut [8]. Tujuan diimplementasikannya PCA adalah untuk mengekstraksi informasi-informasi paling penting dalam sebuah data dan merepresentasikannya dalam variabel-variabel orthogonal yang disebut sebagai principle component.

Oleh karena itu, PCA dapat digunakan untuk mengurangi dimensi datanya bila dirasa perlu [9].

Langkah-langkah implementasi PCA diawali dengan penyediaan *dataset*. Selanjutnya *dataset* ditransformasi dengan mengurangi nilainya dengan nilai *mean* dari fitur. Setelah data ditransformasi, langkah berikutnya adalah menghitung *covariance matrix*. *Covariance matrix* menggambarkan korelasi antara dua buah fitur dalam *dataset*. Untuk fitur *X* dan fitur *Y*, nilai *covariance matrix*-nya sebesar :

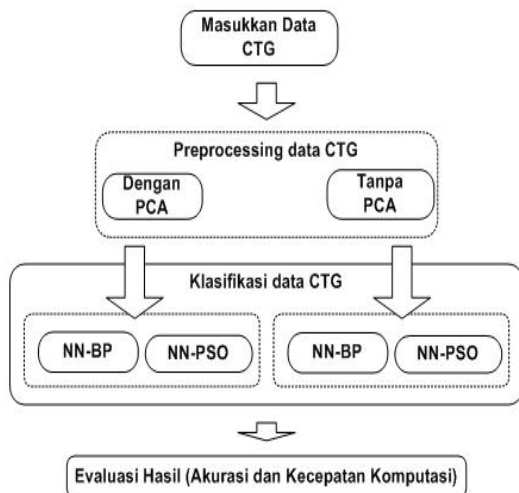
$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \tag{1}$$

dimana,  $X_i$  adalah nilai setiap *dataset* dalam fitur *X* dan  $\bar{X}_i$  adalah mean dari fitur *X*, demikian juga  $Y_i$  dan  $\bar{Y}_i$ , sedangkan *n* adalah jumlah *dataset*. Setelah didapatkan *covariance matrix*, dihitung *eigenvalue* dan *eigenvector* dari *covariance matrix*. Dari nilai *eigenvalue* yang didapatkan dapat ditentukan urutan Principal Component dari fitur-fiturnya. Tahap akhir dari implementasi PCA adalah melakukan *generate dataset* baru berdasarkan *eigenvector* yang didapatkan [8].

**3.2 Particle Swarm Optimization**

PSO pertama kali diusulkan oleh J. Kennedy dan R. C Eberhart merupakan sebuah metode optimasi yang didasarkan pada *swarm intelligence* dan menerapkan *evolution-based procedure* [10]. Dalam kasus optimasi *multivariable*, *swarm* bekerja secara parallel pada masing-masing variabel yang dioptimasi sehingga memungkinkan untuk mendapatkan hasil optimum dalam waktu yang lebih cepat. Dalam konteks inilah PSO diharapkan mampu meningkatkan kecepatan komputasi NN. Dalam penelitian ini, PSO diintegrasikan dengan NN untuk mengoptimasi nilai-nilai parameter NN.

Langkah-langkah implementasi algoritma PSO diawali dengan menentukan ukuran kawanan partikel (*Particle Swarm*) sebesar *N*. Setelah itu degenerate populasi nilai solusi awal, untuk tiap-tiap partikel dengan nilai *X*. Fungsi obyektif tiap-tiap partikel dihitung :  $f(X)$ . Pada langkah ini, iterasi diset nilainya sama dengan 1. Dan nilai 0 diberikan untuk nilai kecepatan masing-masing partikel pada iterasi ini (iterasi ke-1). Pada iterasi berikutnya, dihitung parameter untuk masing-masing partikel *j* yaitu  $P_{best}$  dan  $G_{best}$ .  $P_{best}$  merupakan nilai fungsi obyektif optimum yang ditemui sebuah partikel *j*. Sedangkan  $G_{best}$  yaitu nilai fungsi obyektif optimum di antara semua partikel untuk semua iterasi, Setelah  $P_{best}$  dan  $G_{best}$  ditentukan, dihitung kecepatan partikel *j* pada iterasi ke-*i* dengan menggunakan rumus 2 :



**Gambar 1.** Metode Klasifikasi data CTG

**Tabel 2.** Jumlah iterasi untuk mencapai nilai konvergen

Metode	Hidden Layer	Tanpa PCA		PCA	
		Epoch (i)	Error (%)	Epoch (i)	Error (%)
NNBP	4	250	0.28	250	0.34
NNBP	8	250	0.25	250	0.35
NNBP	12	250	0.24	250	0.29
NNBP	16	250	0.20	250	0.31
NNBP	20	250	0.20	250	0.29
NNPSO	4	36	0.22	36	0.25
NNPSO	8	36	0.28	34	0.25
NNPSO	12	34	0.28	34	0.27
NNPSO	16	34	0.23	35	0.29
NNPSO	20	34	0.35	35	0.27

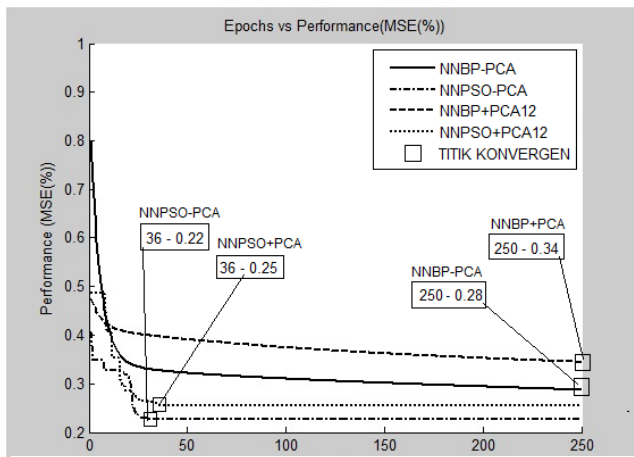
$$V_j(i) = V_j(i - 1) + c_1 r_1 [P_{best,j} - x_j(i - 1)] + c_2 r_2 [G_{best,j} - x_j(i - 1)], j = 1, 2, \dots, N \quad (2)$$

dimana,  $c_1$  dan  $c_2$  masing adalah *learning rate*, sedangkan  $r_1$  dan  $r_2$  yang masing masing adalah standard normal random variabel.

Setelah didapatkan kecepatan masing-masing partikel, langkah berikutnya dilakukan dengan menghitung posisi partikel  $j$  pada iterasi ke- $i$  :

$$X_j(i) = X_j(i - 1) + V_j(i); j = 1, 2, 3, \dots, N \quad (3)$$

Seperti pada iterasi pertama, nilai fungsi obyektif  $f(X)$  dihitung, apabila nilai yang konvergen, langkah-langkah tersebut diulang hingga didapatkan hasil yang konvergen atau hingga batas iterasi yang ditentukan dicapai.



**Gambar 2.** Pengaruh PCA terhadap MSE pada NN dengan Hidden Neuron 4

#### 4. Hasil Uji Coba

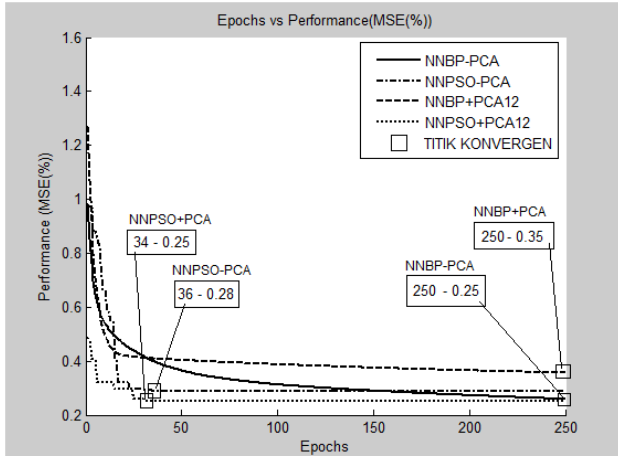
Evaluasi hasil performance sistem dilakukan dengan membandingkan beberapa parameter yang merepresentasikan akurasi klasifikasi dan kecepatan komputasinya. Parameter-parameter tersebut adalah : *Mean Squared Error* (MSE) dan waktu *training* untuk mencapai nilai konvergen.

MSE mengukur rerata kuadrat *error* klasifikasi. Dalam statistik, MSE merupakan salah satu cara untuk menghitung perbedaan antara hasil dengan nilai sebenarnya (*true value*). Dengan demikian MSE mampu memberikan gambaran tentang performance sebuah system, dalam konteks NN untuk klasifikasi, semakin baik performance NN semakin baik pula hasil klasifikasi (akurasi) yang dihasilkan.

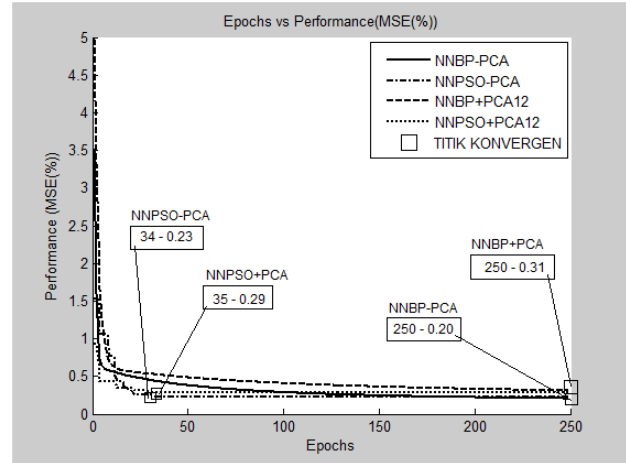
Sedangkan evaluasi implementasi PCA dengan fitur yang telah direduksi menjadi 12 didasarkan pada penghitungan *eigenvalue* dengan menggunakan matlab untuk menentukan *principal component*-nya.

Uji coba pada berbagai jumlah *Hidden Neuron* dan nilai *Learning Rate* yang berbeda menunjukkan adanya trend penurunan MSE sebesar 0.04% pada system yang di *pre-processing* dengan PCA. Hal ini bisa dilihat pada Gambar 2, 3, 4 5, dan 6.

Sedangkan kecepatan komputasi dievaluasi dengan membandingkan *convergence time* masing-masing metode. *Convergence time* adalah waktu yang diperlukan dalam *training* NN sehingga didapatkan nilai parameter NN yang mendekati kebenaran. Pada penelitian ini, *Convergence time* dievaluasi dengan membandingkan jumlah iterasi/*epoch* yang diperlukan untuk mendapatkan nilai konvergen.



Gambar 5 Pengaruh PCA terhadap MSE pada NN dengan Hidden Neuron 8



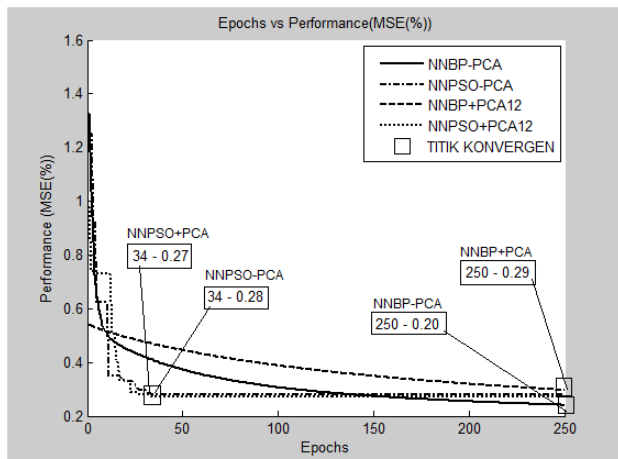
Gambar 3 Pengaruh PCA terhadap MSE pada NN dengan Hidden Neuron 16

5. Pembahasan

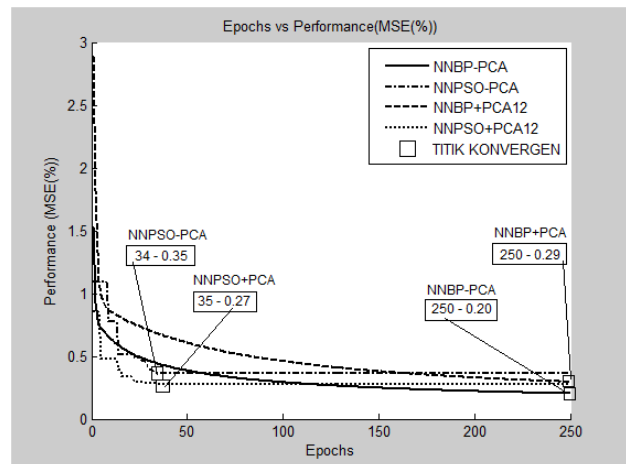
Hasil uji coba NN pada klasifikasi data *cardiotocography* menunjukkan bahwa implementasi *pre-processing* dengan menggunakan PCA menunjukkan adanya trend penurunan MSE sebesar rerata 0.04%. Hal ini mendukung pernyataan Smith [8], bahwa tujuan PCA adalah untuk mentransformasi *dataset* ke arah koordinat fitur yang memiliki korelasi tinggi sehingga memungkinkan fungsi diskriminan dalam proses klasifikasi memisahkan data pada kelas yang berbeda dengan baik untuk mendapatkan hasil klasifikasi yang lebih baik pula.

Selain itu, uji coba menunjukkan bahwa implementasi PSO berdampak pada peningkatan kecepatan komputasi sebesar rerata 6 kali lipat. Hal ini bisa dilihat pada Tabel 2. Pada tabel tersebut kecepatan komputasi dievaluasi dengan membandingkan *convergence time* yang diindikasikan dengan jumlah iterasi yang diperlukan untuk mencapai nilai parameter

yang konvergen. Dari tabel terlihat bahwa penerapan PSO berdampak pada jumlah iterasi yang lebih sedikit untuk mencapai nilai parameter yang konvergen dengan perbedaan nilai *error* yang tidak signifikan. Kondisi ini konsisten pada jumlah *Neuron* dan pada nilai *Learning Rate* yang berbeda. Hal ini menguatkan penjelasan Kennedy bahwa PSO bekerja sebagai sebuah *swarm* (kumpulan) [10] yang bekerja secara paralel dan simultan untuk menemukan nilai optimal sebuah fungsi obyektif sehingga diperlukan waktu komputasi yang lebih cepat [11].



Gambar 4 Pengaruh PCA terhadap MSE pada NN dengan Hidden Neuron 12



Gambar 6 Pengaruh PCA terhadap MSE pada NN dengan Hidden Neuron 20

6. Kesimpulan

Berdasarkan hasil uji coba dapat disimpulkan bahwa integrasi NN dan PSO bisa diimplementasikan untuk melakukan optimasi dalam penentuan parameter bobot pada proses training NN. Pengaruh implementasi PCA

menghasilkan kenaikan akurasi klasifikasi sebesar rerata 0.04%. Selain itu sistem terbukti memiliki keunggulan dalam aspek kecepatan komputasi dalam mencapai titik konvergen dengan rerata 6 kali lebih cepat dibanding BPNN dengan tingkat perbedaan akurasi klasifikasi yang tidak signifikan.

### Referensi

- [1] Z. Alfirefic, D. Devane, G. Gyte, "Continuous Cardiocography (CTG) as a Form of Electronic Fetal Monitoring (EFM) for Fetal Assessment During Labour (Review)", *The Cochrane Library*, Issue 4, 2007.
- [2] S. Nidhal, M. A. Mohd. Ali, H. Najah, "A novel Cardiocography Fetal Heart Rate Baseline", *Scientific Research and Essays*, Vol 5(24), 2010.
- [3] G. Magenes, M.G. Signorini, D. Arduini, "Classification of Cardiocographyc Records by Neural Networks", *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 2000.
- [4] M. Jezewski, J. Wrobel, J. Leski, "Some Practical Remarks on Neural Networks Approach to Fetal Cardiocogram Classification", *Proceeding of the 29<sup>th</sup> Annual International Conference of the IEEE EMBS*, 2007.
- [5] P. Francis, C. Gilles, B. Yoshua, "Neural Network : Synthesis of Backpropagation Optimization Problems", *IEEE Transactions on Neural Networks*, No. 2, 2003.
- [6] J.J. Hopfield, D.W. Tank, "Neural Computation of Decisions in Optimization Problem", *Biological Cibernetics*, 52, 141-152, 1985.
- [7] I.T. Jolliffe, "Principle Component Analysis", *Springer Series in Statistics*, 2002.
- [8] L.I. Smith, "A tutorial on Principal Component Analysis", 2002.
- [9] H. Abdi, L.J. William, "Principle Component Analysis", *Wiley Interdisciplinary Reviews : Computational Statistics 2*, 2010.
- [10] J. Kennedy, R.C. Eberhart, "Particle Swarm Optimization", *Computational Intelligence PC Tool*, 1996.
- [11] R.C. Eberhart, Y. Shi, "Particle Swarm Optimization : Development, Application and Resources", *Proceeding of 2001 Congress of Evolutionary Computation*, 2001.