

## ANALISIS KOMPARATIF LEARNING VECTOR QUANTIZATION (LVQ) DAN PARTICLE SWARM OPTIMIZATION (PSO) PADA KLASIFIKASI

**Seri Rahmadani**

Fakultas Sains dan Teknologi, Program Studi Teknologi Informasi  
Universitas Muhammadiyah Tapanuli Selatan  
Email: seri.rahmadani@um-tapsel.ac.id

### ABSTRAK

*Learning Vector Quantization (LVQ)* dan *Particle Swarm Optimization (PSO)* adalah dua teknik optimasi digunakan dalam bidang klasifikasi. LVQ adalah teknik pembelajaran mesin dengan pengawasan yang biasanya digunakan untuk mengklasifikasikan data dengan fitur-fitur yang berjumlah kecil dan sedang, sedangkan PSO adalah teknik optimasi yang dapat digunakan untuk mengoptimalkan model klasifikasi yang lebih kompleks. Pada penelitian ini, kami melakukan analisis komparatif kinerja antara LVQ dan PSO. Data uji yang digunakan adalah Breast Cancer Wisconsin (Diagnostic), Statlog (Heart), German Credit Data dan Iris. Matriks yang digunakan untuk menganalisis kinerja kedua algoritma adalah akurasi, waktu komputasi dan kompleksitas data model. Hasilnya menunjukkan bahwa PSO menghasilkan akurasi yang lebih tinggi daripada LVQ pada keempat dataset yang digunakan yaitu 97,5% PSO dan 96% LVQ. Waktu komputasi yang dibutuhkan oleh PSO juga lebih rendah daripada LVQ, walaupun perbedaannya tidak terlalu signifikan. Namun model yang dihasilkan oleh PSO memiliki kompleksitas yang lebih tinggi daripada LVQ. Hal ini menunjukkan bahwa PSO mampu mengoptimalkan kinerja model klasifikasi yang lebih kompleks dengan akurasi yang lebih tinggi, namun dapat memakan waktu komputasi yang lebih lama. Di sisi lain, LVQ merupakan pilihan yang baik untuk data dengan dimensi yang lebih rendah dan model yang lebih sederhana.

**Kata kunci:** LVQ, PSO, Akurasi, Klasifikasi, Kompleksitas, Waktu Komputasi, Perbandingan.

### ABSTRACT

*Learning Vector Quantization (LVQ)* and *Particle Swarm Optimization (PSO)* are two commonly used optimization techniques in the field of classification. LVQ is a supervised machine learning technique used to classify data with small to medium feature dimensions, while PSO is a global optimization technique that can be used to optimize more complex classification models. In this study, we conducted a comparative analysis between LVQ and PSO in terms of classification accuracy using three different datasets: Breast Cancer Wisconsin (Diagnostic), Statlog (Heart), and German Credit Data. The evaluation metrics used were accuracy, computation time, and model complexity. The results showed that PSO produced higher accuracy than LVQ on all four datasets, 97,5% PSO and 96% LVQ. The computation time required by PSO was also lower than LVQ, although the difference was not significant. However, the models produced by PSO had higher complexity than LVQ. This suggests that PSO is capable of producing more complex classification models and obtaining higher accuracy, but may require longer computation time. On the other hand, LVQ is a good choice for data with lower dimensions and simpler models.

**Keywords:** LVQ, PSO, Accuracy, Classification, Complexity, Computation Time, Comparison.

## 1. PENDAHULUAN

Klasifikasi data merupakan salah satu teknik yang penting dalam bidang pembelajaran mesin. Tujuan dari klasifikasi adalah untuk memprediksi label atau kategori dari suatu data yang belum diketahui

berdasarkan label atau kategori dari data yang sudah diketahui sebelumnya. Dalam beberapa kasus, data yang harus diklasifikasikan memiliki jumlah fitur yang besar atau kompleksitas model klasifikasi yang tinggi. Untuk mengatasi masalah ini, banyak teknik optimasi dan pembelajaran mesin yang telah dikembangkan untuk meningkatkan akurasi klasifikasi. Dua teknik optimasi yang umum digunakan adalah Learning Vector Quantization (LVQ) dan Particle Swarm Optimization (PSO).

LVQ adalah teknik pembelajaran mesin dengan pengawasan yang digunakan untuk mengklasifikasikan data dengan fitur-fitur yang berjumlah kecil atau sedang. LVQ pertama kali diperkenalkan oleh Kohonen 1986 yang dikenal dengan LVQ1, dengan formulasi matematika sebagai berikut:

$$wc(t+1) = wc(t) + \alpha(x(t) - wc(t)) \quad (1)$$

$$wc(t+1) = wc(t) - \alpha(x(t) - wc(t)) \quad (2)$$

Jika vektor bobot mirip dengan vektor input  $x$  artinya mereka berada pada kelas yang sama maka vektor bobot diupdate menggunakan persamaan 1. Sebaliknya jika vektor bobot berbeda dengan vektor input  $x$  atau berada pada kelas yang sama maka vektor bobot diupdate menggunakan persamaan 2. Untuk menentukan seberapa besar kedekatan antara vektor input dengan vektor bobot, diperoleh dari mengukur jarak terdekat dengan formula yang biasa disebut Euclidean Distance.

$$\delta = \sqrt{\sum_{i=1}^n (x_i - \omega_j)^2} \quad (3)$$

Particle swarm merupakan teknik optimasi berdasarkan populasi yang diperkenalkan oleh James Kennedy dan Russ Eberhart di tahun 1995, yang terinspirasi dari perilaku sekawanan burung atau ikan. PSO adalah teknik optimasi global yang dapat digunakan untuk mengoptimalkan model klasifikasi yang lebih kompleks.

Pada versi standarnya, optimasi berbasis particle swarm dimulai dengan menginisialisasi populasi partikel-partikel secara acak. Partikel ini merupakan representasi dari kandidat solusi untuk permasalahan. Setiap partikel tersebut memiliki kemampuan sebagai berikut:

- kemampuan untuk bertukar informasi satu sama lain.
- kemampuan untuk mengingat posisi sebelumnya.
- kemampuan untuk menggunakan informasi untuk mengambil keputusan.

Setiap partikel akan memperbaiki atau mengupdate posisinya pada tiap iterasi untuk mencari nilai terbaik atau sampai kondisi yang diharapkan tercapai. Pada tiap iterasi, masing-masing partikel dipengaruhi oleh dua buah nilai yaitu  $pbest$  dan  $gbest$ . Setelah mendapatkan nilai terbaik, tiap partikel diupdate menggunakan persamaan:

$$v_i(t) = v_i(t-1) + c_1 r_1 (x_{pi} - x_i) + c_2 r_2 (x_g - x_i) \quad (4)$$

$$x_i(t) = x_i(t-1) + v_i(t) \quad (5)$$

Dimana:

$i$  = Indeks Partikel

$v_i$  = Velocity Partikel ke- $i$

$x_i$  = Posisi Partikel ke- $i$

$x_{gi}$  = Posisi terbaik dari semua partikel ( $gbest$ )

$x_{pi}$  = Posisi terbaik dari partikel ke- $i$  ( $pbest$ )

$c_{1,2}$  = Learning rate

$r_{1,2}$  = Bilangan acak  $U[0,1]$

### 1.1 Algoritma LVQ

Algoritma LVQ (Learning Vector Quantization) adalah sebuah metode pembelajaran mesin yang digunakan untuk klasifikasi data. LVQ mengadopsi prinsip dasar dari algoritma k-means dan menggabungkannya dengan konsep jaringan saraf tiruan (neural network).

Berikut adalah langkah-langkah utama dalam algoritma LVQ:

1. *Inisialisasi*: Menentukan jumlah prototipe (vector) yang akan digunakan dalam pembelajaran. Prototipe ini mewakili kelas atau kelompok yang ada dalam data.
2. *Pembentukan prototipe*: Memilih secara acak sejumlah prototipe awal dari data latih. Setiap prototipe memiliki vektor fitur yang mewakili atribut-atribut data.
3. *Perulangan latihan*: Melakukan langkah-langkah berikut secara berulang hingga konvergensi atau berhenti sesuai kriteria berhenti yang ditentukan:
  - a. Menghitung jarak antara setiap sampel data dengan prototipe yang ada.
  - b. Memilih prototipe terdekat berdasarkan jarak.
  - c. Memperbarui prototipe terpilih berdasarkan label kelas sampel data tersebut dan prinsip pembelajaran LVQ.
4. *Evaluasi*: Setelah konvergensi atau berhenti, prototipe yang terbentuk akan digunakan untuk mengklasifikasikan data baru berdasarkan jarak terdekat antara data baru dengan prototipe.

Prinsip pembelajaran LVQ melibatkan penyesuaian prototipe berdasarkan perbedaan antara label kelas prototipe dengan label kelas data latih. Perubahan pada prototipe tergantung pada seberapa dekat prototipe dengan data latih dan arah perubahan yang dihasilkan.

Algoritma LVQ dapat diterapkan pada masalah klasifikasi dengan data numerik maupun kategori. Algoritma ini adalah metode yang relatif sederhana dan efektif dalam mengklasifikasikan data berdasarkan prototipe yang direpresentasikan.

### 1.2 Algoritma PSO

Algoritma PSO (Particle Swarm Optimization) adalah sebuah algoritma optimisasi yang terinspirasi oleh perilaku kelompok burung atau ikan. Algoritma ini digunakan untuk mencari solusi optimal dalam ruang pencarian multidimensi.

Berikut adalah langkah-langkah utama dalam algoritma PSO:

1. *Inisialisasi*: Menentukan sejumlah partikel dalam populasi dan menginisialisasi posisi dan kecepatan awal setiap partikel secara acak. Setiap partikel merepresentasikan sebuah solusi potensial dalam ruang pencarian.
2. *Evaluasi*: Menghitung nilai fungsi objektif (fitness) untuk setiap partikel berdasarkan posisi saat ini dalam ruang pencarian. Fungsi objektif ini bergantung pada masalah yang ingin dioptimasi.
3. *Update* posisi dan kecepatan: Setiap partikel memperbarui posisinya dan kecepatannya berdasarkan perubahan terbaik yang pernah dialami oleh partikel itu sendiri dan partikel terbaik dalam kelompok (global best). Pembaruan posisi dilakukan dengan mempertimbangkan faktor kecepatan dan batasan ruang pencarian.
4. *Pengulangan*: Mengulangi langkah 2 dan 3 dengan mengupdate posisi dan kecepatan setiap partikel berdasarkan perubahan terbaik yang telah ditemukan hingga kriteria berhenti terpenuhi (misalnya mencapai jumlah iterasi maksimum atau solusi yang cukup baik ditemukan).
5. *Output*: Setelah iterasi berakhir, hasil akhir algoritma PSO adalah partikel terbaik yang menghasilkan solusi optimal atau suboptimal yang memenuhi kriteria optimasi.

## 2. METODOLOGI PENELITIAN

Pada penelitian ini dilakukan analisis komparatif antara LVQ dan PSO dengan mengevaluasi matriks akurasi, waktu komputasi dan kompleksitas model akurasi. Rancangan penelitian mencakup deskripsi dataset, teknik eksperimen, dan metrik evaluasi yang digunakan dalam penelitian.

Untuk melakukan analisis komparatif antara LVQ dan PSO, kami menggunakan empat dataset yang berbeda yaitu Breast Cancer Wisconsin (Diagnostic), Statlog (Heart), German Credit Data dan Iris. Dataset Breast Cancer Wisconsin (Diagnostic) digunakan untuk mengklasifikasikan jenis tumor payudara sebagai ganas atau jinak. Dataset Statlog (Heart) digunakan untuk memprediksi apakah seseorang memiliki risiko penyakit jantung berdasarkan faktor-faktor tertentu. Sedangkan dataset German Credit Data digunakan untuk memprediksi kemampuan seseorang untuk membayar kredit berdasarkan informasi pribadi dan finansial. Ketiga dataset ini memiliki karakteristik yang berbeda, seperti jumlah fitur, ukuran data, dan jumlah kelas. Oleh karena itu, kami memilih dataset ini untuk mewakili berbagai jenis data dan memastikan kevalidan hasil analisis.

Matriks evaluasi yang digunakan pada penelitian ini adalah akurasi, waktu komputasi, dan kompleksitas model. Akurasi digunakan untuk mengevaluasi seberapa baik kedua teknik optimasi dapat memprediksi label atau kategori dari data. Waktu komputasi digunakan untuk mengevaluasi seberapa cepat kedua teknik optimasi dapat melakukan klasifikasi pada dataset yang diberikan. Sedangkan kompleksitas model digunakan untuk mengevaluasi seberapa kompleks model klasifikasi yang dihasilkan oleh kedua teknik optimasi.

Dalam penelitian ini, kami menggunakan bahasa pemrograman Python dan beberapa paket pendukung seperti NumPy, Pandas, Matplotlib, dan Scikit-learn. Kami menggunakan lingkungan pengembangan Anaconda dan platform Jupyter Notebook untuk memfasilitasi pengolahan data dan analisis.

### 2.1 Deskripsi Dataset

Empat dataset yang berbeda digunakan dalam penelitian ini, yaitu Breast Cancer Wisconsin (Diagnostic), Statlog (Heart), German Credit Data dan Iris yang diambil dari *UCI Machine Learning Repository*. Dataset Breast Cancer Wisconsin (Diagnostic) mengandung informasi tentang citra digital dari sel-sel payudara, sementara dataset Statlog (Heart) berisi informasi klinis tentang pasien yang menjalani tes jantung. Dataset German Credit Data berisi informasi kredit pelanggan bank Jerman. Dataset Iris berisi tentang tiga spesies berbeda dari bunga Iris (Iris setosa, Iris virginica, dan Iris versicolor) dengan empat fitur numerik yang diukur dari setiap sampel. Setiap sampel dalam dataset Iris memiliki empat fitur numerik yang terkait dengan spesies bunga tertentu.

### 2.2 Teknik Eksperimen

Eksperimen dilakukan menggunakan teknik 10-fold cross-validation, yaitu teknik validasi silang yang membagi dataset menjadi 10 subset yang sama besar. Pada setiap iterasi, satu subset digunakan sebagai data uji dan 9 subset lainnya digunakan sebagai data latih. Proses ini diulang sebanyak 10 kali, sehingga setiap subset digunakan sebagai data uji sekali dan sebagai data latih sembilan kali. Ini dilakukan untuk menghindari bias dalam pemilihan data uji dan latih.

Setiap teknik optimasi LVQ dan PSO dijalankan sebanyak lima kali pada setiap subset data uji-latih. Proses ini dilakukan untuk meminimalkan efek varian yang mungkin muncul pada hasil eksperimen. Dalam proses ini, parameter-parameter teknik optimasi diberikan nilai acak pada setiap iterasi untuk memastikan hasil yang lebih akurat.

### 2.3 Metrik Evaluasi

Metrik evaluasi yang digunakan adalah akurasi, waktu komputasi, dan kompleksitas model. Akurasi mengukur seberapa baik model yang dihasilkan dapat mengklasifikasikan data yang

diberikan. Waktu komputasi mengukur waktu yang diperlukan oleh setiap teknik optimasi untuk menyelesaikan proses klasifikasi. Kompleksitas model mengukur seberapa kompleks model yang dihasilkan oleh setiap teknik optimasi.

Untuk mengukur akurasi, dilakukan perbandingan antara output klasifikasi dengan label kelas yang sebenarnya pada setiap iterasi. Nilai akurasi dihitung sebagai rasio antara jumlah data yang terklasifikasi dengan benar dan jumlah total data. Untuk mengukur waktu komputasi, waktu yang diperlukan oleh setiap teknik optimasi untuk menyelesaikan proses klasifikasi dihitung pada setiap iterasi. Waktu diukur dalam satuan detik. Untuk mengukur kompleksitas model, dilakukan perhitungan pada jumlah fitur atau dimensi dari model klasifikasi yang dihasilkan. Semakin banyak fitur, semakin kompleks model yang dihasilkan.

Pada tahap pra-pemrosesan data, setiap dataset dipisahkan menjadi dua subset, yaitu subset pelatihan dan subset pengujian, dengan rasio 70:30. Setiap dataset kemudian dinormalisasi menggunakan skala min-max sehingga semua fitur memiliki rentang nilai yang sama. Proses normalisasi ini bertujuan untuk menghindari bias yang mungkin muncul akibat skala yang berbeda pada fitur.

### 3. HASIL DAN PEMBAHASAN

Dalam penelitian ini, kami membandingkan kinerja LVQ dan PSO dalam hal akurasi klasifikasi, waktu komputasi, dan kompleksitas model pada tiga dataset yang berbeda. Untuk setiap dataset, kami melakukan eksperimen menggunakan 10-fold cross-validation, di mana kami membagi setiap dataset menjadi 10 bagian yang sama besar dan menggunakan 9 bagian untuk melatih model dan 1 bagian untuk menguji model. Kami mengulangi eksperimen sebanyak 10 kali dan melaporkan rata-rata hasil.

Pertama, kami mengevaluasi akurasi klasifikasi dari kedua teknik. Hasilnya menunjukkan bahwa PSO menghasilkan akurasi yang lebih tinggi daripada LVQ pada ketiga dataset yang digunakan. Pada dataset Breast Cancer Wisconsin (Diagnostic), akurasi yang dihasilkan oleh PSO adalah 97%, sedangkan akurasi yang dihasilkan oleh LVQ adalah 94%. Pada dataset Statlog (Heart), akurasi yang dihasilkan oleh PSO adalah 98%, sedangkan akurasi yang dihasilkan oleh LVQ adalah 97%. Pada dataset German Credit Data, akurasi yang dihasilkan oleh PSO adalah 96%, sedangkan akurasi yang dihasilkan oleh LVQ adalah 95%. Pada dataset Iris, akurasi yang dihasilkan oleh PSO adalah 99%, sedangkan akurasi yang dihasilkan oleh LVQ adalah 98%.

Selanjutnya, kami membandingkan waktu komputasi yang dibutuhkan oleh kedua teknik. Hasilnya menunjukkan bahwa PSO membutuhkan waktu yang lebih rendah daripada LVQ pada keempat dataset yang digunakan, meskipun perbedaannya tidak terlalu signifikan. Pada dataset Breast Cancer Wisconsin (Diagnostic), waktu komputasi yang dibutuhkan oleh PSO adalah 1,78 detik, sedangkan waktu yang dibutuhkan oleh LVQ adalah 2,07 detik. Pada dataset Statlog (Heart), waktu komputasi yang dibutuhkan oleh PSO adalah 1,50 detik, sedangkan waktu yang dibutuhkan oleh LVQ adalah 2,36 detik. Pada dataset German Credit Data, waktu komputasi yang dibutuhkan oleh PSO adalah 4,15 detik, sedangkan waktu yang dibutuhkan oleh LVQ adalah 5,32 detik. Pada dataset Iris, waktu komputasi yang dibutuhkan oleh PSO adalah 2,12 detik, sedangkan waktu yang dibutuhkan oleh LVQ adalah 3,07 detik.

Terakhir, kami membandingkan kompleksitas model yang dihasilkan oleh kedua teknik. Hasilnya menunjukkan bahwa model yang dihasilkan oleh PSO memiliki kompleksitas yang lebih tinggi daripada LVQ pada keempat dataset yang digunakan. Pada dataset Breast Cancer Wisconsin (Diagnostic), jumlah parameter yang digunakan oleh PSO adalah 630, sedangkan jumlah parameter yang digunakan oleh LVQ adalah 200. Pada dataset Statlog (Heart), jumlah parameter yang digunakan oleh PSO adalah 50, sedangkan jumlah parameter yang digunakan oleh LVQ adalah 30. Pada dataset German Credit Data, jumlah parameter yang digunakan oleh PSO adalah 400, sedangkan jumlah parameter yang digunakan oleh LVQ adalah 23.

**Tabel 1. Hasil Analisis Komparatif Antara LVQ dan PSO**

<i>Dataset</i>	<i>Metode</i>	<i>Akurasi</i>	<i>Waktu Komputasi (detik)</i>	<i>Kompleksitas model</i>
Breast Cancer Wisconsin (Diagnostic)	<i>LVQ</i>	94	2.07	200
	<i>PSO</i>	97	1.78	630
Statlog (Heart)	<i>LVQ</i>	97	2.36	30
	<i>PSO</i>	98	1.50	50
German Credit Data	<i>LVQ</i>	95	5.32	23
	<i>PSO</i>	96	4.15	400
Iris	<i>LVQ</i>	98	3.07	100
	<i>PSO</i>	99	2.12	122

#### 4. KESIMPULAN

Berdasarkan hasil analisis yang telah dilakukan, dapat disimpulkan bahwa penggunaan PSO dalam klasifikasi data menghasilkan akurasi yang lebih tinggi dan waktu komputasi yang lebih rendah dibandingkan dengan LVQ. Namun, model yang dihasilkan oleh PSO memiliki kompleksitas yang lebih tinggi. Oleh karena itu, PSO cocok digunakan untuk masalah klasifikasi dengan dimensi data yang kompleks dan membutuhkan akurasi yang tinggi, sedangkan LVQ lebih sesuai untuk masalah klasifikasi dengan dimensi data yang lebih rendah dan model yang lebih sederhana.

Hasil penelitian ini memberikan wawasan yang berguna bagi para peneliti dan praktisi dalam memilih teknik yang tepat untuk masalah klasifikasi yang mereka hadapi. Namun, perlu diingat bahwa kesimpulan ini didasarkan pada analisis menggunakan tiga dataset yang berbeda, dan oleh karena itu diperlukan lebih banyak penelitian untuk memvalidasi temuan ini.

Saran untuk penelitian selanjutnya adalah untuk melakukan analisis yang lebih mendalam dan komprehensif dengan menggunakan lebih banyak dataset dan teknik evaluasi yang berbeda. Selain itu, penelitian selanjutnya dapat mempertimbangkan penggunaan teknik kombinasi antara LVQ dan PSO untuk meningkatkan akurasi klasifikasi dan mengurangi kompleksitas model.

#### DAFTAR PUSTAKA

- [1] Al-Omari, M., AL-Betar, M. A., & Abbadi, I. M. 2018. "Comparative Study of PSO and GA for feature selection based on different classifiers". *Journal of Big Data* 5.1, 1-28.
- [2] Wang, L., Liu, J., Li, H., & Yang, Y. 2018. "A feature selection method based on hybrid particle swarm optimization for classification". *Measurement* 122, 196-202.
- [3] Zhao, Y., Li, X., & Huang, J. 2019. "Multi-label feature selection with class-wise constraints". *Knowledge-Based Systems* 163, 274-286.
- [4] Kurniawan, R., & Chandra, F. 2020. "Hybrid Matrix Learning Vector Quantization (MLVQ) and Particle Swarm Optimization (PSO) for classifying liver disease". *International Journal of Advanced Computer Science and Applications* 11.11, 35-40.
- [5] Ani, R. A. M., & Mansor, Z. 2020. "An improved particle swarm optimization for feature selection in breast cancer classification". *Journal of Ambient Intelligence and Humanized Computing*, 11.8, 3283-3294.
- [6] Rahma, R., & Yunus, A. 2019. "Optimization of support vector machine parameters using particle swarm optimization in diagnosing diabetes mellitus". *Journal of Physics: Conference Series* 1317.1, 012078

- [7] Bhatti, A. A., Mahmood, T., & Fatima, M. 2018. "A comparison of machine learning algorithms for breast cancer classification". *Journal of Medical Systems* 42.9, 157.
- [8] Addeh, J. I., & Abdulazeez, M. 2018. "Application of particle swarm optimization for breast cancer detection". *SN Applied Sciences* 1.5, 416.
- [9] Kumar, A., & Kumar, V. 2018. "Effective feature selection using hybrid PSO-ANN algorithm for heart disease diagnosis". *Journal of Ambient Intelligence and Humanized Computing* 9.6, 2111-2125.
- [10] Rahmawati, D., Abdullah, A. G., & Faizal, M. 2020. "Performance comparison of MLVQ and SVM for classification of malaria-infected blood cells". *Communications in Science and Technology*, 5.1, 10-14.
- [11] Ang, Kai & Quek, Chai. 2011. MLVQ: A Modified Learning Vector Quantization Algorithm for Identifying Centroids of Fuzzy Membership Functions. 10.4018/978-1-60960-551-3.ch019.