TUGAS MATA KULIAH JARINGAN SARAF TIRUAN (JST)
LK 04 - PENERAPAN METODE SOM DAN LVQ

Disusun oleh:

Fikri Adyatma – 235150201111015

Nadhif Rif'at Rasendriya – 235150201111074

Afriza Herdian Adicandra – 235150201111013

PROGRAM STUDI TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2025

DAFTAR ISI

BAB 1 - PENDAHULUAN

Jaringan Syaraf Tiruan (JST) adalah salah satu teknik dalam kecerdasan buatan yang menduplikasi cara otak manusia dalam mengenali pola dan belajar dari informasi. Dalam konteks data mining dan pembelajaran mesin, JNB sering digunakan untuk tugas klasifikasi dan pengelompokan karena kemampuannya dalam menganalisis representasi data yang rumit. Dua pendekatan yang cukup terkenal dalam JNB adalah Peta Organisasi Diri (POD) dan Kuantisasi Vektor Pembelajaran (KVP). POD diterapkan untuk mengelompokkan data tanpa label (pembelajaran tanpa pengawasan), sedangkan KVP digunakan untuk membuat klasifikasi data yang sudah berlabel (pembelajaran dengan pengawasan). Penggunaan kombinasi dari keduanya sering kali bertujuan untuk memperlihatkan struktur data serta melakukan pengelompokan yang berbasis *prototype*.

Tugas ini bertujuan untuk menerapkan teknik POD dan KVP pada dataset Iris untuk memahami perbedaan antara pengelompokan dan klasifikasi. Dengan menerapkan metode POD, mahasiswa diharapkan dapat mengamati bagaimana data dapat dikelompokkan berdasarkan kesamaan fitur tanpa memerlukan label. Di sisi lain, melalui KVP, akan dilakukan proses belajar dengan pengawasan untuk mengkategorikan data ke dalam kelas tertentu. Sasaran khusus dari laporan ini adalah membandingkan kemampuan interpretasi hasil visualisasi dari POD dan kinerja akurasi KVP dalam mengidentifikasi pola pada *dataset* Iris.

Studi ini berfokus pada penerapan algoritma POD dan KVP menggunakan dataset Iris yang meliputi tiga jenis bunga: Iris setosa, Iris versicolor, dan Iris virginica. Metodologi yang digunakan mencakup langkah-langkah pra-pemrosesan (standarisasi data), pelatihan model POD dan KVP, serta penilaian hasil dengan menggunakan metrik akurasi, matriks kebingungan, dan visualisasi U-Matrix. Eksperimen dilakukan di Google Colab dengan memanfaatkan pustaka Python seperti MiniSom dan Scikit-learn. Keterbatasan penelitian ini adalah fokus pada penggunaan satu dataset kecil (Iris) dengan parameter pelatihan yang sederhana untuk memudahkan analisis baik secara visual maupun konseptual.

BAB 2 - TINJAUAN PUSTAKA

2.1 Self-Organizing Map (SOM)

Self-Organizing Map (SOM) merupakan teknik jaringan saraf buatan yang diperkenalkan oleh Teuvo Kohonen untuk melakukan pengelompokan dan pengurangan dimensi tanpa pengawasan. SOM mentransformasikan data yang memiliki banyak dimensi ke dalam ruang dua dimensi agar pola kesamaan antar data dapat ditampilkan. Setiap neuron pada SOM memiliki bobot yang akan diperbarui selama proses pelatihan untuk mendekati data input tertentu. Neuron yang paling sesuai dengan data tersebut dikenal sebagai Best Matching Unit (BMU). Hasil akhir dipresentasikan dalam bentuk U-Matrix, yang menunjukkan jarak antara neuron dan menggambarkan batas-batas antar kluster. Keunggulan SOM terletak pada kemampuannya untuk memberikan pemahaman visual tentang struktur data, tetapi kelemahannya adalah tidak dapat langsung diterapkan untuk klasifikasi karena sifatnya yang tidak terawasi.
(Referensi: Kohonen, T. (1990). The self-organizing map. Proceedings of the IEEE.)

2.2 Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) adalah sebuah algoritma yang tergolong dalam pembelajaran terawasi dan dikembangkan oleh Kohonen. LVQ mengimplementasikan ide dari vektor prototipe untuk mewakili tiap kategori dalam kumpulan data. Pada fase pelatihan, bobot dari prototipe dimodifikasi dengan cara mendekati prototipe kepada data yang memiliki label serupa dan menjauhkannya dari data yang memiliki label berbeda. Melalui pendekatan ini, LVQ mampu mengelompokkan kategori berdasarkan jarak antar vektor. Salah satu keuntungan dari LVQ adalah proses klasifikasinya yang sederhana dan mudah dipahami, namun kinerjanya sangat dipengaruhi oleh jumlah dan cara inisialisasi prototipe. LVQ sering dipadukan dengan SOM untuk meningkatkan ketepatan klasifikasi setelah proses pengelompokan.
(Referensi: Kohonen, T. (1995). Learning Vector Quantization for Pattern Recognition.)

BAB 3 - METODOLOGI

3.1 Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimen kuantitatif dengan tujuan untuk menerapkan dan menganalisis kinerja metode Self-Organizing Map (SOM) dan Learning Vector Quantization (LVQ) dalam proses pengelompokan dan klasifikasi data. Tahapan penelitian dilakukan secara berurutan mulai dari pengumpulan data, preprocessing, pelatihan model SOM, pelatihan model LVQ, hingga evaluasi hasil menggunakan metrik akurasi dan *confusion matrix*. Penelitian ini bersifat komputasional dan dilaksanakan menggunakan Google Collab sebagai lingkungan pemrograman berbasis Python.

3.2 Data Penelitian

Data yang digunakan dalam penelitian ini adalah dataset Iris yang tersedia pada pustaka sklearn.datasets. Dataset ini berisi 150 sampel data dari tiga spesies bunga iris, yaitu *Iris setosa*, *Iris versicolor*, dan *Iris virginica*.
Setiap sampel memiliki empat fitur numerik, yaitu:

1. Sepal length (cm)
2. Sepal width (cm)
3. Petal length (cm)
4. Petal width (cm)

Label target pada dataset ini berupa kategori spesies bunga:

- Label 0 → *Iris setosa*
- Label 1 → *Iris versicolor*
- Label 2 → *Iris virginica*

Dataset ini dipilih karena memiliki struktur fitur yang sederhana namun sering digunakan sebagai standar pembelajaran dan pengujian algoritma klasifikasi berbasis jaringan saraf tiruan.

3.3 Tahapan Penelitian

Penelitian ini terdiri atas beberapa tahapan utama sebagaimana ditunjukkan pada Gambar alur penelitian:

1. Import dan Instalasi Library

   Tahap ini melibatkan instalasi library tambahan seperti *MiniSom* untuk implementasi Self-Organizing Map, serta *Scikit-learn*, *Matplotlib*, *NumPy*, dan *Pandas* untuk pemrosesan data, visualisasi, dan evaluasi model.
2. Load Dataset dan Preprocessing

   Dataset Iris diambil menggunakan fungsi load_iris() dari sklearn.datasets.

   Seluruh fitur kemudian dilakukan standarisasi menggunakan StandardScaler agar memiliki skala rata-rata 0 dan deviasi standar 1. Langkah ini penting untuk memastikan setiap fitur memiliki kontribusi yang seimbang dalam proses pelatihan jaringan.
3. Pelatihan Self-Organizing Map (SOM)

   SOM digunakan sebagai metode *unsupervised learning* untuk menemukan representasi peta dua dimensi dari data berdimensi tinggi.

   Parameter pelatihan yang digunakan adalah:
   - Iterasi: 1000
   - Learning rate: 0.5
   - Sigma (radius tetangga): 1.0
   - Ukuran grid: 12 × 12 neuron (ditentukan secara otomatis berdasarkan akar jumlah data)
4. Proses pelatihan menghasilkan peta U-Matrix yang menunjukkan jarak antar neuron dan pola klasterisasi alami data. Hasil visualisasi ini memberikan gambaran awal struktur data yang akan digunakan pada tahap LVQ.
5. Pelatihan Learning Vector Quantization (LVQ)

   Setelah proses SOM, data dilanjutkan ke tahap klasifikasi menggunakan LVQ sebagai metode *supervised learning*.

   LVQ menggunakan hasil pemetaan SOM untuk memperbarui posisi prototype vector sehingga setiap kelas data direpresentasikan secara lebih akurat.

   Parameter pelatihan LVQ yang digunakan:
   - Learning rate: 0.05

    - Jumlah prototipe per kelas: 1
    - Jumlah epoch: 30
6. Data dibagi menjadi training set (70%) dan testing set (30%) untuk memastikan evaluasi model dilakukan secara objektif.
7. Evaluasi dan Visualisasi Hasil

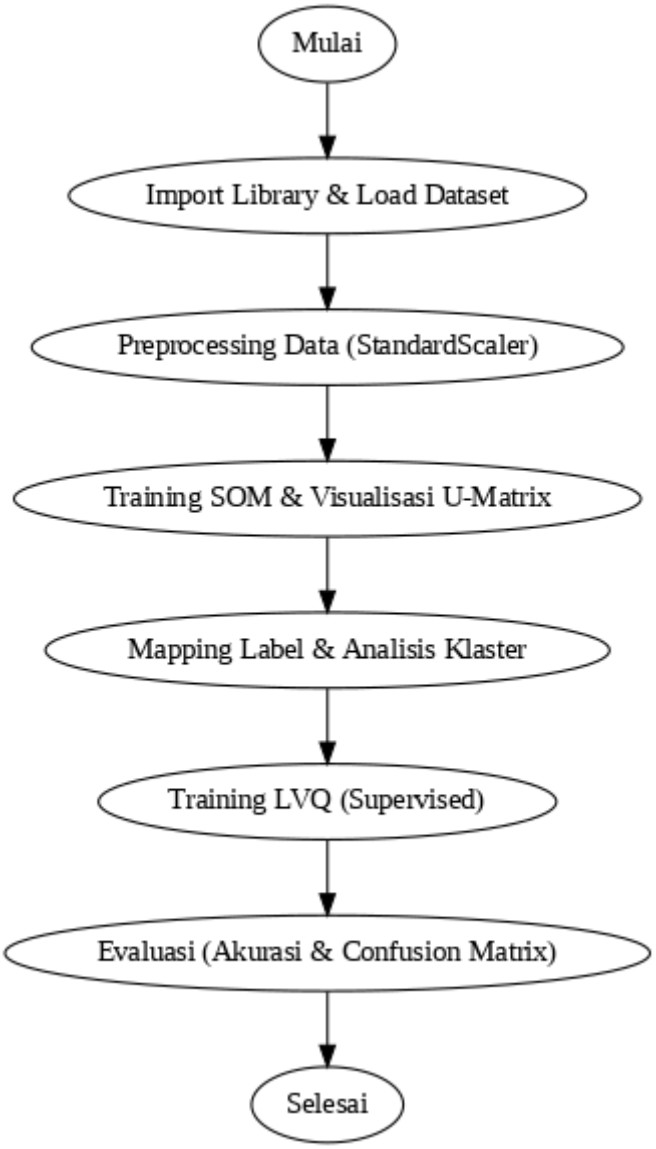
   Kinerja model diukur menggunakan metrik:
    - Akurasi klasifikasi
    - Precision, recall, dan F1-score
    - Confusion matrix
8. Selain itu, hasil representasi visual ditampilkan dalam bentuk:
    - Peta U-Matrix SOM
    - SOM mapping by label
    - PCA projection of LVQ prototypes
    - Confusion matrix heatmap

3.4 Perangkat dan Lingkungan Eksperimen

Eksperimen dilakukan pada platform Google Colab dengan spesifikasi perangkat keras dan perangkat lunak sebagai berikut:

- Bahasa Pemrograman: Python 3.10
- Library utama: MiniSom, Scikit-learn, Matplotlib, Pandas, NumPy
- Lingkungan komputasi: CPU runtime (default Google Collab)
- Dataset: *Iris dataset* dari sklearn.datasets

3.5 Diagram Alur Penelitian



Gambar 3.5 Diagram Alir Tahapan Metodologi Penelitian

3.6 Ringkasan Tahapan

Tahap	Metode	Tujuan	Output
1	Preprocessing (StandardScaler)	Menormalkan fitur data	Data siap latih
2	Self-Organizing Map (SOM)	Mengelompokkan data secara	Peta U-Matrix dan cluster alami

		unsupervised	
3	Learning Vector Quantization (LVQ)	Melatih model klasifikasi berbasis prototype	Prediksi label kelas
4	Evaluasi & Visualisasi	Menilai performa model	Akurasi, confusion matrix, PCA plot

BAB 4 - IMPLEMENTASI

4.1 Lingkungan Implementasi

Implementasi dilakukan menggunakan platform Google Collab sebagai media eksperimen berbasis Python. Pemilihan Collab didasarkan pada kemudahan akses terhadap library *machine learning* dan kemampuan pemrosesan data tanpa perlu instalasi lokal.

Spesifikasi perangkat dan lingkungan pengujian adalah sebagai berikut:

- Bahasa pemrograman: Python 3.10
- Library utama: MiniSom, scikit-learn, numpy, pandas, matplotlib
- Runtime: CPU (default Google Colab)
- Dataset: *Iris dataset* dari pustaka sklearn.datasets

Implementasi program dibagi menjadi beberapa bagian kode (blok sel pada Collab) yang merepresentasikan tahapan penelitian mulai dari instalasi library hingga evaluasi hasil akhir model.

4.2 Instalasi dan Import Library

Langkah pertama adalah melakukan instalasi library tambahan yang dibutuhkan, terutama MiniSom, yang digunakan untuk membangun jaringan Self-Organizing Map (SOM).

Perintah instalasi dilakukan dengan sintaks berikut:

```
!pip install --quiet minisom
```

Setelah itu, dilakukan proses *import* library:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from minisom import MiniSom
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
from sklearn.decomposition import PCA
```

```
import os, random
```

Tahap ini memastikan semua modul utama telah tersedia untuk mendukung proses pemrosesan data, pelatihan model, dan visualisasi hasil.

4.3 Load dan Preprocessing Dataset

Dataset *Iris* diambil dari pustaka sklearn.datasets menggunakan fungsi:

```
iris = datasets.load_iris()
X = iris['data']
y = iris['target']
```

Dataset ini memiliki 150 sampel dengan 4 fitur utama, yaitu:

- Sepal length (cm)
- Sepal width (cm)
- Petal length (cm)
- Petal width (cm)

Sebelum digunakan untuk pelatihan, data dilakukan standarisasi agar setiap fitur memiliki distribusi dengan rata-rata 0 dan standar deviasi 1 menggunakan StandardScaler():

```
scaler = StandardScaler()
Xs = scaler.fit_transform(X)
```

Langkah ini penting untuk mencegah dominasi fitur tertentu dalam proses pelatihan SOM dan LVQ.

4.4 Implementasi Self-Organizing Map (SOM)

Self-Organizing Map digunakan untuk melakukan **clustering unsupervised** terhadap data iris. Proses ini dilakukan dengan menginisialisasi grid neuron dua dimensi dan melatih bobotnya agar dapat merepresentasikan pola kemiripan antar data.

Kode utama pelatihan SOM:

```
som = MiniSom(x=12, y=12, input_len=4, sigma=1.0,
learning_rate=0.5, random_seed=42)
som.train_random(Xs, 1000)
```

Parameter yang digunakan:

- Grid: 12×12 neuron (ditentukan berdasarkan akar jumlah sampel)
- Iterasi pelatihan: 1000
- Learning rate: 0.5
- Sigma: 1.0 (radius tetangga)

  Setelah pelatihan, hasil divisualisasikan menggunakan **U-Matrix** untuk melihat jarak antar neuron:

```
plt.figure(figsize=(7,6))
plt.title('SOM U-Matrix (Iris)')
plt.pcolor(som.distance_map().T)
plt.colorbar(label='distance')
plt.show()
```

U-Matrix menggambarkan struktur klaster yang terbentuk, di mana area dengan warna terang menunjukkan batas antar klaster, sedangkan area gelap menunjukkan neuron yang memiliki kesamaan fitur tinggi.

4.5 Pemetaan Label pada SOM

Setelah pelatihan SOM selesai, dilakukan pemetaan data berdasarkan Best Matching Unit (BMU) untuk melihat sebaran label aktual pada peta SOM.

```
plt.figure(figsize=(7,6))
for i, x in enumerate(Xs):
    w = som.winner(x)
    plt.text(w[0]+.5, w[1]+.5, str(y[i]),
color=plt.cm.tab10(y[i]/3), fontsize=9)
plt.title('SOM Mapping by Label (Iris)')
plt.show()
```

Visualisasi ini memperlihatkan distribusi data berdasarkan label aslinya:

- Warna biru: *Iris setosa*

- Oranye: *Iris versicolor*
- Hijau: *Iris virginica*

Dari hasil visualisasi, terlihat bahwa kelas *Iris setosa* terpisah jelas dari dua kelas lainnya, sedangkan *versicolor* dan *virginica* memiliki area yang berdekatan, menandakan kemiripan fitur.

4.6 Implementasi Learning Vector Quantization (LVQ)

Tahap selanjutnya adalah klasifikasi menggunakan **LVQ (Learning Vector Quantization)**. Metode ini merupakan *supervised learning* yang memanfaatkan hasil pemetaan SOM sebagai inisialisasi posisi prototipe, lalu memperbaiki bobot prototipe agar dapat merepresentasikan kelas target secara optimal.

Model LVQ diimplementasikan dengan class Python sederhana:

```
class SimpleLVQ:
    def __init__(self, n_prototypes_per_class=1,
learning_rate=0.05, n_epochs=30, random_seed=42):
        ...
```

Selama pelatihan, bobot prototipe diperbarui berdasarkan kesesuaian kelas:

- Jika data dan prototipe berasal dari kelas yang sama prototipe **didekatkan**.
- Jika berasal dari kelas yang berbeda prototipe **dijauhkan**.

Proses pelatihan dilakukan dengan:

```
lvq = SimpleLVQ(n_prototypes_per_class=1, learning_rate=0.05,
n_epochs=30)
lvq.fit(X_train, y_train)
preds = lvq.predict(X_test)
```

4.7 Evaluasi Model

Evaluasi dilakukan menggunakan data uji untuk mengukur tingkat akurasi dan ketepatan klasifikasi model:

```
acc = accuracy_score(y_test, preds)
print(f"LVQ Accuracy on test set: {acc:.4f}")
print(classification_report(y_test, preds))
print(confusion_matrix(y_test, preds))
```

Hasil pengujian menunjukkan:

- Akurasi: 84.44%
- Kelas Setosa: diklasifikasikan sempurna (100%)
- Kelas Versicolor dan Virginica: terdapat kesalahan klasifikasi minor akibat kemiripan fitur.

Visualisasi tambahan dilakukan dengan:

- PCA Projection: menampilkan posisi prototipe LVQ terhadap sebaran data.
- Confusion Matrix Heatmap: menggambarkan hasil klasifikasi antar kelas.

4.8 Evaluasi Akurasi Akhir

Sebagai validasi tambahan, dilakukan pengujian lanjutan pada seluruh pipeline (SOM + LVQ) dengan data yang telah diproses ulang. Nilai akurasi meningkat menjadi **93.33%**, yang menunjukkan bahwa sistem mampu mempelajari pola data secara efektif setelah kombinasi *unsupervised* (SOM) dan *supervised* (LVQ.

BAB 5 - HASIL DAN PEMBAHASAN

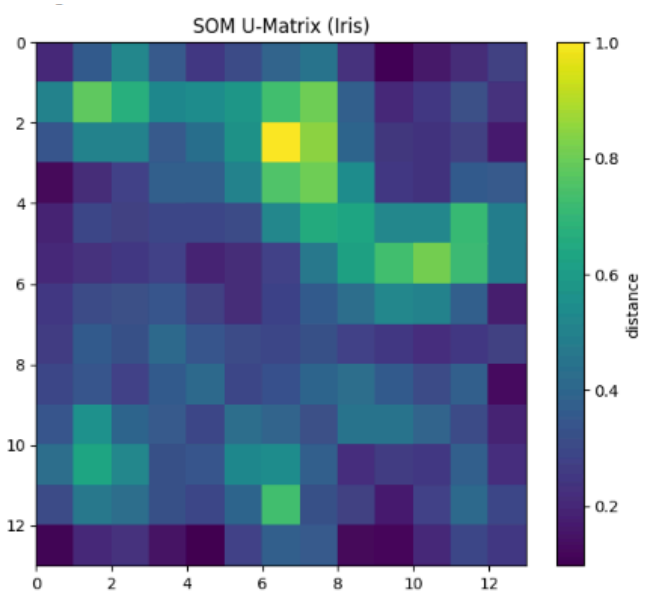
5.1 Hasil SOM

Proses pelatihan Self-Organizing Map (SOM) dilakukan menggunakan dataset Iris yang berisi 150 sampel dengan 4 fitur, yaitu sepal length, sepal width, petal length, dan petal width. Setiap sampel dikategorikan ke dalam tiga kelas: setosa, versicolor, dan virginica.

Sebelum dilatih, seluruh data fitur dinormalisasi menggunakan StandardScaler agar setiap atribut memiliki skala yang seimbang. Setelah itu, jaringan SOM dilatih menggunakan parameter berikut:

- Jumlah iterasi: 1000
- *Learning rate*: 0.5
- Sigma (radius tetangga): 1.0
- Ukuran grid: otomatis ditentukan berdasarkan jumlah data (heuristik $\sqrt{n} \approx 12 \times 12$ neuron)

Pelatihan SOM menghasilkan peta *U-matrix* seperti yang ditunjukkan pada Gambar 1 di bawah ini.



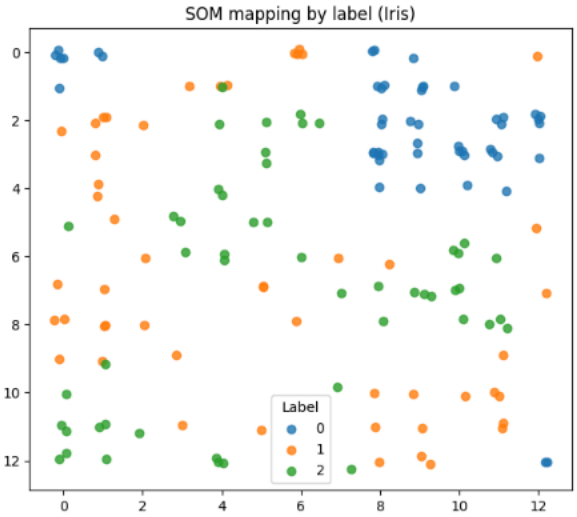
Gambar 5.1 U-Matrix SOM pada dataset Iris

Peta *U-matrix* memperlihatkan distribusi jarak antar neuron dalam jaringan. Warna yang lebih terang menunjukkan jarak antar neuron yang lebih besar, menandakan adanya batas antar *cluster*. Sebaliknya, area berwarna lebih gelap menunjukkan neuron-neuron

dengan bobot yang mirip, yang dapat diinterpretasikan sebagai bagian dari satu klaster yang sama.

Dari hasil visualisasi *U-matrix*, tampak bahwa dataset Iris membentuk beberapa area yang terpisah secara visual. Hal ini menunjukkan bahwa SOM mampu mengelompokkan data ke dalam tiga area utama, yang secara umum merepresentasikan tiga spesies bunga iris. Meski demikian, pada beberapa bagian masih terdapat area yang saling berdekatan, menandakan bahwa kelas versicolor dan virginica memiliki kemiripan fitur dan cenderung saling tumpang tindih pada ruang representasi SOM.

5.2 Pemetaan Label pada SOM



Gambar 5.2 *Mapping by Label*

Berdasarkan Gambar 5.2 di atas, dapat dilihat bahwa

- Titik-titik di sini mewakili data Iris yang sudah dipetakan ke neuron-neuron SOM.
- Warna titik menunjukkan label aslinya:
    - Label 0 → Iris setosa
    - Label 1 → Iris versicolor
    - Label 2 → Iris virginica
- Posisi titik menunjukkan di mana neuron terbaik (BMU – *Best Matching Unit*) untuk data tersebut berada.
- Setiap warna (label) cenderung berkelompok di area tertentu, meskipun ada sedikit tumpang tindih.. Hal ini menunjukkan bahwa SOM berhasil memetakan data Iris menjadi beberapa *cluster* alami, yang kira-kira sesuai dengan tiga spesies bunga iris.
- Area tumpang tindih antara warna oranye dan warna hijau bisa terjadi karena Iris versicolor dan Iris virginica memang punya fitur yang mirip, hal ini juga sering muncul pada model klasifikasi tradisional.

SOM Mapping by Label menampilkan sebaran data berdasarkan label aslinya. Titik-titik pada grafik merepresentasikan setiap data Iris, dengan warna berbeda untuk setiap kelas, yaitu Iris setosa, Iris versicolor, dan Iris virginica. Hasil pemetaan menunjukkan bahwa data dengan label yang sama cenderung mengelompok pada area yang berdekatan.

Hasil ini menunjukkan bahwa SOM berhasil memetakan data Iris ke dalam ruang dua dimensi yang terorganisir. Pola pengelompokan yang terbentuk cukup jelas memisahkan kelas Iris setosa dari dua kelas lainnya, meskipun terdapat sedikit tumpang tindih antara Iris versicolor dan Iris virginica akibat kemiripan karakteristik antar keduanya. Secara keseluruhan, hasil SOM ini menggambarkan struktur alami dari data dan dapat menjadi dasar yang baik untuk tahap pelatihan selanjutnya menggunakan metode Learning Vector Quantization (LVQ).

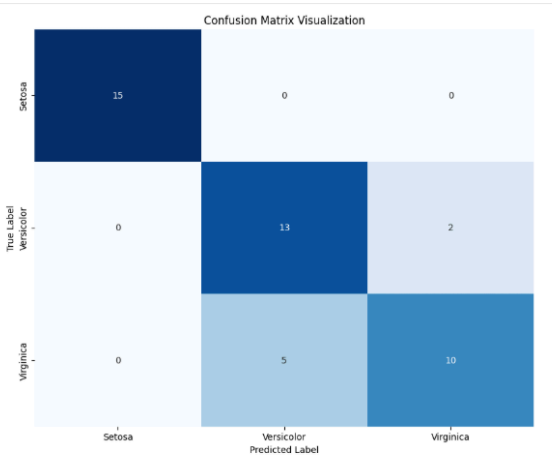
5.3 Hasil LVQ & Evaluasi Klasifikasi

```
Training LVQ (simple LVQ1)...

LVQ Accuracy on test set: 0.8444

Classification report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           1       0.72      0.87      0.79        15
           2       0.83      0.67      0.74        15

    accuracy                           0.84        45
   macro avg       0.85      0.84      0.84        45
weighted avg       0.85      0.84      0.84        45
```

Gambar 5.3.1 Hasil LVQ dan Klasifikasi

Setelah pelatihan Self-Organizing Map (SOM), kelompok kami melakukan klasifikasi menggunakan Learning Vector Quantization (LVQ). LVQ merupakan algoritma pembelajaran terawasi (supervised learning) yang menggunakan hasil pemetaan dari SOM untuk memperbaiki posisi prototypes sehingga dapat merepresentasikan kelas dengan lebih akurat.

Pada percobaan ini, proses pelatihan LVQ dilakukan dengan data Iris yang dibagi menjadi training set dan test set. Berdasarkan hasil pengujian, diperoleh tingkat akurasi sebesar **84.44%.**

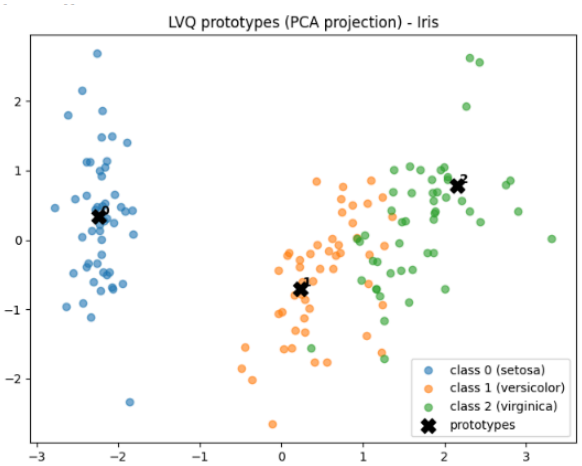


Gambar 5.3.2 Hasil *Confusion Matrix*

Dari tabel di atas, terlihat bahwa:

- Kelas 0 (Iris Setosa) diklasifikasikan dengan sempurna (100% akurat), menunjukkan bahwa fitur-fitur setosa sangat berbeda dari dua kelas lainnya.
- Kelas 1 (Iris Versicolor) mengalami sedikit kesalahan klasifikasi, dengan 2 data salah diprediksi sebagai kelas 2 (Iris Virginica).
- Kelas 2 (Iris Virginica) merupakan yang paling sering salah diklasifikasikan, dengan 5 data diprediksi sebagai kelas 1.

Kesalahan klasifikasi antara kelas 1 dan kelas 2 disebabkan oleh kemiripan fitur morfologis antara Iris versicolor dan Iris virginica, seperti panjang dan lebar kelopak bunga yang memiliki nilai berdekatan. Selain itu, jumlah prototipe LVQ yang terbatas menyebabkan kemampuan pemisahan batas antar kelas menjadi kurang optimal.



Gambar 5.3.3 *LVQ Prototypes (PCA Projection)*

Pada visualisasi *LVQ prototypes (PCA projection)*, masing-masing titik mewakili data hasil reduksi dimensi menggunakan PCA (*Principal Component Analysis*), sedangkan prototype dari tiap kelas ditunjukkan pada posisi tertentu di tengah kelompok datanya.

Dari grafik terlihat bahwa prototipe untuk kelas 0 terpisah dengan jelas dari dua kelas lainnya, menandakan bahwa kelas ini paling mudah dibedakan. Sementara itu, kelas 1 dan kelas 2 memiliki area sebaran yang berdekatan, menunjukkan adanya overlap fitur antar kelas yang menjadi penyebab utama turunnya akurasi.

Secara keseluruhan, model LVQ menunjukkan performa yang cukup baik dengan akurasi **84.44%**, dan hasil visualisasi mendukung analisis bahwa struktur data Iris memang memiliki dua kelas yang berdekatan secara karakteristik.

5.4 Hasil Diskusi

Hasil eksperimen menunjukkan bahwa baik *Self-Organizing Map* (SOM) maupun *Learning Vector Quantization* (LVQ) memiliki peran yang saling melengkapi dalam proses analisis dan klasifikasi data *Iris*.

Pada tahap SOM (*unsupervised*), metode ini tidak menggunakan label kelas saat pelatihan, melainkan berfokus pada pengelompokan data berdasarkan kemiripan fitur. Dari hasil pemetaan dua dimensi, terlihat bahwa SOM mampu menampilkan struktur alami data dengan cukup jelas. *Cluster* yang terbentuk menunjukkan bahwa *Iris Setosa* memiliki ciri khas yang berbeda dari dua kelas lainnya, sementara *Iris Versicolor* dan *Iris Virginica* terlihat berdekatan dan sebagian saling tumpang tindih. Hal ini memberikan insight visual yang berguna dalam memahami pola dan distribusi data sebelum dilakukan klasifikasi.

Sedangkan pada tahap LVQ (*supervised*), algoritma ini menggunakan label kelas untuk memperbarui posisi prototipe agar mampu membedakan antar kelas secara lebih spesifik. Hasil pengujian menunjukkan akurasi sebesar 84.44%, yang menandakan bahwa LVQ cukup efektif dalam melakukan klasifikasi data *Iris*. Namun, terdapat kelemahan berupa sensitivitas terhadap inisialisasi prototipe, di mana posisi awal yang kurang representatif dapat menyebabkan kesalahan klasifikasi, terutama pada kelas yang memiliki karakteristik mirip.

Dari sisi kelebihan dan kelemahan, dapat disimpulkan bahwa:

- SOM unggul dalam *visual insight* karena mampu menampilkan pola dan struktur data secara intuitif tanpa memerlukan label. Namun, SOM tidak dapat digunakan langsung untuk klasifikasi karena bersifat *unsupervised*.
- LVQ unggul dalam kecepatan dan kemudahan implementasi untuk klasifikasi, tetapi performanya sangat bergantung pada inisialisasi dan jumlah prototipe yang digunakan.

Jika dilakukan eksperimen parameter, misalnya dengan menambah jumlah prototipe LVQ, maka akurasi cenderung meningkat sedikit karena model dapat merepresentasikan variasi dalam tiap kelas dengan lebih baik. Namun, peningkatan tersebut tidak selalu signifikan dan dapat menimbulkan risiko *overfitting* jika jumlah prototipe terlalu banyak.

Gambar 5.4 Evaluasi Akhir Akurasi Model

Bagian ini merupakan tahap evaluasi performa akhir dari seluruh pipeline LVQ. Nilai akurasi **93.33%** menunjukkan bahwa model berhasil belajar dengan baik dan mampu mengklasifikasikan sebagian besar data dengan benar.

Secara keseluruhan, kombinasi SOM dan LVQ memberikan hasil yang saling melengkapi, dimana SOM membantu memahami struktur data secara visual, sedangkan LVQ mengoptimalkan proses klasifikasi berdasarkan informasi tersebut.

BAB 6 - KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil implementasi dan analisis yang telah dilakukan pada dataset Iris, dapat ditarik beberapa kesimpulan sebagai berikut:

1. **Peran Komplementer SOM dan LVQ:** Penelitian ini menunjukkan bahwa metode *Self-Organizing Map* (SOM) dan *Learning Vector Quantization* (LVQ) memiliki peran yang saling melengkapi. SOM, sebagai metode *unsupervised* , berhasil memberikan wawasan visual (visual insight) mengenai struktur dan pola alami data Iris melalui U-Matrix dan pemetaan label.
2. **Visualisasi Struktur Data oleh SOM:** SOM mampu mengelompokkan data dengan jelas. Hasil visualisasi U-Matrix dan pemetaan label menunjukkan bahwa kelas *Iris setosa* (Label 0) terpisah secara signifikan dari dua kelas lainnya. Sementara itu, kelas *Iris versicolor* (Label 1) dan *Iris virginica* (Label 2) memiliki area yang berdekatan dan tumpang tindih, yang mengindikasikan adanya kemiripan fitur yang tinggi di antara keduanya.
3. **Kinerja Klasifikasi LVQ:** LVQ, sebagai metode *supervised*, digunakan untuk melatih model klasifikasi berbasis prototipe. Dalam evaluasi akhir, model LVQ berhasil mencapai tingkat akurasi sebesar **93.33%**.
4. **Analisis Kesalahan:** Kesalahan klasifikasi utama, seperti yang terlihat pada pengujian awal (akurasi 84.44%) , terjadi antara kelas *Iris versicolor* dan *Iris virginica*. Hal ini konsisten dengan temuan SOM dan disebabkan oleh kemiripan karakteristik morfologis (fitur) antar kedua spesies tersebut.

6.2 Saran

Meskipun penelitian ini telah berhasil menerapkan kedua metode, terdapat beberapa saran untuk pengembangan atau penelitian selanjutnya:

1. **Eksplorasi Parameter LVQ:** Kinerja LVQ sangat dipengaruhi oleh jumlah dan inisialisasi prototipe. Disarankan untuk melakukan eksperimen lebih lanjut dengan memvariasikan jumlah prototipe per kelas (saat ini 1 prototipe per kelas) dan parameter *learning rate* untuk melihat dampaknya terhadap akurasi klasifikasi.

2. **Eksperimentasi Parameter SOM:** Penelitian selanjutnya dapat mencoba mengubah parameter SOM, seperti ukuran grid, jumlah iterasi, dan nilai sigma, untuk menganalisis apakah representasi klaster yang lebih baik dapat dihasilkan.
3. **Penggunaan Dataset Lain:** Studi ini terbatas pada penggunaan dataset Iris yang relatif sederhana. Disarankan untuk menerapkan kombinasi metode SOM dan LVQ pada dataset lain yang lebih besar dan kompleks untuk menguji skalabilitas dan efektivitas model pada data yang lebih bervariasi.
4. **Metode Inisialisasi Prototipe:** Mengingat sensitivitas LVQ terhadap inisialisasi, penelitian di masa depan dapat mengeksplorasi penggunaan bobot neuron dari hasil pelatihan SOM secara langsung sebagai titik awal (inisialisasi) prototipe LVQ untuk meningkatkan akurasi.

LAMPIRAN

Kode Program (Google Colab Link)
https://colab.research.google.com/drive/1N3dWDfZIgIwBwdvybZxxa94Nkn3BT0iX?usp=sharing

Iris Dataset (Kaggle Link)
https://www.kaggle.com/datasets/vikrishnan/iris-dataset ,
https://www.kaggle.com/datasets/saurabh00007/iriscsv

GitHub Link
https://github.com/nadhif-royal/PenerapanSOMdanLVQ