Penerapan Metode *Learning Vector Quantization* untuk Prediksi Indikasi Diabetes

Oleh:

Valentino Hariyanto	**222410101023**
Elvira Vanny Rahmasari	**222410101079**
Divo Tahta Imannulloh	**222410101083**

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2024

DAFTAR ISI

BAB 1 PENDAHULUAN

1.1 Judul Penelitian

Penerapan Metode *Learning Vector Quantization* untuk Prediksi Indikasi Diabetes

1.2 Latar Belakang

Diabetes merupakan salah satu penyakit kronis yang paling umum dan serius di dunia, dengan angka prevalensi yang terus meningkat setiap tahunnya. Menurut data dari IDF Diabetes Atlas, prevalensi diabetes secara global diperkirakan akan meningkat dari 415 juta pada tahun 2015 menjadi 642 juta pada tahun 2040 (Ogurtsova et al., 2017). Kondisi ini tidak hanya membawa dampak negatif terhadap kualitas hidup individu yang terkena, tetapi juga menimbulkan beban ekonomi yang signifikan bagi sistem kesehatan global.

Deteksi dini dan pengelolaan diabetes yang efektif adalah kunci untuk mengurangi komplikasi serius yang berkaitan dengan penyakit ini, seperti penyakit jantung, stroke, dan kerusakan ginjal. Teknologi dan metode analisis data telah menjadi alat yang sangat penting dalam membantu prediksi dan diagnosis dini diabetes. Dalam konteks ini, metode machine learning, khususnya *Learning Vector Quantization* (LVQ), telah menunjukkan potensi yang signifikan.

Learning Vector Quantization (LVQ) adalah salah satu metode pembelajaran mesin yang berbasis pada jaringan saraf tiruan. Metode ini terkenal karena kemampuannya dalam melakukan klasifikasi dengan akurasi tinggi melalui proses pembelajaran yang terawasi (supervised learning). LVQ bekerja dengan cara memetakan input data ke dalam vektor-vektor tertentu yang mewakili kelas-kelas yang berbeda. Dengan demikian, metode ini sangat cocok untuk tugas-tugas prediksi seperti deteksi indikasi diabetes.

Penggunaan LVQ dalam prediksi diabetes menawarkan beberapa keuntungan, antara lain kemampuan untuk menangani data yang besar dan kompleks, serta kemampuannya untuk melakukan klasifikasi yang efisien dengan waktu komputasi yang relatif singkat. Namun, untuk mencapai hasil yang optimal, penting untuk melakukan penyesuaian parameter dan pemilihan fitur yang tepat dalam proses pelatihan model.

Dengan memahami dan mengoptimalkan penggunaan metode LVQ dalam prediksi indikasi diabetes, diharapkan penelitian ini dapat memberikan kontribusi yang signifikan dalam upaya deteksi dini diabetes serta mendukung pengambilan keputusan klinis yang lebih baik dalam pengelolaan penyakit ini.

1.3 Rumusan Masalah

Berdasarkan permasalahan pada latar belakang, maka rumusan masalah pada penelitian ini dapat direvisi sebagai berikut:

1. Bagaimana mengimplementasikan metode *Learning Vector Quantization* (LVQ) untuk memprediksi indikasi status diabetes pada dataset yang diberikan?
2. Bagaimana kinerja model klasifikasi yang dibangun menggunakan metode LVQ dalam memprediksi indikasi status diabetes (menderita atau tidak menderita diabetes) berdasarkan fitur-fitur yang tersedia dalam dataset?

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dibuat, berikut adalah tujuan dari penelitian:

1. Mengimplementasikan metode *Learning Vector Quantization* (LVQ) untuk memprediksi indikasi status diabetes pada dataset yang diberikan.
2. Mengetahui akurasi model klasifikasi yang dibangun menggunakan metode LVQ dalam memprediksi indikasi status diabetes (menderita atau tidak menderita diabetes) berdasarkan fitur-fitur dalam dataset.

1.5 Batasan Penelitian

Batasan masalah dari penelitian ini yaitu:

1. Dataset yang digunakan adalah dataset diabetes dari Kaggle.
2. Penelitian menggunakan metode *Learning Vector Quantization* (LVQ) sebagai algoritma klasifikasi utama.
3. Penelitian ini berfokus pada penggunaan metode LVQ untuk memprediksi indikasi status diabetes (menderita atau tidak menderita diabetes).

4. Penelitian ini tidak mempertimbangkan faktor-faktor eksternal lainnya yang mungkin mempengaruhi diabetes, seperti gaya hidup atau riwayat kesehatan keluarga.
5. Evaluasi hasil akan dilakukan untuk menilai kinerja model klasifikasi LVQ dalam memprediksi indikasi status diabetes berdasarkan fitur-fitur yang tersedia dalam dataset.

1.6 Manfaat Penelitian

Berdasarkan tujuan penelitian yang dibuat sebelumnya, maka manfaat dari penelitian ini yaitu:

1. Penelitian ini dapat memberikan kontribusi dalam meningkatkan efektivitas prediksi indikasi dini penyakit diabetes dengan mengimplementasikan metode *Learning Vector Quantization* (LVQ) untuk memprediksi indikasi status diabetes.
2. Penelitian ini dapat membuka wawasan bagi pengembangan metode analisis lanjutan dalam memahami dan memprediksi indikasi penyakit diabetes. Dengan menggabungkan teknik-teknik *machine learning* seperti LVQ untuk prediksi status diabetes, dapat melakukan pengembangan lebih lanjut dalam bidang ini yang dapat diterapkan pada skala yang lebih luas.

BAB 2 TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

No.	Peneliti	Tahun	Judul	Rangkuman Singkat
1.	Abdul Aziz, Fitri Insani, Jasril, Fadhilah Syafria	2023	Implementasi Metode *Learning Vector Quantization* (LVQ) Untuk Klasifikasi Keluarga Beresiko Stunting	Penelitian ini mengimplementasikan metode *Learning Vector Quantization* (LVQ) untuk mengklasifikasikan keluarga beresiko stunting di Kampung KB Tunas Harapan. LVQ digunakan karena kemampuannya dalam mengolah data besar dengan cepat. Pengujian dilakukan dengan variasi parameter LVQ, termasuk jumlah neuron input, jenis jarak, learning rate, jumlah epoch, dan persentase data latih.

				Hasil pengujian menunjukkan bahwa konfigurasi LVQ dengan jumlah neuron input sebanyak 7, menggunakan Chebychev distance sebagai jenis jarak, learning rate 0.1, jumlah epoch 7, dan 30% data latih memberikan akurasi mencapai 99.83%. Hasil ini menunjukkan bahwa LVQ efektif dalam meningkatkan akurasi dalam mengidentifikasi keluarga yang berpotensi mengalami stunting. Penelitian ini memiliki implikasi

				positif dalam upaya pengentasan stunting di Kampung KB Tunas Harapan dan dapat menjadi kontribusi penting dalam peningkatan program kesehatan masyarakat. Saran untuk penelitian selanjutnya adalah melakukan perbandingan dengan metode klasifikasi lainnya untuk mendapatkan pemahaman yang lebih baik tentang efektivitas LVQ dalam konteks ini.
2.	Elvia Budianita, Widodo Prijodiprodjo	2013	Penerapan *Learning Vector Quantization* (LVQ) untuk Klasifikasi Status Gizi Anak	Penelitian ini menggunakan metode *Learning Vector Quantization* (LVQ) dan algoritma pengembangannya,

				LVQ3, untuk mengklasifikasikan status gizi anak berdasarkan indeks berat badan menurut tinggi badan (BB/TB). Pengujian dilakukan dengan menggunakan variabel seperti jenis kelamin, berat badan, tinggi badan, penyakit infeksi, nafsu makan, dan pekerjaan kepala keluarga (KK). Hasil penelitian menunjukkan bahwa LVQ3 lebih baik dalam mengklasifikasikan status gizi anak dibandingkan dengan LVQ1. Penggunaan parameter window (ε) pada LVQ3

				memberikan pengaruh positif terhadap performa klasifikasi, meningkatkan akurasi hingga 100% dalam kasus ini. Jumlah data latih juga mempengaruhi akurasi, dimana semakin banyak data latih, semakin tinggi akurasi yang dicapai. Kesimpulan dari penelitian ini adalah bahwa LVQ dapat mengenali pola dan mengklasifikasikan status gizi anak dengan baik. LVQ3 dengan parameter yang tepat dapat meningkatkan performa klasifikasi, dan jumlah data latih

				yang mencukupi sangat penting untuk mencapai akurasi yang tinggi. Penelitian ini memberikan kontribusi penting dalam penilaian status gizi anak dan dapat digunakan sebagai referensi untuk penelitian lebih lanjut dalam bidang ini.

2.2 Landasan Teori

2.2.1 Diabetes

Diabetes adalah penyakit kronis yang disebabkan oleh resistensi insulin atau kekurangan insulin, yang dapat menyebabkan tingginya konsentrasi glukosa darah. Diabetes dapat menyebabkan berbagai komplikasi, seperti neuropati, nefropati, dan retinopati, jika tidak diobati atau tidak diatur dengan baik.

2.2.2 Faktor-Faktor Risiko Diabetes

Faktor-faktor risiko diabetes meliputi usia, kebiasaan makan, aktivitas fisik, dan riwayat kesehatan keluarga. Orang yang memiliki riwayat kesehatan keluarga dengan diabetes memiliki risiko lebih tinggi mengalami diabetes. Selain itu, orang yang memiliki berat badan yang berlebihan, jarang berolahraga, dan memiliki pola makan yang tidak seimbang juga memiliki risiko lebih tinggi mengalami diabetes.

2.2.3 Data mining

Data mining merupakan proses ekstraksi pengetahuan yang tersembunyi, pola tersembunyi, dan informasi yang menarik dari suatu basis data. Hal ini melibatkan berbagai teknik yang mencakup pemrosesan data, pembelajaran mesin, dan statistik (Han, Kamber, & Pei, 2011).

Tujuan utama dari data mining adalah untuk menemukan pola yang bermanfaat dan dapat diprediksi dalam data, sehingga membantu dalam pengambilan keputusan yang lebih baik dan efisien (Witten, Frank, & Hall, 2011).

Proses data mining terdiri dari beberapa tahapan, yaitu pemilihan data, preprocessing data, transformasi data, data mining, interpretasi, dan evaluasi data mining (Tan, Steinbach, & Kumar, 2006).

Ada banyak algoritma yang digunakan dalam data mining, seperti K-Means, Apriori, Decision Tree, LVQ, dan lain-lain. Algoritma-algoritma ini digunakan untuk menemukan pola dalam data (Witten, Frank, & Hall, 2011).

Data mining telah banyak diterapkan dalam berbagai bidang, seperti bisnis, kedokteran, keamanan informasi, dan lain-lain. Contohnya adalah dalam prediksi penjualan, indikasi penyakit, dan deteksi kecurangan (Han, Kamber, & Pei, 2011).

2.2.4 Klasifikasi

Klasifikasi secara umum mengacu pada proses pengelompokan atau pengkategorian suatu objek ke dalam kelas atau kategori tertentu berdasarkan karakteristik atau atribut yang dimiliki oleh objek tersebut. Dalam konteks data mining, klasifikasi adalah salah satu teknik yang digunakan untuk melakukan prediksi atau estimasi terhadap kelas atau label dari objek berdasarkan informasi yang terdapat pada data training.

Menurut Fatichah (2015), klasifikasi dalam data mining adalah proses pengelompokan data ke dalam kategori-kategori berdasarkan atribut-atribut tertentu. Tujuannya adalah untuk mengklasifikasikan data baru ke dalam kelas yang sudah ada. Algoritma klasifikasi yang umum digunakan antara lain Decision Tree, k-Nearest Neighbors (k-NN), dan Support Vector Machine (SVM).

2.2.5 *Learning Vector Quantization* (LVQ)

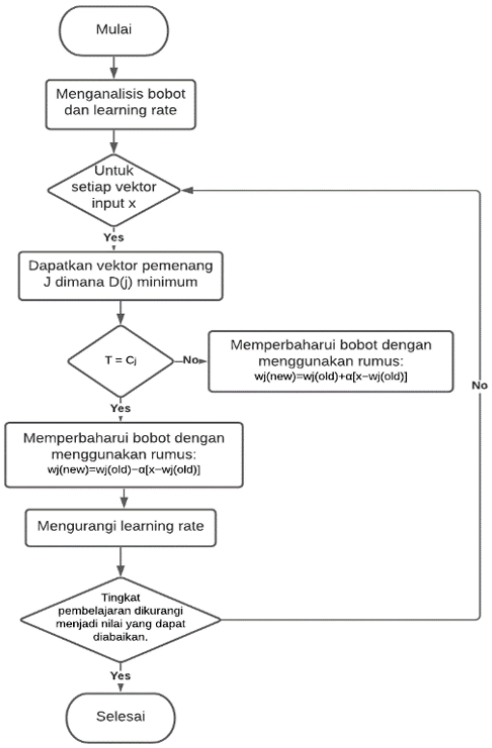
Learning Vector Quantization (LVQ) adalah algoritma klasifikasi terarah berbasis prototipe yang terinspirasi oleh model sistem saraf biologis. LVQ merupakan turunan terawasi dari sistem kuantisasi vektor dan melatih jaringannya melalui algoritma pembelajaran kompetitif yang mirip dengan Self Organizing Map (SOM) (Akbari, et al., 2017).

LVQ bekerja dengan mengklasifikasikan data masukan berdasarkan kedekatannya dengan prototipe yang telah dilatih. Prototipe ini mewakili kelas-kelas yang berbeda dalam data pelatihan. Algoritma LVQ memperbarui prototipe secara iteratif untuk meningkatkan akurasi klasifikasi.

Berikut adalah proses algoritma LVQ:

1. **Inisialisasi Prototipe:** Pilih prototipe awal untuk setiap kelas.
2. **Klasifikasi Data Masukan:** Hitung jarak antara data masukan dan setiap prototipe. Klasifikasikan data masukan ke kelas prototipe yang terdekat.
3. **Pembaruan Prototipe:** Perbarui prototipe dari kelas data masukan yang diklasifikasikan dengan benar dan kelas data masukan yang diklasifikasikan secara salah.
4. **Iterasi:** Ulangi langkah 2 dan 3 hingga kriteria terminasi terpenuhi.

Berikut adalah gambaran alur dari algoritma LVQ



Gambar 1. Alur Algoritma LVQ

Untuk melakukan pembelajaran LVQ, beberapa parameter yang dibutuhkan antara lain:

1. Vektor Pelatihan (X): Kumpulan data pelatihan (X1, X2, ..., Xn).
2. Target (T): Kelas target yang sesuai dengan vektor pelatihan.
3. Bobot Unit Pelatihan (Wj): Bobot pada setiap unit vektor pelatihan (W1j, W2j, ..., Wnj).
4. Kelas Unit Keluaran (Cj): Kelas yang mewakili unit keluaran ke-j.
5. Learning Rate (α): Tingkat pembelajaran, dengan nilai $0 < \alpha < 1$.

Proses pembaharuan bobot dilakukan berdasarkan kondisi berikut:

- Jika T = Cj, maka bobot Wj diperbaharui dengan rumus: $Wj(t+1) = Wj(t) + \alpha(t)[X(t) - Wj(t)]$.
- Jika $T \neq Cj$, maka bobot Wj diperbaharui dengan rumus: $Wj(t+1) = Wj(t) - \alpha(t)[X(t) - Wj(t)]$.

BAB 3 METODOLOGI PENELITIAN

3.1. Jenis Penelitian

Penelitian ini menggunakan metode terapan, yang menurut *American Educational Research Association* (AERA) adalah jenis penelitian yang dirancang untuk memberikan kontribusi praktis langsung dalam pemahaman, pemecahan masalah, atau pengambilan keputusan dalam konteks nyata. Metode terapan dalam penelitian ini bertujuan untuk memberikan solusi konkret dalam memprediksi indikasi diabetes dengan menerapkan algoritma *Learning Vector Quantization* (LVQ). Melalui penelitian ini, diharapkan dapat ditemukan cara yang efektif untuk mengidentifikasi dan mengklasifikasikan faktor-faktor yang menyebabkan penyakit diabetes sehingga dapat digunakan sebagai dasar untuk pengambilan keputusan dalam pencegahan penyakit.

3.2. Objek Penelitian

Objek penelitian ini adalah berbagai komponen yang berperan dalam memprediksi indikasi penyakit diabetes. Komponen-komponen ini meliputi *Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, Diabetes Pedigree dan Age*. Penelitian ini bertujuan untuk mengklasifikasikan komponen-komponen tersebut berdasarkan dampaknya terhadap prediksi status penyakit diabetes. Dengan memahami bagaimana masing-masing komponen ini dalam mengetahui status penyakit diabetes, penelitian ini dapat membantu dalam mengembangkan prediksi penyakit.

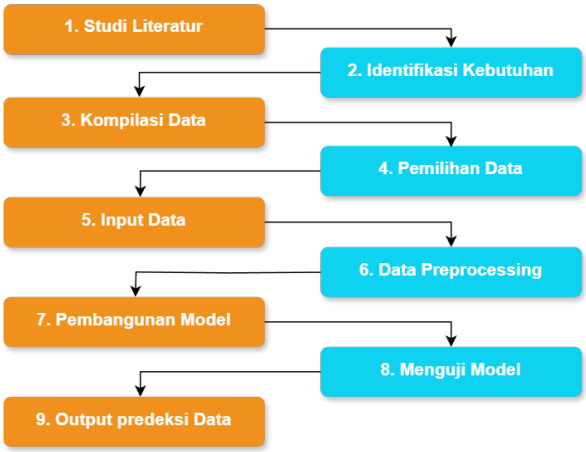
3.3. Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Fakultas Ilmu Komputer, Universitas Jember, dari tanggal 7 Mei 2024 hingga 6 Juni 2024. Pemilihan lokasi ini didasarkan pada ketersediaan fasilitas yang memadai dan dukungan teknis yang diperlukan untuk melaksanakan penelitian. Selama periode penelitian, berbagai kegiatan seperti pengumpulan data, preprocessing data, implementasi algoritma LVQ, serta evaluasi dan analisis hasil akan dilaksanakan secara sistematis dan terstruktur. Waktu yang dialokasikan untuk penelitian ini mencakup semua tahapan yang diperlukan untuk memastikan bahwa hasil yang diperoleh akurat dan dapat diandalkan.

3.4. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah pengumpulan data sekunder. Data sekunder adalah data yang sudah dikumpulkan dan dipublikasikan oleh pihak lain, dalam hal ini dataset yang tersedia di platform Kaggle. Dataset yang digunakan berjudul "Diabetes" dan berisi informasi lengkap mengenai berbagai komponen yang mempengaruhi prediksi indikasi penyakit diabetes.

3.5. Tahapan Penelitian



a. Studi Literatur

Tahap ini melibatkan pengumpulan informasi dari berbagai sumber ilmiah, seperti jurnal, buku, dan publikasi terkait metode *Learning Vector Quantization* (LVQ) serta parameter-parameter yang dapat memprediksi penyakit diabetes. Studi literatur ini bertujuan untuk memahami konsep dasar, teknik yang relevan, serta temuan-temuan sebelumnya yang dapat dijadikan referensi dalam penelitian ini.

b. Identifikasi Kebutuhan

Pada tahap ini, dilakukan identifikasi terhadap kebutuhan penelitian, termasuk pemahaman mendalam tentang komponen-komponen yang dapat memprediksi penyakit diabetes. Selain itu, ditentukan tujuan spesifik dari penelitian, seperti jenis data yang diperlukan dan metode evaluasi yang akan digunakan untuk menilai kinerja model LVQ.

c. Kompilasi Data

Tahap ini melibatkan pengumpulan dataset yang relevan dari berbagai sumber. Dalam penelitian ini, data diambil dari platform Kaggle yang menyediakan dataset mengenai prediksi diabetes. Proses kompilasi ini memastikan bahwa semua data yang diperlukan untuk analisis tersedia dan terorganisir dengan baik.

d. Pemilihan Data

Setelah data terkumpul, dilakukan pemilihan data dengan menyeleksi atribut-atribut yang relevan untuk penelitian. Atribut-atribut yang dipilih mencakup berbagai komponen yang menjadi penentu prediksi penyakit diabetes, seperti *Pregnancies*, *Glucose*, *BloodPressure*, *SkinThickness*, *Insulin*, BMI, *Diabetes Pedigree* dan *Age*. . Pemilihan data yang tepat sangat penting untuk memastikan bahwa analisis yang dilakukan akurat dan relevan.

e. Input Data

Pada tahap ini, data yang telah dipilih dimasukkan ke dalam sistem yang akan digunakan untuk analisis. Data diorganisir dan disusun sedemikian rupa agar siap untuk diproses lebih lanjut dalam tahap-tahap berikutnya.

f. Data Preprocessing

Tahap preprocessing melibatkan pembersihan dan transformasi data agar sesuai dengan format yang diperlukan untuk analisis. Langkah-langkah ini mencakup pengisian nilai yang hilang (missing values), normalisasi data, data duplikat dan penghapusan data yang tidak konsisten. Tujuan dari preprocessing adalah untuk meningkatkan kualitas data sehingga model dapat dilatih dengan lebih efektif.

g. Pembangunan Model

Pada tahap ini, model *Learning Vector Quantization* (LVQ) dibangun menggunakan data yang telah dipreproses. Proses pembangunan model meliputi pemilihan parameter, pelatihan model dengan data latih, dan pengoptimalan model untuk mencapai kinerja yang optimal.

h. Menguji Model

Setelah model dibangun, dilakukan pengujian terhadap model menggunakan data uji yang telah disiapkan. Tujuan pengujian ini adalah untuk mengevaluasi kinerja model dalam melakukan prediksi penyakit diabetes berdasarkan komponen-komponen yang telah dipilih. Hasil pengujian akan menunjukkan seberapa baik model dapat mengidentifikasi dan mengklasifikasikan data baru.

i. Output Prediksi Data

Tahap akhir adalah menghasilkan output prediksi data. Model yang telah teruji akan digunakan untuk memprediksi penyakit diabetes berdasarkan data input baru. Hasil prediksi ini akan dianalisis dan dibandingkan dengan data aktual untuk mengevaluasi akurasi dan efektivitas model dalam mengetahui prediksi penyakit diabetes. Output ini diharapkan dapat memberikan wawasan yang berguna bagi pengelolaan kualitas air yang lebih baik.

3.6. Hasil Implementasi

3.6.1. Atribut

No.	Atribut	Keterangan
1.	*Pregnancies*	Berapa kali orang mengandung
2.	*Glucose*	Tingkat glukosa dalam darah pasien
3.	*BloodPressure*	Tekanan darah pasien

4.	*SkinThickness*	Tingkat ketebalan kulit pasien
5.	*Insulin*	Tingkat insulin dalam darah pasien
6.	*BMI*	Indeks massa tubuh pasien
7.	*DiabetesPedigreeFunction*	Ukuran genetik pasien terhadap diabetes
8.	*Age*	Umur pasien
9.	*Outcome*	Hasil (0 - negatif diabetes, 1 - positif diabetes)

3.6.2. Tahapan dan Hasil Perhitungan

1. Melakukan import library penunjang proses klasifikasi

```
# Import Library Umum
import numpy as np
np.int = int
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Import Library Sklearn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

# Import Library Sklvq
from sklvq import GLVQ
```

Adapun beberapa penjelasan singkat untuk setiap library yang digunakan

1. NumPy: Manipulasi array numerik multidimensi.
2. Pandas: Analisis dan manipulasi data dalam bentuk tabel dua dimensi.

3. Matplotlib: Pembuatan berbagai jenis plot dan grafik dalam Python.
4. Seaborn: Visualisasi data statistik dengan gaya yang menarik secara default.
5. Sklearn: Beragam algoritma machine learning dan fungsi evaluasi untuk proses pemodelan dan evaluasi.
    a. *from sklearn.model_selection import train_test_split*: Digunakan untuk membagi dataset menjadi subset pelatihan dan pengujian.
    b. *from sklearn.preprocessing import MinMaxScaler*: Digunakan untuk penskalaan fitur ke rentang yang ditentukan (misalnya, 0 hingga 1).
    c. *from sklearn.metrics import accuracy_score*: Digunakan untuk mengukur akurasi dari model klasifikasi.
    d. *from sklearn.metrics import classification_report*: Digunakan untuk menghasilkan laporan evaluasi klasifikasi dalam bentuk teks yang mencakup presisi, recall, f1-score, dan support untuk setiap kelas.
6. Sklvq: Sebuah library khusus untuk model neural network LVQ
    a. *from sklvq import GLVQ*: Digunakan untuk mengimport kelas GLVQ dari library sklvq, memungkinkan penggunaan model Generalized *Learning Vector Quantization* dalam pengembangan aplikasi berbasis Python.

2. Membaca dataset

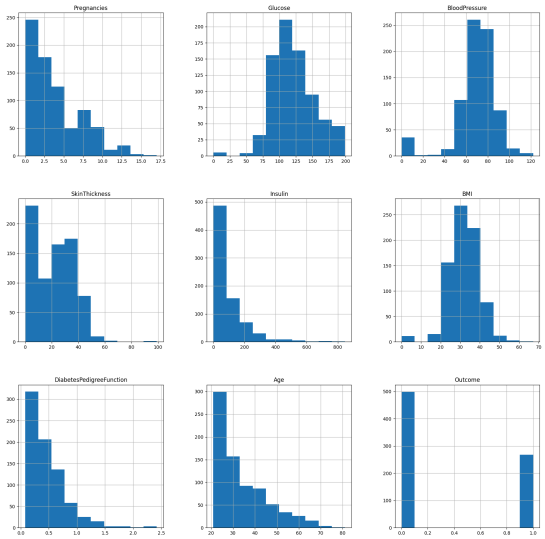
Pembacaan dataset dilakukan menggunakan fungsi dari library pandas yaitu read_csv. Berikut implementasinya

```
# Membaca dataset
dataset = pd.read_csv('diabetes.csv')
dataset.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

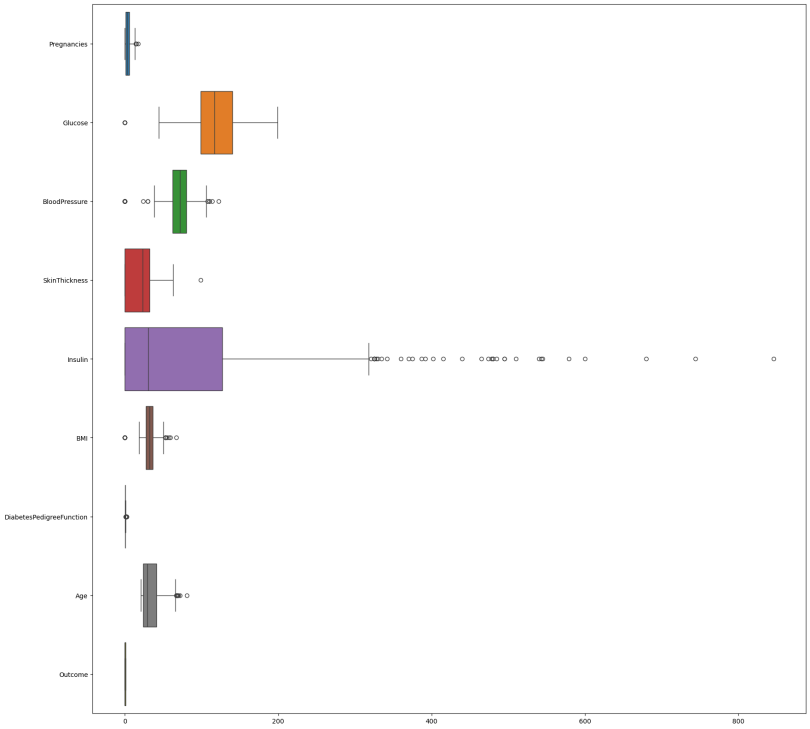
3. Visualisasi data

   a. Histogram



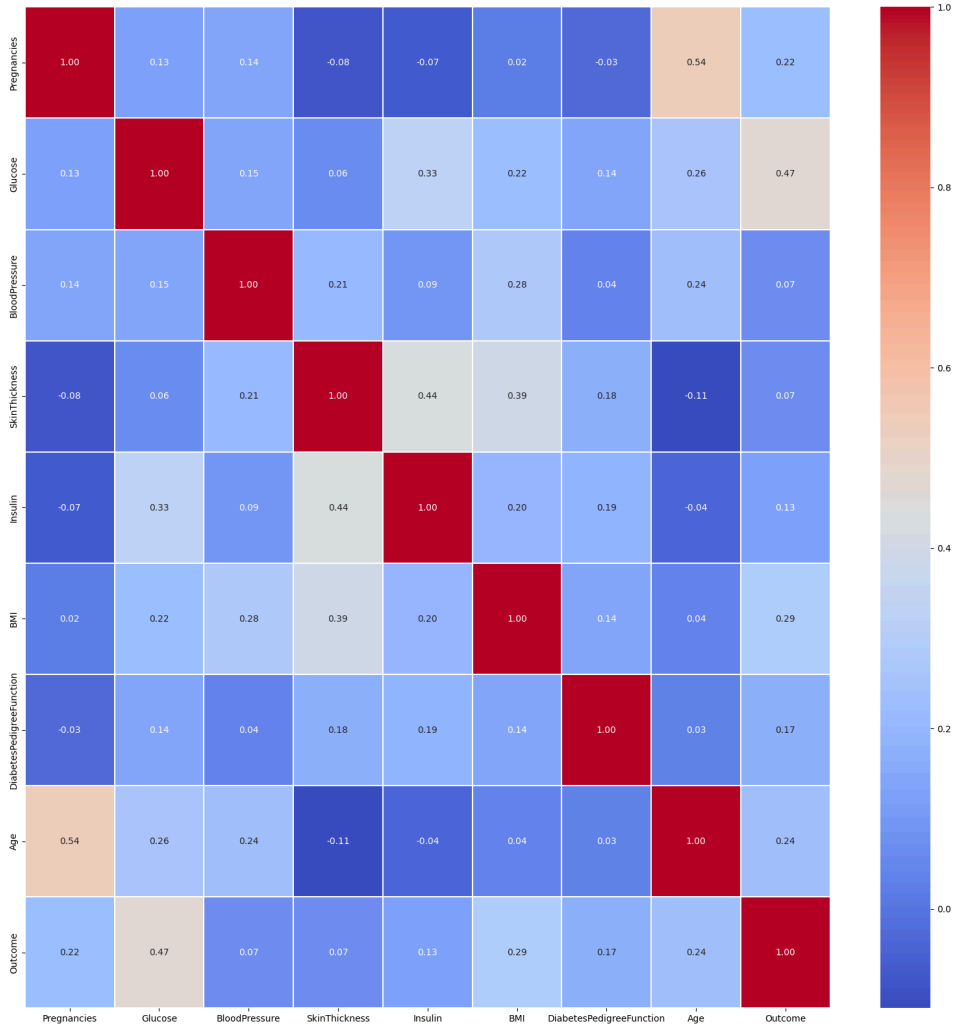
Histogram digunakan untuk melihat persebaran atau distribusi data yang digunakan. Dari histogran di atas, dapat dilihat bahwa terdapat beberapa fitur atau kolom yang distribusi nya cenderung ke kiri, kanan, dan belum ada yang benar - benar normal.

b. Boxplot



Boxplot memiliki fungsi yang kurang lebih sama seperti histogram, namun disini terdapat hal lain yang ingin ditemukan menggunakan boxplot, yaitu outlier. Ada beberapa data yang memiliki outlier, namun setelah ditelusuri lebih lanjut, ternyata outlier tersebut adalah data yang benar.

c. Korelasi attribute



Gambar ini menampilkan sebuah matriks korelasi, yang menunjukkan koefisien korelasi antara berbagai variabel atau fitur. Koefisien korelasi memiliki rentang nilai dari -1 hingga 1, di mana -1 mengindikasikan korelasi negatif sempurna, 0 mengindikasikan tidak ada korelasi, dan 1 mengindikasikan korelasi positif sempurna.

Matriks ini diberi kode warna, dengan warna merah mewakili korelasi positif dan warna biru mewakili korelasi negatif. Semakin gelap warnanya, semakin kuat korelasinya.

Matriks atau grafik ini menunjukkan bahwa:

1. Variabel "Outcome" (kolom terakhir) memiliki korelasi positif dengan hampir semua variabel atau kolom lainnya, mengindikasikan bahwa nilai yang lebih tinggi dari variabel-variabel ini terkait dengan risiko yang lebih tinggi terhadap outcome (yang berarti akan terdeteksi sebagai mengidap diabetes).
2. Variabel "Insulin" memiliki korelasi yang relatif lemah dengan variabel lain, menunjukkan bahwa variabel ini mungkin tidak terlalu terkait dengan fitur lainnya.
3. Terdapat beberapa korelasi positif yang kuat di antara variabel tertentu, seperti "Pregnancies" dan "Glucose", "BloodPressure" dan "SkinThickness", serta "BMI" dan "DiabetesPedigreeFunction".
4. Terdapat juga beberapa korelasi negatif yang cukup signifikan, seperti antara "Glucose" dan "BMI", serta antara "BloodPressure" dan "Age".

4. Melakukan data cleaning
   a. Cek data *NULL* atau *NaN*

      Setelah dilakukan pembacaan dataset, maka dilakukan cek apakah terdapat data yang kosong atau biasa disebut *missing values*.

```
The number of missing value on dataset:
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Tidak ada data yang kosong atau *missing values*.

b. Cek data duplikasi

Selanjutnya dilakukan cek data yang terduplikasi atau data yang sama persis.

```
Is there any duplicated data?
0
```

Ternyata tidak ada data yang terduplikasi.

c. Pemisahan data dan label

Dilakukan pemisahan antara data penentu atau komponen penentu dan komponen target atau label, dimana komponen label terletak di kolom paling akhir dalam dataset. Maka digunakan kode sebagai berikut.

```
data = dataset.iloc[:, :-1]
label = dataset['Outcome']
```

d. Normalisasi

Normalisasi dilakukan agar setiap fitur memiliki skala yang seragam. Pada pembuataan model ini, dataset dinormalisasi menggunakan MinMaxScaler, yang berfungsi mentransformasi data ke dalam rentang nilai 0 hingga 1.

```
# Normalisasi min max
scaler = MinMaxScaler()
result = scaler.fit_transform(data)
data = pd.DataFrame(data=result,
columns=data.columns)
```

5. Proses klasifikasi

a. Mendefinisikan fungsi GLVQ dari sklearn

Setelah dilakukan data cleaning, maka dilanjutkan memanggil fungsi GLVQ dan menentukan value dari parameter - parameternya.

```
model = GLVQ(
```

```
    distance_type="squared-euclidean",
    activation_type="swish",
    activation_params={"beta": 2},
    solver_type="steepest-gradient-descent",
    solver_params={"max_runs": 200, "step_size":
0.01},
)
```

Berikut adalah penjelasan setiap parameter yang digunakan dalam model GLVQ tersebut:

1. distance_type="squared-euclidean"

   Parameter ini menentukan jenis metrik jarak yang digunakan dalam model GLVQ. "squared-euclidean" berarti menggunakan jarak Euclidean kuadrat. Jarak Euclidean mengukur jarak antara dua titik dalam ruang Euclidean. Menggunakan jarak kuadrat dapat mempercepat perhitungan dan memiliki beberapa sifat matematika yang berguna dalam pembelajaran.

2. activation_type="swish"

   Parameter ini menentukan fungsi aktivasi yang digunakan dalam model. Fungsi ini bisa memperbaiki kinerja model dengan memungkinkan gradien yang lebih halus selama proses pembelajaran.

3. activation_params={"beta": 2}:

   Parameter ini adalah argumen tambahan yang diberikan kepada fungsi aktivasi. Dalam hal ini, parameter "beta" diatur ke 2 untuk fungsi aktivasi "swish". Nilai "beta" mengontrol seberapa cepat fungsi aktivasi bertransisi antara input negatif dan positif.

4. solver_type="steepest-gradient-descent"

   Parameter ini menentukan jenis algoritma optimasi yang digunakan untuk melatih model.

"steepest-gradient-descent" adalah metode optimasi yang menggunakan gradien tertinggi untuk menemukan titik minimum dari fungsi kehilangan (loss function). Metode ini iteratif dan secara bertahap menyesuaikan parameter model untuk meminimalkan kesalahan.

5. solver_params={"max_runs": 100, "step_size": 0.01}

   Parameter ini adalah argumen tambahan yang diberikan kepada algoritma optimasi.

6. max_runs: 100

   Menentukan jumlah maksimum iterasi (atau langkah) yang akan dijalankan oleh algoritma optimasi selama pelatihan. Dalam hal ini, algoritma akan berhenti setelah 100 iterasi.

7. step_size: 0.01

   Menentukan ukuran langkah (learning rate) yang digunakan dalam setiap iterasi optimasi. Ukuran langkah mengontrol seberapa besar perubahan yang dilakukan pada parameter model dalam setiap iterasi. Nilai yang lebih kecil berarti perubahan yang lebih kecil dan lebih hati-hati, sedangkan nilai yang lebih besar berarti perubahan yang lebih besar dan lebih cepat.

b. Proses training dan testing model

```
X_train, X_test, y_train, y_test =
train_test_split(data, label, test_size=0.3,
random_state=42)

model.fit(X_train, y_train)

predicted_labels = model.predict(X_test)

print(classification_report(y_test,
predicted_labels))
```

	precision	recall	f1-score	support
0	0.80	0.80	0.80	151
1	0.62	0.61	0.62	80
accuracy			0.74	231
macro avg	0.71	0.71	0.71	231
weighted avg	0.74	0.74	0.74	231

Berikut adalah penjelasan dari masing - masing metrik yang dituliskan di atas:

Precision: Merupakan rasio prediksi yang benar positif (positif yang diprediksi benar) terhadap keseluruhan prediksi positif (positif yang diprediksi benar dan yang sebenarnya positif). *Precision* dihitung dengan rumus TP / (TP + FP), di mana TP adalah *True Positive* (positif yang diprediksi benar) dan FP adalah False Positive (negatif yang diprediksi salah).

Recall: Merupakan rasio prediksi yang benar positif terhadap keseluruhan data yang sebenarnya positif. *Recall* dihitung dengan rumus TP / (TP + FN), di mana FN adalah *False Negative* (positif yang diprediksi salah).

F1-score: Merupakan rata-rata harmonis dari precision dan recall, yang memberikan keseimbangan antara kedua metrik tersebut. *F1-score* dihitung dengan rumus 2 * (*precision* * *recall*) / (*precision* + *recall*).

Support: Merupakan jumlah data yang mendukung setiap kelas.

Accuracy: Merupakan rasio prediksi yang benar (positif dan negatif) terhadap keseluruhan data. *Accuracy* dihitung dengan rumus (TP + TN) / (TP + TN + FP + FN), di mana TN adalah *True Negative* (negatif yang diprediksi benar).

Dari hasil evaluasi tersebut, model klasifikasi memiliki akurasi sebesar 0.74, yang berarti sekitar 74% dari data telah diprediksi dengan benar. *Precision* dan *recall* untuk masing-masing kelas (0 dan 1) juga cukup tinggi, menunjukkan bahwa model cenderung dapat memprediksi kedua kelas dengan baik.

c. Melihat median dari masing - masing data yang dikategorikan sebagai 1 dan 0.

Setelah dilakukan evaluasi model, maka diperlukan validasi ulang mengenai data dengan ciri - ciri apa yang bisa diklasifikasikan sebagai 1 (Terindikasi diabetes) dan 0 (Tidak terindikasi diabetes). Oleh karena itu, dilakukan proses membuat variabel baru untuk menyimpan DataFrame data yang diklasifikasikan sebagai 1 atau 0 dan prediksi nya sesuai dengan kelas sebenarnya.

```
# Mengonversi X_test menjadi DataFrame
df_X_test = pd.DataFrame(X_test)
df_X_test["predicted_outcome"] = predicted_labels

# Mengonversi y_test menjadi DataFrame dengan nama kolom yang sesuai
df_y_test = pd.DataFrame(y_test, columns=["Outcome"])

# Menggabungkan X_test dengan y_test berdasarkan indeks
df_X_test = df_X_test.join(df_y_test)

# Memfilter dataframe untuk mendapatkan data yang diprediksi dengan benar
df_correct_predictions = df_X_test[df_X_test["Outcome"] == df_X_test["predicted_outcome"]]

# Memfilter dataframe untuk mendapatkan data yang dikategorikan sebagai 1 oleh model
df_category_1 = df_correct_predictions[df_correct_predictions["predicted_outcome"] == 1]

# Memfilter dataframe untuk mendapatkan data yang dikategorikan sebagai 0 oleh model
df_category_0 = df_correct_predictions[df_correct_predictions["predicted_outcome"] == 0]
```

Setelah itu, dilakukan *inverse transform* untuk mengembalikan nilai setelah normalisasi ke nilai sebelum dinormalisasi. Selanjutnya, hasil dari inverse tersebut diubah ke bentuk DataFrame dengan bantuan pandas, dan mencari median dari setiap masing - masing data pada kelas 1 maupun 0.

1. Median data yang diklasifikasikan sebagai 1 (Terindikasi diabetes)

```
Pregnancies                 6.000
Glucose                   158.000
BloodPressure              76.000
SkinThickness              31.000
Insulin                     0.000
BMI                        35.900
DiabetesPedigreeFunction    0.528
Age                        38.000
```

2. Median data yang diklasifikasikan sebagai 0 (Tidak Terindikasi diabetes)

Pregnancies	2.0000
Glucose	100.5000
BloodPressure	70.0000
SkinThickness	20.0000
Insulin	43.0000
BMI	30.1000
DiabetesPedigreeFunction	0.3235
Age	25.5000

3.7. Hasil dan Analisis Data

Dari implementasi yang dilakukan, hasil yang didapat adalah sebuah model yang menerapkan algoritma klasifikasi LVQ (GLVQ) dengan akurasi sebesar 74%. Akurasi 74% seharusnya sudah cukup baik mengingat jumlah data pada dataset yang relatif kecil.

Namun, meskipun akurasi yang didapat relatif kecil, kami mencoba untuk melakukan pengujian menggunakan data sintetis yang dibuat secara manual dan beracuan pada median data di setiap kelas yang ada (0 dan 1). Berikut adalah implementasinya:

```
# Data sintesis

# Pregnancies
# Glucose
# BloodPressure
# SkinThickness
# Insulin
# BMI
# DiabetesPedigreeFunction
# Age

data_sintetis = np.array([
    [6, 98, 58, 33, 190, 34, 0.43, 43],
    [2, 90, 65, 20, 80, 25.0, 0.2, 40],
    [8, 180, 90, 35, 120, 32.0, 0.5, 55],
    [3, 95, 75, 30, 90, 27.0, 0.4, 30],
    [6, 140, 85, 28, 110, 30.0, 0.6, 45]
])
```

```
# Melakukan scaling pada data uji
data_sintetis_scaled = scaler.fit_transform(data_sintetis)

# Melakukan prediksi pada data uji yang telah di-scale
predicted_classes = model.predict(data_sintetis_scaled)

# Mencetak hasil prediksi
for i, prediction in enumerate(predicted_classes):
    print(f"Prediksi kelas untuk data ke-{i+1}: {prediction}")
```

Di awal didefinisikan terlebih dahulu data - data apa saya yang dimasukkan, lalu diubah ke bentuk array menggunakan numpy. Setelah itu data tersebut dinormalisasi dengan MinMaxScaler untuk menyesuaikan dengan pelatihan model. Selanjutnya dilakukan prediksi untuk setiap data sintetis atau data buatan dan menghasilkan sebagai berikut.

```
Prediksi kelas untuk data ke-1: 0
Prediksi kelas untuk data ke-2: 0
Prediksi kelas untuk data ke-3: 1
Prediksi kelas untuk data ke-4: 0
Prediksi kelas untuk data ke-5: 1
```

Jika dilakukan pencocokan secara manual dengan median dari data - data yang dikategorikan sebagai 1 atau 0 pada pembahasan sebelumnya, maka model sudah dapat memprediksi dengan benar untuk data - data sintetis yang diujikan.

3.8. Jadwal Penelitian

No	Tahapan Penelitian	Mei				Juni
		1	2	3	4	1
1	Studi Literatur					
2	System Requirement					
3	Implementasi					
4	Evaluasi					
5	Pembuatan Laporan					
6	Revisi					

BAB 4 KESIMPULAN DAN SARAN

4.1. Kesimpulan

LVQ merupakan metode klasifikasi metode klasifikasi berstruktur, kompetitif, dan berlapis yang secara otomatis belajar untuk mengelompokkan data berdasarkan jarak antara vektor masukan. Dua vektor masukan yang memiliki kemiripan akan ditempatkan dalam kelompok yang sama.

Metode *Learning Vector Quantization* (LVQ) dapat diaplikasikan untuk memprediksi indikasi status diabetes dengan langkah-langkah seperti *preprocessing* data, pembagian data, pelatihan model LVQ, dan klasifikasi data baru. Kinerja model LVQ dievaluasi menggunakan metrik seperti akurasi, presisi, recall, dan F1-score. Penelitian lebih lanjut dapat dilakukan untuk meningkatkan akurasi model dan memberikan manfaat seperti prediksi yang lebih cepat dan tepat, serta peningkatan kualitas hidup pasien diabetes.

Berdasar pada hasil yang didapatkan, model yang telah dibuat dapat memprediksi dengan cukup baik, meskipun akurasi model hanya berada di angka 74%. Selain itu, percobaan pengujian dengan data sintetis oleh penulis juga mendukung bahwa model dapat memprediksi apakah seorang pasien menderita diabetes atau tidak.

4.2. Saran

Penelitian prediksi indikasi diabetes menggunakan *Metode Learning Vector Quantization* (LVQ) menunjukkan hasil yang cukup bagus. Namun, diperlukan pengembangan lebih lanjut untuk meningkatkan akurasi dan generalisasi model. Saran untuk penelitian selanjutnya meliputi perluasan dataset, optimasi parameter model, interpretasi dan visualisasi hasil, penerapan dan validasi pada data real, dan investigasi faktor risiko lain. Dengan pengembangan dan penelitian lebih lanjut, metode LVQ dapat menjadi alat yang canggih untuk memprediksi diabetes dan membantu dalam pencegahan dan pengelolaan penyakit ini.

DAFTAR PUSTAKA

A. S. Nugroho, T. D. P. Prayitno, and A. T. Raharjo. (2019). "Klasifikasi Citra Matahari dengan Menggunakan *Learning Vector Quantization* (LVQ)". Jurnal Gaussian, 8(4), 439-446.

Fauzan Ishaq, Dr.Ir.Bambang Hidayat, IPM, drg. Yurika Ambar Lita. (2018). "Minangkabau and Sunda Tribes Detection Based on Lip Print Pattern Using Discrete Cosine Transform (DCT) and *Learning Vector Quantization* (LVQ)".

Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.

M. Arifin, S. Harjoko, and B. Argyanto. (2018). "Klasifikasi Citra Menggunakan *Learning Vector Quantization* (LVQ) Berbasis Jaringan Syaraf Tiruan". Jurnal Rekayasa Elektrika, 13(2), 75-82.

M. R. S. Lubis, E. D. Suhendra, and S. Syamsuddin. (2020). "Pengenalan Pola Wajah dengan Menggunakan Metode *Learning Vector Quantization* (LVQ) untuk Sistem Keamanan". Jurnal Ilmiah Teknologi Informasi Asia, 14(2), 139-147.

Ogurtsova, K., da Rocha Fernandes, J. D., Huang, Y., Linnenkamp, U., Guariguata, L., Cho, N. H., Cavan, D., Shaw, J. E., & Makaroff, L. E. (2017). IDF Diabetes Atlas: Global estimates for the prevalence of diabetes for 2015 and 2040. Diabetes Research and Clinical Practice, 128, 40-50. https://doi.org/10.1016/j.diabres.2017.03.024

Sudarmaji. (2015). Analisis Kualitas Air Menggunakan Parameter Fisika, Kimia dan Mikrobiologi di Desa Cimuncang Kecamatan Tegalwaru Kabupaten Purwakarta. Jurnal Kesehatan Lingkungan Indonesia, 14(2), 73-78.

Simanjuntak, T. P. (2017). Analisis Kualitas Air Berdasarkan Parameter Fisika, Kimia dan Mikrobiologi di Kota Medan. Jurnal Kesehatan Masyarakat Andalas, 11(1), 46-52.

Sitorus, R., et al. (2017). Analisis Air Minum di Kota Pekanbaru Berdasarkan Parameter Fisik, Kimia, dan Mikrobiologi. Jurnal Kesehatan Lingkungan Indonesia, 16(2), 71-76.

Tan, P. N., Steinbach, M., & Kumar, V. (2006). Introduction to Data Mining. Pearson Education.

Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining: Practical Machine Learning Tools and Techniques (3rd ed.). Morgan Kaufmann.