Analisis Perbandingan Metrik Jarak pada Algoritma K-Means Clustering untuk Data Penyakit Jantung

**Abstrak**

Clustering merupakan salah satu teknik fundamental dalam machine learning untuk mengelompokkan data tanpa label. K-Means adalah algoritma clustering yang paling populer dan banyak digunakan karena kesederhanaan dan efisiensinya. Kinerja K-Means sangat bergantung pada metrik jarak yang digunakan untuk mengukur kedekatan antar data. Laporan ini menganalisis dan membandingkan beberapa metrik jarak, baik standar maupun kustom, dalam kerangka algoritma K-Means menggunakan dataset penyakit jantung. Metrik yang diuji meliputi Euclidean, Manhattan (CityBlock), Chebyshev, Minkowski, Canberra, Bray-Curtis, serta divergensi Kullback-Leibler dan Jensen-Shannon. Untuk metrik yang tidak optimal dengan pembaruan centroid berbasis mean, diimplementasikan juga pendekatan alternatif seperti K-Medians dan optimasi menggunakan Genetic Algorithm (GA). Hasil clustering dievaluasi secara kualitatif melalui visualisasi Principal Component Analysis (PCA) dan secara kuantitatif menggunakan Silhouette Score untuk menunjukkan bagaimana pilihan metrik dan metode optimasi centroid memengaruhi kualitas pengelompokan data.

# Pendahuluan

Analisis cluster adalah proses mengelompokkan sekumpulan objek data ke dalam beberapa grup (cluster) sedemikian rupa sehingga objek dalam satu cluster memiliki kemiripan yang tinggi satu sama lain, namun sangat berbeda dengan objek di cluster lain. K-Means adalah salah satu algoritma partisi clustering yang paling dikenal, yang bertujuan untuk mempartisi n pengamatan ke dalam k cluster di mana setiap pengamatan termasuk dalam cluster dengan mean terdekat (centroid cluster) (MacQueen, 1967).

Secara inheren, algoritma K-Means standar dirancang untuk meminimalkan *sum of squared Euclidean distances* antara titik data dan centroid clusternya. Hal ini menjadikan jarak Euclidean sebagai pilihan metrik default. Namun, dalam berbagai kasus, metrik jarak lain mungkin lebih sesuai tergantung pada karakteristik data.

Laporan ini akan mengeksplorasi implementasi algoritma K-Means dengan berbagai metrik jarak pada dataset heart\_data.csv. Langkah-langkah yang dilakukan meliputi pra-pemrosesan data, implementasi K-Means dengan metrik jarak yang berbeda, metode optimasi centroid alternatif (K-Medians dan Genetic Algorithm), serta visualisasi hasil menggunakan PCA.

# Dasar Teori

## Algoritma K-Means

Algoritma K-Means bekerja melalui langkah-langkah iteratif berikut:

1. **Inisialisasi**: Pilih k titik centroid secara acak dari data.
2. **Langkah Penugasan (Assignment Step)**: Tetapkan setiap titik data ke centroid terdekat berdasarkan metrik jarak yang dipilih.
3. **Langkah Pembaruan (Update Step)**: Hitung ulang posisi centroid sebagai titik mean dari semua data yang termasuk dalam clusternya.
4. **Iterasi**: Ulangi langkah 2 dan 3 hingga posisi centroid tidak lagi berubah secara signifikan atau jumlah iterasi maksimum tercapai.

## Metrik Jarak dan Divergensi

Berikut adalah metrik yang dianalisis dalam laporan ini:

* **Euclidean Distance**: Jarak garis lurus antara dua titik. Untuk titik p dan q dalam ruang n-dimensi, jaraknya adalah:
* **Manhattan Distance (Cityblock)**: Jumlah dari selisih absolut koordinat-koordinatnya. Formulanya adalah:

Metrik ini lebih optimal jika centroid diperbarui menggunakan median (K-Medians).

* **Chebyshev Distance**: Jarak maksimum di sepanjang sumbu koordinat.
* **Minkowski Distance**: Generalisasi dari Euclidean dan Manhattan.

Ketika p=1 menjadi Manhattan, dan p=2 menjadi Euclidean.

* **Canberra Distance**: Varian dari Manhattan yang sensitif terhadap nilai mendekati nol.
* **Bray-Curtis Dissimilarity**: Ukuran yang sering digunakan dalam ekologi.
* **Kullback-Leibler & Jensen-Shannon Divergence**: Ukuran untuk mengukur perbedaan antara dua distribusi probabilitas, bukan jarak geometris. Data harus dinormalisasi menjadi distribusi probabilitas (misalnya, menggunakan Softmax) sebelum digunakan.

## Principal Component Analysis (PCA)

PCA adalah teknik reduksi dimensi yang mengubah data ke sistem koordinat baru sehingga varians terbesar dari data terletak pada komponen utama pertama, kedua terbesar pada komponen kedua, dan seterusnya (Pearson, 1901). Dalam laporan ini, PCA digunakan untuk mereduksi data 4 dimensi menjadi 2 dimensi agar hasil clustering dapat divisualisasikan.

# Metodologi

Langkah-langkah yang dilakukan dalam analisis ini sesuai dengan alur pada notebook yang disediakan.

## Data Loading dan Eksplorasi Awal

Data heart\_data.csv yang berisi 4 fitur numerik dimuat ke dalam DataFrame pandas.

import pandas as pd

df = pd.read\_csv('heart\_data.csv')

df.head()

Analisis data eksplorasi (EDA) dilakukan untuk memahami distribusi dan hubungan antar variabel.

Python

# Informasi dasar dataset

df.info()

# Visualisasi distribusi

import seaborn as sns

sns.pairplot(df)

sns.boxplot(data=df)

## Pra-pemrosesan Data

#### Penanganan Outlier

Ditemukan satu outlier pada kolom cholesterol menggunakan metode IQR (Interquartile Range) dan data tersebut dihapus.

Q1 = df['cholesterol'].quantile(0.25)

Q3 = df['cholesterol'].quantile(0.75)

IQR = Q3 - Q1

df = df[(df['cholesterol'] >= (Q1 - 1.5 \* IQR)) & (df['cholesterol'] <= (Q3 + 1.5 \* IQR))]

#### Pengecekan Duplikat

Tidak ditemukan data duplikat.

#### Standardisasi

Data distandarisasi menggunakan StandardScaler dari Scikit-learn untuk memastikan setiap fitur memiliki skala yang sama, yang krusial untuk algoritma berbasis jarak.

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

df\_scaled = scaler.fit\_transform(df)

## Implementasi Clustering dan Visualisasi

Untuk setiap metrik jarak, algoritma K-Means (atau variannya) diimplementasikan. Hasilnya kemudian direduksi dimensinya menggunakan PCA dan divisualisasikan dalam bentuk scatter plot.

# Hasil dan Pembahasan

## K-Means dengan Euclidean Distance (Standar)

Implementasi menggunakan KMeans dari Scikit-learn yang secara default menggunakan jarak Euclidean.

from sklearn.cluster import KMeans

from sklearn.decomposition import PCA

import matplotlib.pyplot as plt

# Algoritma K-means, k=2

kmeans = KMeans(n\_clusters=2, random\_state=42, n\_init='auto')

labels\_euclidean = kmeans.fit\_predict(df\_scaled)

# Reduksi dimensi dengan PCA

pca = PCA(n\_components=2)

df\_pca = pca.fit\_transform(df\_scaled)

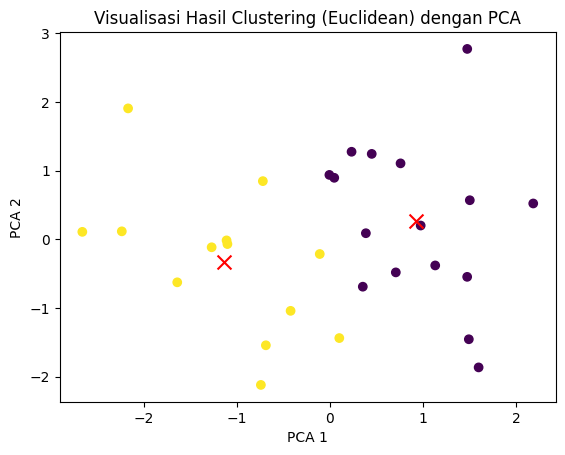
# Visualisasi

plt.scatter(df\_pca[:, 0], df\_pca[:, 1], c=labels\_euclidean)

plt.title('Hasil Clustering (Euclidean) dengan PCA')

plt.show()

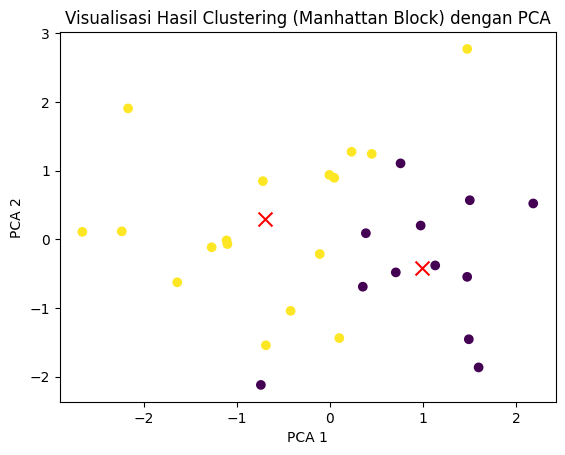
Hasil visualisasi menunjukkan dua cluster yang cukup terpisah, yang menjadi dasar perbandingan dengan metrik lainnya.



Gambar 4.1 Hasil clustering dengan PCA menggunakan metrik Euclidean

## K-Means dengan Manhattan dan Chebyshev Distance

Karena pembaruan centroid berbasis *mean* tidak optimal untuk metrik ini, dua pendekatan diuji: (1) K-Means kustom dengan *mean update*, dan (2) K-Medians kustom dengan *median update*.

**** **A chart with yellow and purple dots

AI-generated content may be incorrect.**

Gambar 4.2 *Hasil clustering menggunakan K-Means (kiri) dan K-Medians (kanan) dengan jarak Manhattan*

A diagram of a number of dots

AI-generated content may be incorrect. A chart with red and yellow dots

AI-generated content may be incorrect.

Gambar 4.3 *Hasil clustering menggunakan K-Means (kiri) dan K-Medians (kanan) dengan jarak Chebyshev*

Dari hasil evaluasi pada notebook untuk jarak Chebyshev, **Silhouette Score** untuk K-Medians (0.2478) jauh lebih tinggi dibandingkan K-Means (0.1177). Ini secara kuantitatif membuktikan bahwa pembaruan centroid menggunakan median lebih cocok untuk metrik ini.

## K-Means dengan Minkowski Distance

Karena Minkowski dengan ekuivalen dengan Euclidean, hasilnya identik dengan implementasi standar Scikit-learn.

## Optimasi K-Means dengan Canberra dan Bray-Curtis menggunakan Genetic Algorithm (GA)

Metrik ini sulit dioptimalkan dengan metode heuristik sederhana. Oleh karena itu, pendekatan metaheuristik menggunakan Genetic Algorithm (GA) diterapkan untuk mencari posisi centroid yang meminimalkan total jarak Canberra di dalam cluster.

# Contoh fungsi objektif untuk GA

def objective(individual):

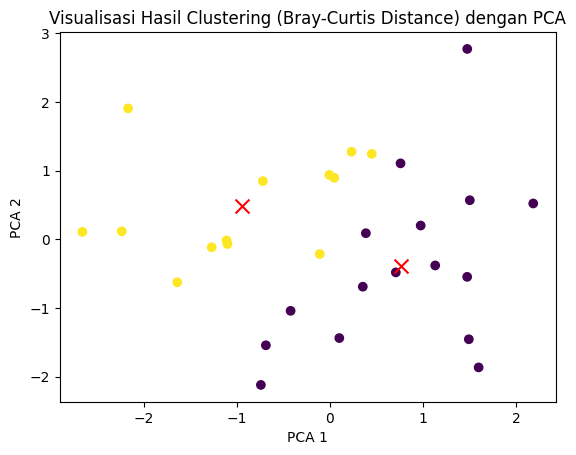
    return (np.sum([canberra\_distance(individual, x) for x in cluster\_points]),)

A diagram of a number of dots

AI-generated content may be incorrect.A chart with yellow and red dots

AI-generated content may be incorrect.

Gambar 4.4 *Hasil clustering menggunakan K-Means (Canberra) (kiri) dan dengan optimasi GA(kanan).*

A graph with yellow and purple dots

AI-generated content may be incorrect.

Gambar 4.5 *Hasil clustering menggunakan K-Means (Bray-Curtis) (kiri) dan dengan optimasi GA(kanan).*

Hasil evaluasi menunjukkan peningkatan signifikan:

* **Canberra**: Silhouette Score meningkat dari 0.0764 (K-Means) menjadi **0.2513** (K-Means + GA).
* **Bray-Curtis**: Silhouette Score meningkat dari 0.1898 (K-Means) menjadi **0.2090** (K-Means + GA).

Ini membuktikan bahwa untuk metrik yang kompleks, metode optimasi seperti GA dapat menghasilkan pengelompokan yang jauh lebih baik.

## K-Means dengan Divergensi Kullback-Leibler (DKL) dan Jensen-Shannon Distance (JSD)

Metrik ini memerlukan input berupa distribusi probabilitas. Data yang telah distandarisasi diubah menggunakan fungsi *softmax* pada setiap barisnya.

from scipy.special import softmax

# Konversi tiap row ke distribusi dengan softmax

distributions = np.apply\_along\_axis(softmax, 1, df\_scaled)

Hasil clustering dengan DKL dan JSD menunjukkan pola pengelompokan yang berbeda dari metrik geometris, karena keduanya mengukur perbedaan informasi antar distribusi, bukan jarak spasial.

**5. Kesimpulan**

Berdasarkan analisis yang telah dilakukan, dapat ditarik beberapa kesimpulan:

1. Pilihan metrik jarak secara fundamental memengaruhi hasil dari algoritma K-Means.
2. Algoritma K-Means standar yang menggunakan *mean* sebagai centroid secara matematis optimal untuk **Euclidean Distance**.
3. Untuk metrik jarak yang berbasis L1-norm seperti **Manhattan** dan **Chebyshev**, penggunaan *median* (K-Medians) sebagai pusat cluster memberikan hasil yang lebih baik dan lebih konsisten secara teoretis, yang didukung oleh peningkatan Silhouette Score.
4. Metrik yang kompleks dan bergantung pada skala data seperti **Canberra** dan **Bray-Curtis** tidak dapat dioptimalkan secara efektif menggunakan pembaruan centroid berbasis mean. Metode optimasi metaheuristik seperti **Genetic Algorithm** terbukti mampu meningkatkan kualitas clustering secara signifikan untuk metrik ini.
5. Metrik berbasis divergensi seperti **Kullback-Leibler** dan **Jensen-Shannon** menawarkan perspektif yang berbeda dalam mengukur "jarak" dan menghasilkan pengelompokan yang berfokus pada perbedaan distribusi informasi antar titik data.

Secara keseluruhan, tidak ada satu metrik jarak yang superior untuk semua kasus. Pemilihan metrik harus didasarkan pada pemahaman terhadap karakteristik data dan tujuan dari analisis clustering itu sendiri.

# Daftar Pustaka

Fortin, F. A., De Rainville, F. M., Gardner, M. A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research, 13*, 2171-2175.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, No. 14, pp. 281-297).

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2*(11), 559-572.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics, 20*, 53-65.