EVALUASI CLUSTERING K-MEANS DAN K-MEDOID PADA PERSEBARAN COVID-19 DI INDONESIA DENGAN METODE DAVIES-BOULDIN INDEX (DBI)

Fathurrahman [1*], Sri Harini [2], Ririen Kusumawati [3]
[1,2,3] Magister Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Indonesia
fathurrahman637@gmail.com

ABSTRAK

Tingginya persebaran Covid-19 di Indonesia, dan persebaran di tiap-tiap daerah yang berbeda-beda, menjadikan perlu adanya pengelompokan daerah dengan masing-masing tingkat penyebarannya, untuk mengetahui kemiripan karakteristik atau kriteria dari setiap daerah dengan tingkat penyebaran Covid-19 yang akan terkumpul dalam suatu cluster tertentu. Penelitian ini menggunakan komparasi analisis cluster menggunakan K-Means dan K-Medoid untuk menganalisis perseberan virus Covid-19 di indonesia. Analisis komparasi kedua algoritma dibuktikan dengan adanya nilai davies bouldin index (DBI) sebagai parameter evaluasi menggunakan Bahasa Pemrograman Python Version 3 yang dijalankan pada tools Jupyter Notebook. Langkah penelitian dimulai dari import library atau modul yang digunakan dalam berbagai tahapan dalam penelitian ini. Tahapan yang dilakukan antara lain melakukan pre-processing berupa proses binning data hingga normalisasi data. Selanjutnya, menampilkan visualisasi data sebaran Covid-19. Kemudian, melakukan modeling Algoritma K-Means dan K-Medoids. Hingga diakhiri dengan langkah terakhir berupa evaluasi menggunakan Davies-Bouldin Index (DBI). Setelah dilakukan evaluasi DBI, K-Means mendapatkan nilai 0.9762331449809145, sedangkan K-Medoids mendapatkan nilai 0.9809235412405508. Karena K-Means memiliki nilai DBI yang lebih rendah dibandingkan K-medoids, maka dapat dikatakan K-Means menghasilkan klasterisasi yang lebih baik dalam klasterisasi data sebaran Covid-19 di Indonesia.

Keyword *: Covid-19, Davies-Bouldin Index, K-Means Clustering, K-Medoids Clustering*

1. PENDAHULUAN

Tingginya persebaran Covid-19 di Indonesia, dan persebaran di tiap-tiap daerah yang berbeda-beda, menjadikan perlu adanya pengelompokan daerah dengan masing-masing tingkat penyebarannya, untuk mengetahui kemiripan karakteristik atau kriteria dari setiap daerah dengan tingkat penyebaran Covid-19 yang akan terkumpul dalam suatu cluster tertentu. Dua diantara metode *non-hierarchical clustering* yang populer dan banyak diterapkan adalah metode K-Means dan K-Medoid [1]. Mengacu pada penelitiannya, Rodriguez menyatakan bahwa K-Means dan K-Medoid populer digunakan karena keduanya memiliki kemiripan konsep. Diantaranya adalah kemiripan dalam alur pembacaan data, penentuan k kelompok serta kesamaan metode jarak yang digunakan, yaitu (*euclidean distance*). Selain itu, K-Means dan K-Medoid memiliki kelebihan masing-masing. Mengacu pada penelitian Deswiaqsa, dkk [2], K-Means dan K-Medoid adalah dua algoritma pengelompokan yang berbeda namun memiliki kesamaan konseptual. K-Means melakukan partisi data menjadi beberapa k-cluster yang telah ditentukan sebelumnya, sementara K-Medoid juga melakukan partisi data menjadi beberapa kelompok dengan karakteristik yang serupa. Perbedaan utama antara kedua algoritma ini terletak pada representasi pusat cluster [3]. K-Medoid menggunakan objek sebagai medoid yang mewakili pusat cluster untuk setiap kelompok, sementara K-Means menggunakan nilai rata-rata sebagai pusat cluster. Oleh karena itu, membandingkan kinerja K-Means dan K-Medoid dalam sebuah penelitian menjadi tantangan menarik berdasarkan tinjauan literatur mengenai kesamaan dan perbedaan konsep di antara keduanya [4]. Terdapat beragam implementasi aplikatif dari K-Means dan K-Medoid dalam analisis *cluster*, salah satunya adalah untuk mengelompokan persebaran penyakit Covid-19 sebagaimana penelitian Virgantari, dkk [5] dan Sindi, dkk [6]. Penyakit covid-19 adalah penyakit menular yang disebabkan oleh jenis baru virus corona. Sebagian orang yang terinfeksi akan mengalami penyakit pernapasan ringan hingga berat. Penyakit ini telah menjadi ancaman bagi kesehatan masyarakat di seluruh dunia. Pasalnya, penyakit ini telah menyebar dengan cepat hingga ke 199 negara di seluruh dunia sepanjang tahun 2020, sehingga menjadikan status pandemi secara global. Menurut laporan Our World in Data, pada tanggal 17 Maret 2022, Indonesia memiliki tingkat kematian (case fatality rate/CFR) Covid-19 sebesar 2,58%. Data ini menunjukkan bahwa Indonesia berada di peringkat kedua tertinggi di Asia Tenggara dalam hal persentase kematian akibat Covid-19 [7]. Dalam penelitian ini, dilakukan perbandingan analisis cluster menggunakan algoritma K-Means dan K-Medoid untuk menganalisis penyebaran virus Covid-19 di Indonesia. Evaluasi perbandingan kedua algoritma dilakukan dengan menggunakan Davies Bouldin Index (DBI) sebagai parameter evaluasi. Dalam analisis cluster, terdapat kelemahan ketika titik cluster dipilih secara acak, yang dapat menghasilkan variasi data yang berbeda. Jika nilai DBI rendah, maka pengelompokan yang dihasilkan dianggap

lebih optimal. Oleh karena itu, dalam penelitian ini, peneliti mengevaluasi kedua algoritma menggunakan DBI, dimana semakin mendekati angka 0, semakin optimal pula cluster yang terbentuk.

2. TINJAUAN PUSTAKA

Jika Analisis *cluster* yang digunakan dalam penelitian ini adalah metode K-Means dan K-Medoid. Sub bab ini menyajikan penelitian terkait sebagai *state of the art* penelitian. Pembahasan *state of the art* difokuskan pada algoritma K-Means dan K-Medoid yang secara umum yang telah dilakukan untuk menyelesaikan permasalahan *clustering*. Selain itu, penelitian terkait pada subbab ini juga membahas mengenai klasterisasi K-Means dan K-Medoid yang diimplementasikan secara khusus untuk Covid-19.

Pertama, peneliti mengkaji penelitian komparatif antar algoritma klastering yang dilakukan oleh Rodriguez [1]. Dua diantara algoritma klasterisasi yang dibandingkan adalah K-Means dan K-Medoid. Mengacu pada penelitiannya, Rodriguez menyatakan bahwa K-Means dan K-Medoid populer digunakan karena keduanya memiliki kemiripan konsep. Diantaranya adalah kemiripan dalam alur pembacaan data, penentuan k kelompok serta kesamaan metode jarak yang digunakan, yaitu (*euclidean distance*). Sementara itu, terdapat perbedaan mendasar dari kedua algoritma ini. K-Medoid menggunakan objek sebagai perwakilan (*medoid*) sebagai pusat *cluster* untuk setiap *cluster*, sedangkan K-Means menggunakan nilai rata-rata (*mean*) sebaga pusat *cluster*. Oleh karena itu, penelitian untuk mengkomparasikan kinerja K-Means dan K-Medoid menjadi *research challenge* yang menarik dan menantang. Hal ini untuk menguji secara komparatif perihal performa K-Means dan K-Medoid untuk objek klasterisasi tertentu.

Berikutnya, peneliti mengkaji penelitian terdahulu yang secara komparatif telah membandingkan K-Means dengan K-Medoid untuk klasterisasi objek yang berbeda. Hasilnya, terbukti bahwa K-Medoid belum tentu lebih unggal daripada K-Means, pun juga sebaliknya. Penelitian komparatif tentang K-Means dan K-Medoid oleh Qomariyah [8] dan Suarna [9] menyatakan bahwa K-Means lebih baik daripada K-Medoid berdasarkan evaluasi DBI. Namun, penelitian Nurhayati [10] menyimpulkan bahwa K-Medoid lebih unggal dari K-Means berdasarkan nilai akurasi, waktu eksekusi (*execution time*) dan kompleksitas waktu (*time complexity*). Lebih rinci, Qomariyah [8] menggunakan perbandingan K-Means dan K-Medoid untuk klasterisasi mahasiswa. Kemudian berdasarkan evaluasi DBI, K-Means menghasilkan nilai 0,781, sedangkan K-Medoid mendapatkan nilai 0,929. Pada evaluasi DBI, nilai yang mendekati 0 adalah nilai yang terbaik. Penelitian berikutnya yang dilakukan oleh Suarna [9] juga membuktikan bahwa K-Means lebih unggal dibandingkan K-Medoid untuk klasterisasi *fish cooking menu*. Disimpulkan bahwa, K-Means menghasilkan nilai DBI sebesar -1,535, sedangkan K-Medoid menghasilkan nilai -1,777. Keunggulan K-Medoid atas K-Means dibuktikan oleh Nurhayati [10] melalui penelitian menggunakan teknologi *big data*. Keunggulan K-Medoid dinyatakan berdasarkan akurasi, waktu eksekusi (*execution time*) dan kompleksitas waktu (*time complexity*). K-Medoid memperoleh akurasi lebih baik dengan nilai 63,24%, sedangkan 52,11% untuk akurasi K-Means. Waktu eksekusi (*execution time*) dari K-Medoid juga menunjukkan performa yang lebih baik. Tercatat bahwa rata-rata kecepatan eksekusi untuk K-Medoid adalah 3,1 ms, sedangkan K-Means sebesar 3,45 ms. Mengacu ke kompleksitas waktu (*time complexity*), penelitian tersebut menggunakan pembuktian melalui Big O (n2). Disimpulkan bahwa rata-rata nilai yang dihasilkan oleh K-Medoid adalah 310,157 sedangkan K-Means sebesar 377,886. Ketiga penelitian tersebut membuktikan bahwa studi komparatif antara K-Means dan K-Medoid tetap relevan untuk dilakukan mengingat objek penelitian saat ini berbeda dengan penelitian terdahulu, yakti sebaran Covid-19 di Indonesia.

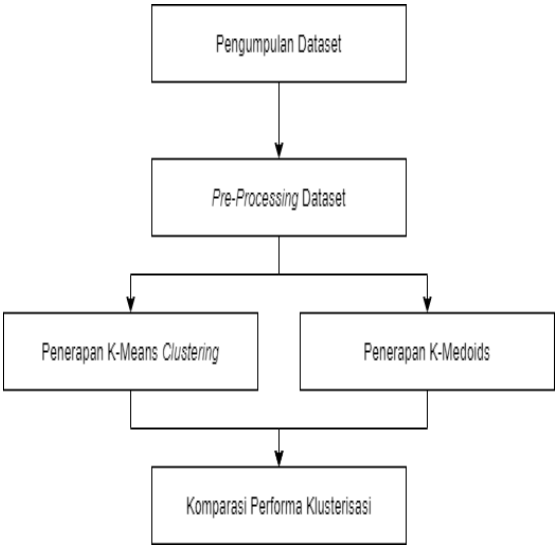
Penelitian klasterisasi Covid-19 pernah dilakukan oleh Abdullah [11], yaitu melakukan klasterisasi provinsi di Indonesia yang beresiko terpapar Covid-19 berdasarkan data yang terkonfirmasi positif Covid-19, kematian, serta pemulihan. Penelitian tersebut menggunakan algoritma K-Means yang membentuk 3 klaster provinsi. Selain itu,

Kemudian, penelitian yang dilakukan oleh Silvi [12] membandingkan metode centroid linkage dan K-Means dalam analisis cluster untuk mengelompokkan indikator HIV/AIDS di Indonesia dengan mempertimbangkan adanya data outlier. Penelitian ini menggunakan gap statistik untuk menentukan jumlah cluster yang ideal, dan hasilnya adalah terbagi menjadi 7 cluster. Hasil penelitian ini menyimpulkan bahwa untuk data yang memiliki outlier, metode centroid linkage memberikan hasil yang lebih sesuai dibandingkan dengan metode K-Means. Rasio Summer of Square Within/Summer of Square Between dari metode K-Means adalah 0.12232, sementara metode centroid linkage menghasilkan angka 0.067307. Metode centroid linkage menghasilkan kelompok yang lebih homogen, yang mengakibatkan nilai rasio yang lebih rendah. Hal ini menunjukkan bahwa metode centroid linkage memiliki tingkat ketepatan dalam pengelompokan yang lebih baik daripada K-Means..Kedua adalah pada penelitian yang dilakukan oleh Supriyadi [13] membahas perbandingan antara algoritma K-Means dan K-Medoid dalam pengelompokan armada kendaraan truk berdasarkan produktivitas. Dalam penelitian

ini, kedua metode clustering digunakan untuk mengelompokkan setiap armada kendaraan berdasarkan kinerja produktivitasnya. Penelitian tersebut juga melibatkan uji validasi terhadap hasil cluster yang terbentuk. Davies Bouldin Index (DBI) digunakan sebagai metode evaluasi dalam analisis cluster, dan menghasilkan nilai validitas sebesar 0,67 untuk K-Means dan 1,78 untuk K-Medoid.mada kendaraan truk berdasarkan produktivitas. Penelitian ketiga adalah sebagaimana yang dilakukan oleh Fira [14]. Penelitian tersebut mengelompokkan penyebaran Coivd-19 di Indonesia. Penelitian ini bertujuan untuk mengelompokan provinsi yang memiliki penyakit Covid-19 dengan tingkat tinggi dan rendah di Indonesia. Penelitian tersebut juga melakukan perbandingan dengan metode algoritma yang digunakan, yaitu K-Means dan K-Medoid. Hasil yang didapatkan pada penelitian tersebut adalah K-Means memiliki *cluster* 1 yang beranggotakan 2 wilayah dan dikategorikan tinggi, sedangkan *cluster* 2 sebanyak 32 wilayah dan dikategorikan rendah. Sedangkan, hasil menggunakan algoritma K-Medoid yaitu untuk *cluster* 1 beranggotakan 4 wilayah dan dikategorikan tinggi, sedangkan *cluster* 2 sebanyak 30 wilayah dan dikategorikan rendah. Komparasi perbandingan tersebut menghasilkan nilai *silhouette coefficient* dengan metode K-Means adalah sebesar 0,207 sedangkan nilai *silhouette coefficient* dengan metode K-Medoid adalah sebesar 0,347. Terakhir, adalah penelitian dari Adha [15] yang melakukan penelitian untuk mengelompokkan negara-negara yang memiliki pola kasus Covid-19 di dunia. Hasil pengelompokan dapat dijadikan acuan dan pola gambaran negara yang memiliki tingkat pemulihan rendah dapat mengamati proses pemulihan negara yang memiliki tingkat pemulihan tinggi yang berada dalam kelompoknya. Hasil penelitian ini menyatakan bahwa K-Means lebih unggul dari pada DBSCAN dalam mengelompokkan kasus Covid-19. Algoritma K-Means memiliki nilai SI terbaik sebesar 0.6902 yang terletak pada percobaan dengan nilai k=8.

3. METODE PENELITIAN

Pada bagian ini memuat metode yang digunakan pada penelitian yang dilakukan. Prosedur penelitian untuk membandingkan kinerja Algoritma K-Means *Clustering* dan K-Medoids ditunjukkan pada Gambar 1. Penelitian ini dimulai dari pengumpulan dataset. Kemudian, dilakukan *pre-processing* dataset untuk memastikan bahwa dataset siap digunakan sebagai masukan kedua algoritma. Setelahnya, dilakukan perhitungan Algoritma K-Means *Clustering* dan K-Medoids. Perbandingan kinerja kedua algoritma lantas diukur menggunakan metode evaluasi internal, yakni Davies Bouldin Index (DBI).



Gambar 1. Prosedur penelitian

3.1. Pengumpulan Dataset

Prosedur penelitian diawali dengan tahapan pengumpulan dataset. Dataset yang digunakan adalah dataset yang diunduh melalui laman https://www.kaggle.com/datasets/hendratno/covid19-indonesia. Dataset ini menunjukkan sebaran Covid-19 di Indonesia yang disajikan secara *time series*, terhitung mulai tanggal 1 Maret 2020 hingga 17 Maret 2021. Jumlah keseluruhan atribut dalam dataset adalah 38 dan jumlah *instances* adalah 21760.

3.2. Pre-Processing Dataset

Pre-processing dataset adalah proses yang mengubah data mentah ke dalam bentuk yang digunakan sebagai masukan algoritma. Proses ini penting dilakukan karena data mentah sering kali tidak memiliki format yang teratur. Selain itu, algoritma K-Means *Clustering* dan K-Medoids juga tidak dapat memproses data mentah, sehingga proses ini sangat penting dilakukan untuk mempermudah proses berikutnya, yakni analisis data menggunakan algoritma. Dalam penelitian ini, pre-processing dilakukan dengan cara mereduksi atribut yang tidak berkaitan langsung dengan perhitungan algoritma, serta mereduksi *instances* yang memiliki *missing value*. Untuk melakukan deteksi missing value, digunakan perkakas bantu data mining, yakni Waikato Environment for Knowledge Analysis yang dikenal dengan WEKA. Tabel 1 menunjukkan atribut-atribut yang dilakukan pre-processing berupa reduksi atribut dan reduksi instances tertentu yang memiliki missing value. Mengacu ke Tabel 1 atribut provinsi menjadi output dari klasterisasi. Hal ini dilakukan atas dasar bahwa Pemerintah Indonesia, melalui situs resminya, covid19.go.id menginformasikan sebaran Covid-19 berdasarkan provinsi di Indonesia. Selain itu, penelitian ini juga dikuatkan oleh penelitian

terdahulu yang melakukan klasterisasi berdasarkan provinsi, sebagaimana penelitian Abdullah [11].

Tabel 1. Atribut pada dataset

No.	Atribut	Tipe Atribut
1.	Date	Date
2.	Location ISO Code	Kategorikal
3.	Location	Kategorikal
4.	New Cases	Numerik
5.	New Deaths	Numerik
6.	New Recovered	Numerik
7.	New Active Cases	Numerik
8.	Total Cases	Numerik
9.	Total Deaths	Numerik
10.	Total Recovered	Numerik
11.	Total Active Cases	Numerik
12.	Location Level	Kategorikal
13.	City or Regency	Kategorikal
14.	Province	Kategorikal
15.	Country	Kategorikal
16.	Continent	Kategorikal
17.	Island	Kategorikal
18.	Time Zone	Time
19.	Special Status	Kategorikal
20.	Total Regencies	Numerik
21.	Total Cities	Numerik
22.	Total Districts	Numerik
23.	Total Urban Villages	Numerik
24.	Total Rural Villages	Numerik
25.	Area	Numerik
26.	Population	Numerik
27.	Population Density	Numerik
28.	Longitude	Numerik
29.	Latitude	Numerik
30.	New Cases per Million	Numerik
31.	Total Cases per Million	Numerik
32.	New Deaths per Million	Numerik
33.	Total Deaths per Million	Numerik
34.	Total Deaths per 100 rb	Numerik
35.	Case Fatality Rate	Numerik
36.	Case Recovered Rate	Numerik
37.	Growth Factor of New Cases	Numerik
38.	Growth Factor of New Deaths	Numerik

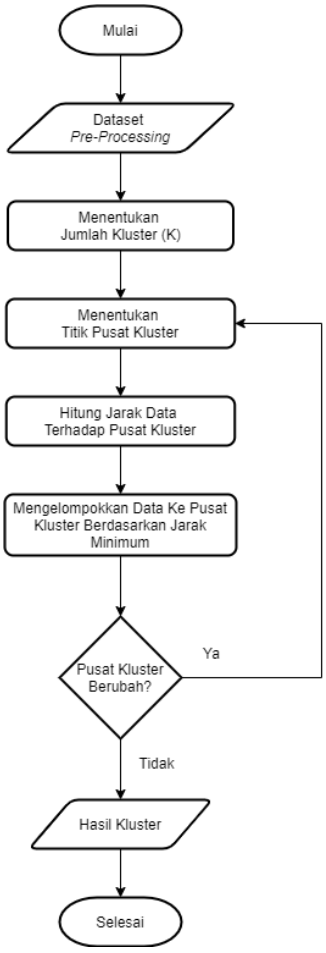
3.3. Perhitungan Algoritma K-Means

Gambar 2 menunjukkan diagram alir yang menggambarkan urutan langkah dari Algoritma K-Means. Merujuk penelitian Abdullah [11], langkah-langkah Algoritma K-Means dalam penelitian ini antara lain:

a. Mendapatkan nilai (k) sebagai jumlah kluster yang dibentuk. Nilai k optimal mengacu pada Metode Elbow.
b. Menentukan centroid sebagai titik pusat kluster awal secara acak pada dataset sebaran Covid-19.
c. Menghitung jarak antara pusat kluster dan seluruh *instances* pada dataset sebaran Covid-19 menggunakan rumus *Euclidience distance* sesuai Persamaan (1).

$$d(x,y) = \sqrt{(\sum_{i=1}^{n}(xi - yi)^2)} \quad (1)$$

Keterangan :
Xi = pusat titik centroid
Yi = data *instances*
i = atribut data Covid-19



Gambar 2. Diagram alir Algoritma K-Means

d. Mengalokasikan masing-masing objek dalam centroid terdekat
e. Melakukan iterasi, serta mendapatkan centroid baru menggunakan Persamaan (2).

$$v = \frac{\sum_{i=1}^{n} x_i}{n} ; 1,2,3, \dots n \quad (2)$$

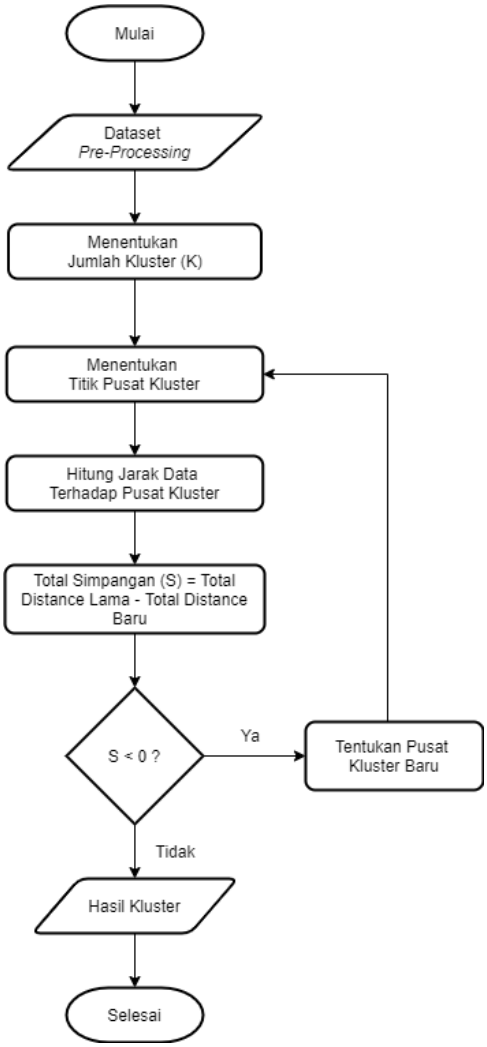
Keterangan :
Xi = pusat titik centroid
n = jumlah data Covid-19
i = atribut data Covid-19

f. Mengulangi langkah ketiga jika centroid baru tidak sama dengan centroid lama.

3.4. Perhitungan Algoritma K-Medoids

Gambar 3 merupakan diagram alir dari perhitungan algoritma K-Medoids. Proses perhitungan dimulai dari menentukan jumlah kluster, inisialisasi jumlah kluster, menghitung jarak *euclidean*, serta menghitung total jarak simpangan (S). Jarak simpangan (S) diperoleh dengan

menghitung selisih total jarak baru – total jarak lama. Jika jarak simpangan (S) > 0 maka proses perhitungan selesai. Akan tetapi, jika jarak simpangan (S) < 0 maka lakukan perhitungan sampai menemukan hasil > 0 kemudian proses perhitungan selesai.



Gambar 3. Diagram Alir Algoritma K-Medoids

Adapun urutan langkah-langkah algoritma K-Medoids berdasarkan penelitian Septiani [16], secara rinci sebagai berikut:

a. Memilih secara acak medoid awal sebanyak k dari n *instances* pada dataset sebaran Covid-19. Pada tahap ini dilakukan pemilihan medoid awal secara acak dari dataset yang telah di-*preprocessing*.
b. mengelompokkan setiap data Covid-19 ke *cluster* terdekat menggunakan pendekatan *Euclidian Distance* untuk menghitung jarak antar data Covid-19 dengan rumus pada Persamaan (3):

$$d(x,y) = \sqrt{(\sum_{i=1}^{n}(xi - yi)^2)} \quad (3)$$

Keterangan :
Xi = pusat titik medoid
Yi = data *instances*
i = atribut data Covid-19

c. Menandai jarak k terdekat objek data ke medoid dan menghitung totalnya. Untuk menghitung nilai kedekatannya, dengan cara mencari nilai minimum dari cost 1 dan cost 2 kemudian menghitung total kedekatannya.
d. Menentukan anggota *cluster* ke medoid sementara. Rumus untuk perhitungan *cluster* yaitu jika cost 1 = Minimal (Cost 1 dan Cost 2) maka bernilai (1,2)
e. Melakukan iterasi medoid. Untuk mencari iterasi medoid, langkah-langkah yang dilakukan yaitu memilih medoid sementara dan ikuti langkah-langkah diatas mulai dari langkah b, c dan d.
f. Kemudian menghitung total jarak simpangan (S) dengan menghitung nilai total *distance* baru – total *distance* lama. Apabila S < 0, maka tukar objek dengan data untuk membentuk sekumpulan k baru sebagai medoids. Lakukan iterasi sampai diperoleh nilai S > 0.

3.5. Komparasi Kinerja Algoritma Menggunakan DBI

Komparasi kinerja dari klusterisasi K-Means *Clustering* dan K-Medoids menggunakan Davies Bouldin Index sebagai metode evaluasinya. Mengacu pada penelitian Singh [17], metode evaluasi DBI memiliki kelebihan dalam hal mengukur evaluasi *cluster* pada suatu metode pengelompokan disebabkan nilai kohesi dan separasi yang dihasilkannya. Kohesi didefinisikan sebagai jumlah dari kedekatan data terhadap centroid dari *cluster* yang diikuti, dikenal sebagai *sum of square within cluster*. Sedangkan, separasi didasarkan pada jarak antar centroid dari *cluster*nya, dikenal sebagai *sum of square between kluster*. Melalui nilai kohesi dan separasi inilah yang menjadikan semakin kecil nilai DBI yang didapatkan (mendekati 0 atau sama dengan 0) maka menunjukkan semakin baik *cluster* dari hasil pengelompokan.

Perhitungan DBI diawali dengan mencari nilai melalui perhitungan nilai *sum of square within cluster* (SSW), yaitu untuk mengetahui matrik kohesi dalam sebuah *cluster* ke-*i* yang dirumuskan pada Persamaan (4).

$$SSW_i = \frac{1}{m_i}\sum_{j=1}^{m_i} d(x_j, c_i) \quad (4)$$

Keterangan :
SSW : sum of squere within *cluster*
$\boldsymbol{m_i}$: Jumlah data dalam kluster ke- *i*
$\boldsymbol{c_i}$: *Centroid cluster* ke-i
$\mathbf{d}(\boldsymbol{x_j}, \boldsymbol{c_i})$: Jarak dari data ke-i ke titik *cluster i*

Berikutnya, dilakukan langkah selanjutnya menggunakan Persamaan (5) untuk mengetahui separasi antar *cluster*.

$$SSB_{ij} = d(i,j) \quad (5)$$

Keterangan :

SSB : sum of squere between *cluster*

d(*i,j*) : Jarak *Euclidience distance* data ke I dan data ke-j

Selanjutnya, yaitu mencari Rasio dari hasil perhitungan SSW dan SSB seperti pada Persamaan (6):

$$R_{ij} = \frac{SSW_i + SSW_j}{SSB_{ij}} \quad (6)$$

Untuk proses perhitungan tahap akhir, Persamaan (7) digunakan untuk memperoleh nilai DBI.

$$DBI = \frac{1}{k}\sum_{i=1}^{k} \max i \neq j \, (R_{i,j}) \quad (7)$$

Keterangan :

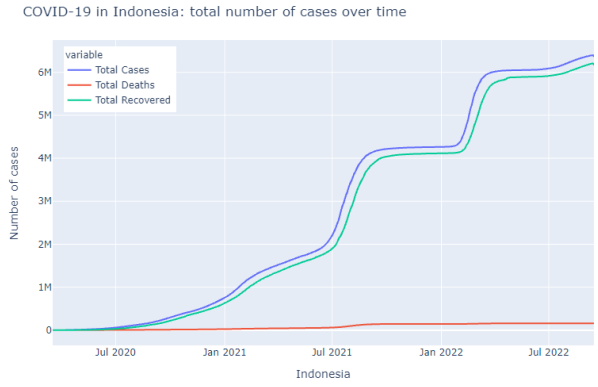
k : Jumlah *cluster* yang dihitung

Semakin kecil nilai DBI yang didapatkan, yakni nilai yang mendekati 0 atau sama dengan 0 maka menunjukkan semakin baik *cluster* dari hasil pengelompokan. Nantinya, nilai DBI didapatkan dari Algoritma K-Means dan algoritma K-Medoids. Kedua nilai DBI dikomparasi sebagai evaluasi kinerja dari klusterisasi.

4. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan Bahasa Pemrograman Python Version 3 yang dijalankan pada tools Jupyter Notebook. Langkah penelitian dimulai dari import library atau modul yang digunakan dalam berbagai tahapan dalam penelitian ini. Tahapan yang dilakukan antara lain melakukan pre-processing berupa proses binning data hingga normalisasi data. Selanjutnya, menampilkan visualisasi data sebaran Covid-19. Kemudian, melakukan modeling Algoritma K-Means dan K-Medoids. Hingga diakhiri dengan langkah terakhir berupa evaluasi menggunakan Davies-Bouldin Index (DBI).

4.1. Visualisasi Data



Gambar 4. Grafik Covid-19 di Indonesia dari waktu ke waktu

Gambar 4 menyajikan informasi berupa grafik baris mengenai perkembangan Covid-19 di Indonesia dari waktu ke waktu, mulai dari Maret 2020 hingga September 2022. Terdapat lonjakan yang signifikan pada total kasus yang diiringi oleh total kesembuhan. Untuk total kematian, terdapat lonjakan cukup signifikan pada bulan Juli 2021. Melalui grafik ini, dapat diketahui bahwa terdapat lonjakan kasus Covid-19 di Indonesia pada periode bulan-bulan tertentu.

4.2. Analisis Menggunakan K-Means

Tabel 1. Seleksi fitur pada proses modelling K-Means

```
# Modelling
# Seleksi Fitur
X = d[['Mortality', 'Total Cases','Total Active Cases', 'Population Density', 'Population', 'Total Deaths']]
```

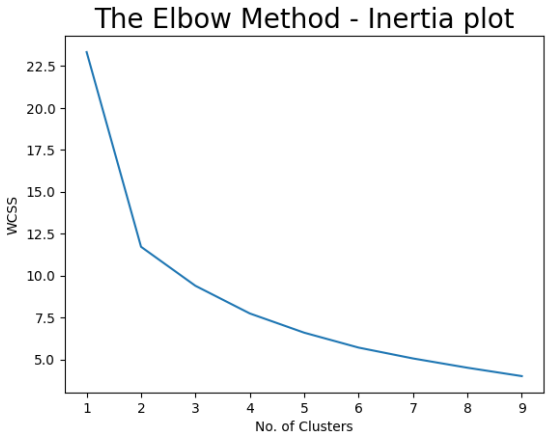
Tabel 1 merupakan *source code* untuk tahap seleksi fitur pada proses modelling dengan menggunakan dataset Covid-19 di Indonesia. Fitur-fitur yang dipilih untuk dilibatkan dalam model ini adalah **Mortality, Total Cases, Total Active Cases, Population Density, Population**, dan **Total Deaths**. Fitur-fitur tersebut dipilih karena mempengaruhi perkembangan kasus Covid-19 di Indonesia. Dalam *source code* tersebut, variabel X diisi dengan data dari fitur-fitur tersebut yang sudah dipilih. Data fitur-fitur tersebut digunakan sebagai input dalam model K-Means yang dibangun untuk memprediksi kasus Covid-19 di Indonesia.

Tabel 2. Metode Elbow untuk penentuan jumlah klaster optimal pada K-Means

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

#Elbow Method - Inertia plot
inertia = []
#looping the inertia calculation for each k
for k in range(1, 10):
    #Assign KMeans as cluster_model
    cluster_model = KMeans(n_clusters = k, random_state = 24)
    #Fit cluster_model to X
    cluster_model.fit(X)
    #Get the inertia value
    inertia_value = cluster_model.inertia_
    #Append the inertia_value to inertia list
    inertia.append(inertia_value)
##Inertia plot
plt.plot(range(1, 10), inertia)
plt.title('The Elbow Method - Inertia plot', fontsize = 20)
plt.xlabel('No. of Clusters')
plt.ylabel('WCSS')
plt.show()
```

Tabel 2 merupakan implementasi dari metode elbow untuk menentukan jumlah klaster yang optimal pada data sebaran Covid-19. Elbow method adalah salah satu teknik untuk menentukan jumlah klaster optimal dalam pengelompokkan data dengan algoritma K-Means. Pada *source code* tersebut, dilakukan looping dari k=1 hingga k=9, dimana pada setiap iterasi dibuat objek KMeans dengan jumlah klaster k yang bersesuaian. Kemudian objek tersebut dilatih dengan fitur X dan dihitung nilai inertia (*Within Cluster Sum of Squares*/WCSS) nya. Nilai inertia tersebut kemudian di-append ke dalam list inertia. Setelah proses loop selesai, nilai inertia diplot pada grafik Inertia plot dengan sumbu-x merepresentasikan jumlah klaster, dan sumbu-y merepresentasikan nilai inertia. Tujuan dari plot ini adalah untuk menentukan nilai k yang optimal, yaitu ketika penurunan nilai inertia mulai mengalami perubahan yang lebih lambat (bentuk seperti siku), sehingga sering disebut juga sebagai "elbow point", sebagaimana ditunjukkan pada Gambar 5.



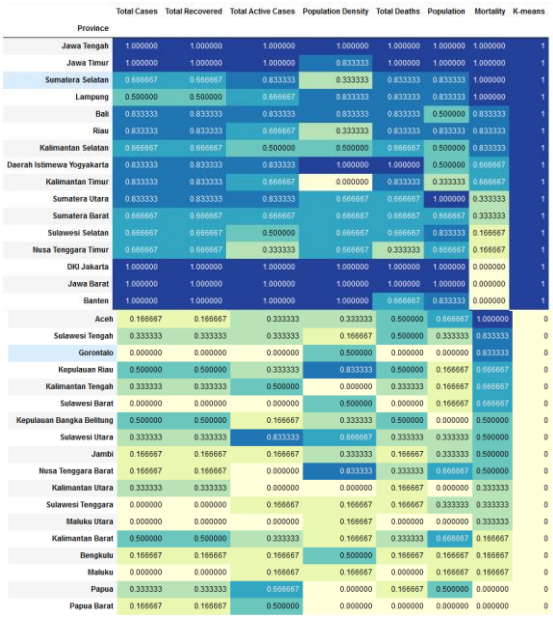
Gambar 5. Grafik hasil metode Elbow untuk penentuan jumlah klaster optimal pada K-Means

Berdasarkan Gambar 5, penentuan nilai k sebagai jumlah klaster yang optimal didapatkan ketika penurunan nilai inertia mulai mengalami perubahan yang lebih lambat, sehingga k yang optimal didapatkan pada nilai k = 2. Nilai k tersebut digunakan dalam proses selanjutnya, yakni modelling algoritma K-Means.

Tabel 3. Klastering menggunakan K-Means

```
kmeans = KMeans(n_clusters=2)
pred = kmeans.fit_predict(d[d.columns])
t['K-means'], d['K-means'] = [pred, pred]
d[d.columns].sort_values(['K-means',
ColumnData.mortality, ColumnData.cases,
ColumnData.actives_cases,
ColumnData.density],
ascending=False).style.background_gradient(
   cmap='YlGnBu', low=0, high=0.2)
```

Tabel 3 adalah *source code* untuk melakukan *clustering* dengan menggunakan K-Means dengan jumlah cluster sebanyak 2 (n_clusters=2) pada data COVID-19 Indonesia yang telah diolah sebelumnya. Selain itu, terdapat proses untuk menambahkan kolom baru bernama '**K-means**' pada data **t** dan **d**, yang berisi hasil prediksi klaster dari K-Means. Luaran dari *source code* tersebut adalah Gambar 6 yang menampilkan data **d** yang diurutkan berdasarkan klaster '**K-means**', '**Mortality**', '**Total Cases**', '**Total Active Cases**', dan '**Population Density**'. Pada Gambar 4.9 semakin gelap warna hijau pada sel data, semakin besar nilainya. Dari gambar tersebut, dapat dilihat bagaimana data terbagi menjadi dua kelompok (klaster) yang dihasilkan oleh K-Means.



Province	Total Cases	Total Recovered	Total Active Cases	Population Density	Total Deaths	Population	Mortality	K-means
Jawa Tengah	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
Jawa Timur	1.000000	1.000000	1.000000	0.833333	1.000000	1.000000	1.000000	1
Sumatera Selatan	0.666667	0.666667	0.833333	0.333333	0.833333	0.833333	1.000000	1
Lampung	0.500000	0.500000	0.666667	0.833333	0.833333	0.833333	1.000000	1
Bali	0.833333	0.833333	0.833333	0.833333	0.833333	0.500000	0.833333	1
Riau	0.833333	0.833333	0.666667	0.333333	0.833333	0.833333	0.833333	1
Kalimantan Selatan	0.666667	0.666667	0.500000	0.500000	0.666667	0.500000	0.833333	1
Daerah Istimewa Yogyakarta	0.833333	0.833333	0.833333	1.000000	1.000000	0.500000	0.666667	1
Kalimantan Timur	0.833333	0.833333	0.666667	0.000000	0.833333	0.333333	0.666667	1
Sumatera Utara	0.833333	0.833333	0.833333	0.666667	0.666667	1.000000	0.333333	1
Sumatera Barat	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.333333	1
Sulawesi Selatan	0.666667	0.666667	0.500000	0.666667	0.666667	0.833333	0.166667	1
Nusa Tenggara Timur	0.666667	0.666667	0.333333	0.666667	0.333333	0.666667	0.166667	1
DKI Jakarta	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1
Jawa Barat	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1
Banten	1.000000	1.000000	1.000000	1.000000	0.666667	0.833333	0.000000	1
Aceh	0.166667	0.166667	0.333333	0.333333	0.500000	0.666667	1.000000	0
Sulawesi Tengah	0.333333	0.333333	0.333333	0.166667	0.500000	0.333333	0.833333	0
Gorontalo	0.000000	0.000000	0.000000	0.500000	0.000000	0.000000	0.833333	0
Kepulauan Riau	0.500000	0.500000	0.333333	0.833333	0.500000	0.166667	0.666667	0
Kalimantan Tengah	0.333333	0.333333	0.500000	0.000000	0.333333	0.166667	0.666667	0
Sulawesi Barat	0.000000	0.000000	0.000000	0.500000	0.000000	0.166667	0.666667	0
Kepulauan Bangka Belitung	0.500000	0.500000	0.166667	0.333333	0.500000	0.000000	0.500000	0
Sulawesi Utara	0.333333	0.333333	0.833333	0.666667	0.333333	0.333333	0.500000	0
Jambi	0.166667	0.166667	0.166667	0.333333	0.166667	0.333333	0.500000	0
Nusa Tenggara Barat	0.166667	0.166667	0.000000	0.833333	0.333333	0.666667	0.500000	0
Kalimantan Utara	0.333333	0.333333	0.000000	0.000000	0.166667	0.000000	0.333333	0
Sulawesi Tenggara	0.000000	0.000000	0.166667	0.166667	0.166667	0.333333	0.333333	0
Maluku Utara	0.000000	0.000000	0.000000	0.166667	0.000000	0.000000	0.333333	0
Kalimantan Barat	0.500000	0.500000	0.333333	0.166667	0.333333	0.666667	0.166667	0
Bengkulu	0.166667	0.166667	0.166667	0.500000	0.166667	0.166667	0.166667	0
Maluku	0.000000	0.000000	0.166667	0.166667	0.000000	0.166667	0.166667	0
Papua	0.333333	0.333333	0.666667	0.000000	0.166667	0.500000	0.000000	0
Papua Barat	0.166667	0.166667	0.500000	0.000000	0.000000	0.000000	0.000000	0

Gambar 6. Hasil klasterisasi data sebaran Covid-19 di Indonesia menggunakan K-Means

Berdasarkan hasil klasterisasi data sebaran Covid-19 di Indonesia menggunakan K-Means, didapatkan bahwa sebaran Covid-19 di Indonesia terbagi menjadi 2 klaster. Jika dipetakan menurut provinsi, maka dapat dibagi sebagai berikut:

a. **Klaster 0**, antara lain: Aceh, Sulawesi Tengah, Gorontalo, Kepulauan Riau, Kalimantan Tengah, Kepulauan Bangka Belitung, Sulawesi Utara, Jambi, Nusa Tenggara Barat, Kalimantan Utara, Sulawesi Tenggara, Maluku Utara, Kalimantan Barat, Bengkulu, Maluku, Papua, Papua Barat
b. **Klaster 1,** antara lain: Jawa Tengah, Jawa Timur, Sumatera Selatan, Lampung, Bali, Riau, Kalimantan Selatan, Daerah Istimewa Yogyakarta, Kalimantan Timur, Sumatera Utara, Sumatera Barat, Sulawesi Selatan, Nusa Tenggara Timur, DKI Jakarta, Jawa Barat, Banten

Dalam klasterisasi sebaran Covid-19 menggunakan algoritma K-Means, provinsi di Indonesia masuk dalam klaster yang sama jika

atribut-atribut yang digunakan untuk klasterisasi memiliki kemiripan yang cukup tinggi di antara provinsi-provinsi tersebut sebagaimana ditampilkan pada Gambar 4.9. Kemiripan ini diukur berdasarkan jarak euclidean antara titik data. Misalnya, jika menggunakan atribut-atribut seperti **Total Cases, Total Recovered, Total Active Cases, Population Density, Total Deaths, Population**, dan **Mortality** sebagai fitur untuk klasterisasi, maka provinsi-provinsi yang memiliki nilai-nilai serupa untuk atribut-atribut ini cenderung masuk ke dalam klaster yang sama. Sebagai contoh, terdapat dua provinsi yang memiliki jumlah kasus Covid-19, angka kematian, jumlah kasus aktif, dan kepadatan penduduk yang hampir sama seperti Provinsi Jawa Timur dan Provinsi Jawa Tengah, algoritma K-Means menganggap kedua provinsi tersebut memiliki kemiripan yang tinggi dan akan menempatkannya dalam klaster yang sama

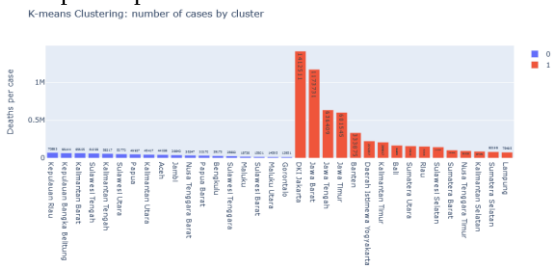
Tabel 4. *Source code treemap* untuk visualisasi hasil klasterisasi menggunakan K-Means

```
vis_tmap = px.treemap(t.reset_index(),
path=['K-means', ColumnData.province],
values=ColumnData.cases)
vis_tmap.update_layout(title='K-means clusters')
vis_tmap.show()
```

Source code pada Tabel 4 digunakan untuk membuat visualisasi *Treemap* menggunakan library *Plotly Express*. *Treemap* adalah jenis grafik yang menampilkan hirarki data dengan menggunakan persegi panjang dan sub persegi panjang untuk merepresentasikan setiap kategori. Visualisasi treemap menunjukkan jumlah kasus COVID-19 untuk setiap provinsi di Indonesia yang diklasifikasikan ke dalam dua cluster hasil dari algoritma K-means sebelumnya. Pada axis **horizontal (x)**, kita dapat melihat kedua klaster, yaitu 0 dan 1, sedangkan pada axis **vertical (y)**, kita dapat melihat nama-nama provinsi. Ukuran persegi panjang menunjukkan jumlah kasus COVID-19 di masing-masing provinsi. Luaran dari *source code* ini adalah sebuah visualisasi *treemap* yang menunjukkan dua klaster hasil dari algoritma K-means yang ditampilkan pada Gambar 7. Setiap persegi panjang pada *treemap* menunjukkan jumlah kasus COVID-19 untuk setiap provinsi di Indonesia.

Gambar 7. Visualisasi hasil klasterisasi K-Means menggunakan *treemap*

Hasil klasterisasi juga ditunjukkan pada *bar chart* yang menampilkan jumlah kasus Covid-19 di setiap provinsi di Indonesia, yang diurutkan berdasarkan cluster hasil dari K-means *clustering*. Setiap *cluster* ditampilkan dengan warna yang berbeda pada plot bar tersebut. Pada sumbu x terdapat daftar provinsi di Indonesia, sedangkan pada sumbu y terdapat jumlah kasus Covid-19. Terdapat juga nilai jumlah kasus yang ditampilkan pada setiap bar pada plot tersebut, sebagaimana ditampilkan pada Gambar 8.



Gambar 8. Visualisasi jumlah kasus Covid-19 di setiap provinsi di Indonesia

```
#Evaluasi DBI
from sklearn.metrics import davies_bouldin_score
import matplotlib.pyplot as plt

db_index = davies_bouldin_score(X, pred)
print(db_index)
```

```
0.9762331449809145
```

Gambar 9. Hasil evaluasi DBI untuk hasil klasterisasi menggunakan K-Means

Hasil evaluasi model *clustering* dengan menggunakan metode Davies-Bouldin Index (DBI) diperoleh sebagaimana pada Gambar 9 dengan menggunakan *source code* yang tertera pada gambar. DBI digunakan untuk mengukur kualitas pengelompokan data pada metode *clustering*. Semakin kecil nilai DBI, semakin baik pengelompokan data yang terjadi. Pada hasil output, didapatkan nilai DBI sebesar 0.9762331449809145. Nilai ini menunjukkan kualitas pengelompokan data yang baik karena mendekati 0. Semakin mendekati 0, semakin baik hasil pengelompokan data. Oleh karena itu, dapat dikatakan bahwa model *clustering* yang digunakan pada *source code* tersebut telah memberikan hasil yang baik dalam melakukan klasterisasi data sebaran Covid-19 di Indonesia.

4.3. Analisis Algoritma K-Medoids

Tabel 5. Seleksi fitur pada proses modelling K-Medoids

```
# Modelling
# Seleksi Fitur
X = d[['Mortality', 'Total Cases','Total Active Cases', 'Population Density', 'Population', 'Total Deaths']]
```

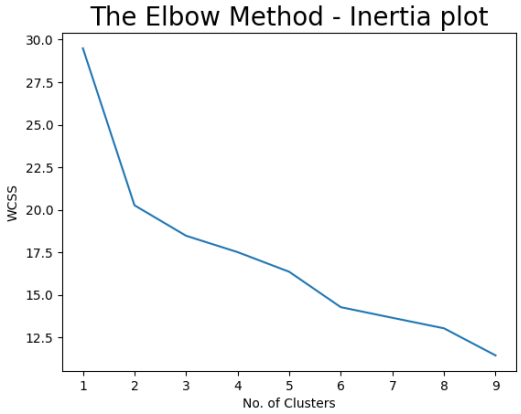
Tabel 5 merupakan *source code* untuk tahap seleksi fitur pada proses modelling dengan menggunakan dataset Covid-19 di Indonesia. Fitur-fitur yang dipilih untuk dilibatkan dalam model ini adalah **Mortality, Total Cases, Total Active Cases, Population Density, Population**, dan **Total Deaths**. Fitur-fitur tersebut dipilih karena mempengaruhi perkembangan kasus Covid-19 di Indonesia. Dalam *source code* tersebut, variabel X diisi dengan data dari fitur-fitur tersebut yang sudah dipilih. Data fitur-fitur tersebut digunakan sebagai input dalam model K-Medoids.

Tabel 6. Metode Elbow untuk penentuan jumlah klaster optimal pada K-Means

```
from sklearn_extra.cluster import KMedoids
import matplotlib.pyplot as plt

#Elbow Method - Inertia plot
inertia = []
#looping the inertia calculation for each k
for k in range(1, 10):
   #Assign KMeans as cluster_model
   cluster_model = KMedoids(n_clusters = k, random_state = 24)
   #Fit cluster_model to X
   cluster_model.fit(X)
   #Get the inertia value
   inertia_value = cluster_model.inertia_
   #Append the inertia_value to inertia list
   inertia.append(inertia_value)
##Inertia plot
plt.plot(range(1, 10), inertia)
plt.title('The Elbow Method - Inertia plot', fontsize = 20)
plt.xlabel('No. of Clusters')
plt.ylabel('WCSS')
plt.show()
```

Tabel 6 merupakan implementasi dari metode elbow untuk menentukan jumlah klaster yang optimal pada data sebaran Covid-19. Elbow method adalah salah satu teknik untuk menentukan jumlah klaster optimal dalam pengelompokkan data dengan algoritma K-Medoids. Pada *source code* tersebut, dilakukan looping dari k=1 hingga k=9, dimana pada setiap iterasi dibuat objek KMedoids dengan jumlah klaster k yang bersesuaian. Kemudian objek tersebut dilatih dengan fitur X dan dihitung nilai inertia (*Within Cluster Sum of Squares*/WCSS) nya. Nilai inertia tersebut kemudian di-append ke dalam list inertia. Setelah proses loop selesai, nilai inertia diplot pada grafik Inertia plot dengan sumbu-x merepresentasikan jumlah klaster, dan sumbu-y merepresentasikan nilai inertia. Tujuan dari plot ini adalah untuk menentukan nilai k yang optimal, yaitu ketika penurunan nilai inertia mulai mengalami perubahan yang lebih lambat (bentuk seperti siku), sehingga sering disebut juga sebagai "elbow point" yang ditampilkan pada Gambar 4.14.



Gambar 10. Grafik hasil metode Elbow untuk penentuan jumlah klaster optimal pada K-Medoids

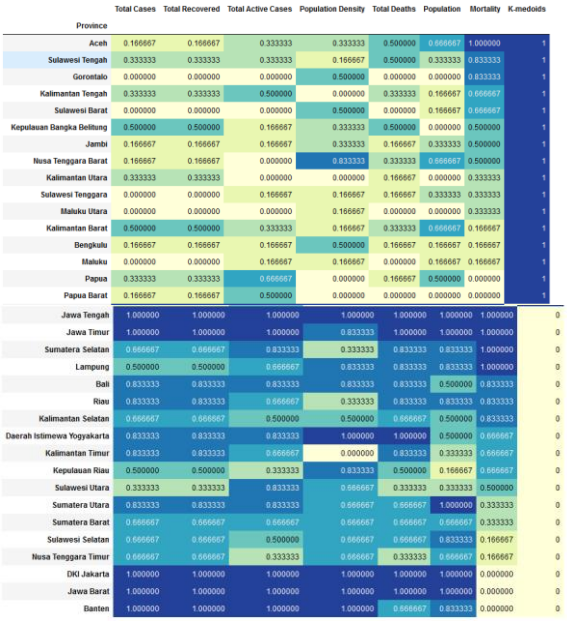
Berdasarkan Gambar 4.14, penentuan nilai k yang optimal didapatkan ketika penurunan nilai inertia mulai mengalami perubahan yang lebih lambat, sehingga k yang optimal didapatkan pada nilai k = 2. Nilai k tersebut digunakan dalam proses selanjutnya, yakni modelling algoritma K-Medoids.

Tabel 7. Klastering menggunakan K-Means

```
kmedoids = KMedoids(n_clusters=2)
pred = kmedoids.fit_predict(d[d.columns])
t['K-medoids'], d['K-medoids'] = [pred, pred]
d[d.columns].sort_values(['K-medoids', ColumnData.mortality, ColumnData.cases, ColumnData.actives_cases, ColumnData.density], ascending=False).style.background_gradient(
   cmap='YlGnBu', low=0, high=0.2)
```

Tabel 7 adalah *source code* untuk melakukan *clustering* dengan menggunakan K-Means dengan jumlah cluster sebanyak 2 (n_clusters=2) pada data COVID-19 Indonesia yang telah diolah sebelumnya. Selain itu, terdapat proses untuk menambahkan kolom baru bernama **'K-medoids'** pada data **t** dan **d**, yang berisi hasil prediksi klaster dari K-Medoids. Luaran dari *source code* tersebut adalah Gambar 11 yang menampilkan data **d** yang diurutkan berdasarkan klaster **'K-means', 'Mortality', 'Total Cases', 'Total Active Cases'**, dan **'Population Density'**. Pada Gambar 11 semakin gelap warna pada sel data, semakin besar nilainya. Dari gambar tersebut, dapat dilihat

bagaimana data terbagi menjadi dua kelompok (klaster) yang dihasilkan oleh K-Medoids.



Province	Total Cases	Total Recovered	Total Active Cases	Population Density	Total Deaths	Population	Mortality	K-medoids
Aceh	0.166667	0.166667	0.333333	0.333333	0.500000	0.666667	1.000000	1
Sulawesi Tengah	0.333333	0.333333	0.333333	0.166667	0.500000	0.333333	0.833333	1
Gorontalo	0.000000	0.000000	0.000000	0.500000	0.000000	0.000000	0.833333	1
Kalimantan Tengah	0.333333	0.333333	0.500000	0.000000	0.333333	0.166667	0.666667	1
Sulawesi Barat	0.000000	0.000000	0.000000	0.500000	0.000000	0.166667	0.666667	1
Kepulauan Bangka Belitung	0.500000	0.500000	0.166667	0.333333	0.500000	0.000000	0.500000	1
Jambi	0.166667	0.166667	0.166667	0.333333	0.166667	0.333333	0.500000	1
Nusa Tenggara Barat	0.166667	0.166667	0.000000	0.833333	0.333333	0.666667	0.500000	1
Kalimantan Utara	0.333333	0.333333	0.000000	0.000000	0.166667	0.000000	0.333333	1
Sulawesi Tenggara	0.000000	0.000000	0.166667	0.166667	0.166667	0.333333	0.333333	1
Maluku Utara	0.000000	0.000000	0.000000	0.166667	0.000000	0.000000	0.333333	1
Kalimantan Barat	0.500000	0.500000	0.333333	0.166667	0.333333	0.666667	0.166667	1
Bengkulu	0.166667	0.166667	0.166667	0.500000	0.166667	0.166667	0.166667	1
Maluku	0.000000	0.000000	0.166667	0.166667	0.000000	0.166667	0.166667	1
Papua	0.333333	0.333333	0.666667	0.000000	0.166667	0.500000	0.000000	1
Papua Barat	0.166667	0.166667	0.500000	0.000000	0.000000	0.000000	0.000000	1
Jawa Tengah	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0
Jawa Timur	1.000000	1.000000	1.000000	0.833333	1.000000	1.000000	1.000000	0
Sumatera Selatan	0.666667	0.666667	0.833333	0.333333	0.833333	0.833333	1.000000	0
Lampung	0.500000	0.500000	0.666667	0.833333	0.833333	0.833333	1.000000	0
Bali	0.833333	0.833333	0.833333	0.833333	0.833333	0.500000	0.833333	0
Riau	0.833333	0.833333	0.666667	0.333333	0.833333	0.833333	0.833333	0
Kalimantan Selatan	0.666667	0.666667	0.50000	0.500000	0.666667	0.500000	0.833333	0
Daerah Istimewa Yogyakarta	0.833333	0.833333	0.833333	1.000000	1.000000	0.500000	0.666667	0
Kalimantan Timur	0.833333	0.833333	0.666667	0.000000	0.833333	0.333333	0.666667	0
Kepulauan Riau	0.500000	0.500000	0.333333	0.833333	0.500000	0.166667	0.666667	0
Sulawesi Utara	0.333333	0.333333	0.833333	0.666667	0.333333	0.333333	0.500000	0
Sumatera Utara	0.833333	0.833333	0.833333	0.666667	0.666667	1.000000	0.333333	0
Sumatera Barat	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.333333	0
Sulawesi Selatan	0.666667	0.666667	0.500000	0.666667	0.666667	0.833333	0.166667	0
Nusa Tenggara Timur	0.666667	0.666667	0.333333	0.666667	0.333333	0.666667	0.166667	0
DKI Jakarta	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0
Jawa Barat	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0
Banten	1.000000	1.000000	1.000000	1.000000	0.666667	0.833333	0.000000	0

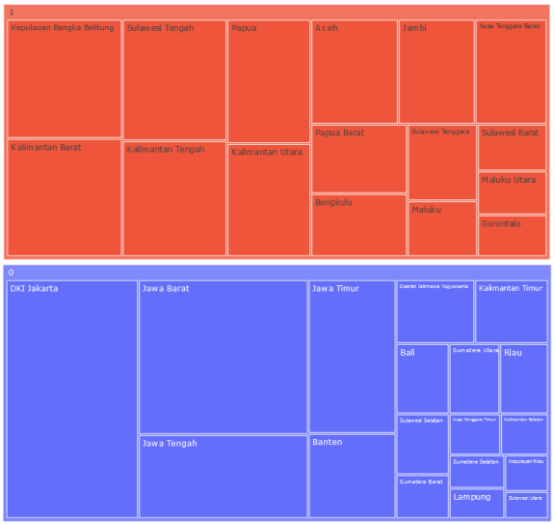
Gambar 11. Hasil klasterisasi data sebaran Covid-19 di Indonesia menggunakan K-Medoids

Berdasarkan hasil klasterisasi data sebaran Covid-19 di Indonesia menggunakan K-Medoids, didapatkan bahwa sebaran Covid-19 di Indonesia terbagi menjadi 2 klaster. Dalam klasterisasi, provinsi di Indonesia masuk dalam klaster yang sama jika atribut-atribut yang digunakan untuk klasterisasi memiliki kemiripan yang cukup tinggi di antara provinsi-provinsi tersebut sebagaimana ditampilkan pada Gambar 11. Kemiripan ini diukur berdasarkan jarak euclidean antara titik data. Misalnya, jika menggunakan atribut-atribut seperti **Total Cases, Total Recovered, Total Active Cases, Population Density, Total Deaths, Population**, dan **Mortality** sebagai fitur untuk klasterisasi, maka provinsi-provinsi yang memiliki nilai-nilai serupa untuk atribut-atribut ini cenderung masuk ke dalam klaster yang sama. Sebagai contoh, terdapat dua provinsi yang memiliki jumlah kasus Covid-19, angka kematian, jumlah kasus aktif, dan kepadatan penduduk yang hampir sama seperti Provinsi Jawa Timur dan Provinsi Jawa Tengah, algoritma K-Medoids menganggap kedua provinsi tersebut memiliki kemiripan yang tinggi dan akan menempatkannya dalam klaster yang sama

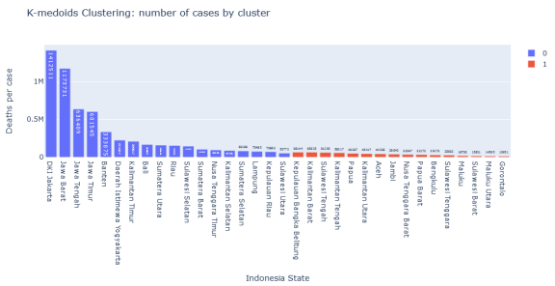
Source code pada Tabel 8 digunakan untuk membuat visualisasi *Treemap* menggunakan library *Plotly Express*. *Treemap* adalah jenis grafik yang menampilkan hirarki data dengan menggunakan persegi panjang dan sub persegi panjang untuk merepresentasikan setiap kategori. Visualisasi treemap menunjukkan jumlah kasus COVID-19 untuk setiap provinsi di Indonesia yang diklasifikasikan ke dalam dua cluster hasil dari algoritma K-medoids sebelumnya. Pada axis **horizontal (x)**, kita dapat melihat kedua klaster, yaitu 0 dan 1, sedangkan pada axis **vertical (y)**, kita dapat melihat nama-nama provinsi. Ukuran persegi panjang menunjukkan jumlah kasus COVID-19 di masing-masing provinsi. Luaran dari *source code* ini adalah sebuah visualisasi *treemap* yang menunjukkan dua klaster hasil dari algoritma K-medoids yang ditampilkan pada Gambar 4.16. Setiap persegi panjang pada *treemap* menunjukkan jumlah kasus COVID-19 untuk setiap provinsi di Indonesia.

Tabel 8. *Source code treemap* untuk visualisasi hasil klasterisasi menggunakan K-Medoids

```
vis_tmap = px.treemap(t.reset_index(),
path=['K-medoids', ColumnData.province],
values=ColumnData.cases)
vis_tmap.update_layout(title='K-medoids
clusters')
vis_tmap.show()
```



Gambar 13. Visualisasi hasil klasterisasi K-Medoids menggunakan *treemap*



Gambar 14. Visualisasi jumlah kasus Covid-19 di setiap provinsi di Indonesia menggunakan K-Medoids

Terdapat *bar chart* yang menampilkan jumlah kasus Covid-19 di setiap provinsi di Indonesia, yang diurutkan berdasarkan cluster hasil dari K-medoids *clustering*. Setiap *cluster* ditampilkan dengan warna yang berbeda pada plot bar tersebut. Pada sumbu x terdapat daftar provinsi di Indonesia, sedangkan

pada sumbu y terdapat jumlah kasus Covid-19. Terdapat juga nilai jumlah kasus yang ditampilkan pada setiap bar pada plot tersebut, sebagaimana ditampilkan pada Gambar 14.

```
#Evaluasi DBI
from sklearn.metrics import davies_bouldin_score
import matplotlib.pyplot as plt

db_index = davies_bouldin_score(X, pred)
print(db_index)
```

```
0.9809235412405508
```

Gambar 15. Hasil evaluasi DBI untuk hasil klasterisasi menggunakan K-Medoids

Evaluasi model K-Medoids dengan menggunakan metode Davies-Bouldin Index (DBI). DBI digunakan untuk mengukur kualitas pengelompokan data pada metode *clustering*. Semakin kecil nilai DBI, semakin baik pengelompokan data yang terjadi. Pada hasil output, didapatkan nilai DBI sebesar 0.9809235412405508 seperti ditunjukkan pada Gambar 15. Nilai ini menunjukkan kualitas pengelompokan data yang baik karena mendekati 0. Semakin mendekati 0, semakin baik hasil pengelompokan data. Oleh karena itu, dapat dikatakan bahwa model *clustering* yang digunakan pada *source code* tersebut telah memberikan hasil yang baik dalam melakukan klasterisasi data sebaran Covid-19 di Indonesia.

4.4. Perbandingan K-Means dan K-Medoids

DBI (Davies-Bouldin Index) adalah metrik evaluasi yang digunakan untuk mengukur kualitas partisi dalam *clustering*. Semakin rendah nilai DBI, semakin baik partisi yang dihasilkan oleh model. Dalam penelitian ini, K-Means dan K-Medoids adalah dua metode *clustering* yang digunakan untuk mempartisi data sebaran Covid-19 di Indonesia. Setelah dilakukan evaluasi DBI, K-Means mendapatkan nilai 0.9762331449809145, sedangkan K-Medoids mendapatkan nilai 0.9809235412405508. Karena K-Means memiliki nilai DBI yang lebih rendah dibandingkan K-medoids, maka dapat dikatakan K-Means menghasilkan klasterisasi yang lebih baik dalam klasterisasi data sebaran Covid-19 di Indonesia.

5. KESIMPULAN DAN SARAN

Penelitian ini menganalisis sebaran Covid-19 di Indonesia menggunakan Algoritma K-Means dan K-Medoids, adapun langkah penelitian dimulai dari *import library* atau modul menggunakan bahasa pemrograman Python versi 3. Tahapan yang dilakukan antara lain melakukan *pre-processing* berupa proses binning data hingga normalisasi data. Selanjutnya, menampilkan visualisasi data sebaran Covid-19. Dalam melakukan modeling Algoritma K-Means dan K-Medoids, penentuan nilai k sebagai jumlah klaster yang optimal didapatkan menggunakan Metode Elbow. Hal ini ditandai ketika penurunan nilai inertia mulai mengalami perubahan yang lebih lambat, sehingga k yang optimal didapatkan pada nilai k = 2. Nilai k tersebut digunakan dalam proses selanjutnya, yakni modelling algoritma K-Means dan K-Medoids. Dalam klasterisasi sebaran Covid-19 menggunakan algoritma K-Means dan K-Medoids, provinsi di Indonesia masuk dalam klaster yang sama jika atribut-atribut yang digunakan untuk klasterisasi memiliki kemiripan yang cukup tinggi di antara provinsi-provinsi tersebut. Kemiripan ini diukur berdasarkan jarak euclidean antara titik data menggunakan atribut-atribut seperti Total Cases, Total Recovered, Total Active Cases, Population Density, Total Deaths, Population, dan Mortality sebagai fitur untuk klasterisasi. DBI (Davies-Bouldin Index) adalah metrik evaluasi yang digunakan untuk mengukur kualitas partisi dalam *clustering* data sebaran Covid-19. Hasilnya, semakin rendah nilai DBI, semakin baik partisi yang dihasilkan oleh model. Dalam penelitian ini, K-Means dan K-Medoids adalah dua metode *clustering* yang digunakan untuk mempartisi data sebaran Covid-19 di Indonesia. Setelah dilakukan evaluasi DBI, K-Means mendapatkan nilai 0.9762331449809145, sedangkan K-Medoids mendapatkan nilai 0.9809235412405508. Karena K-Means memiliki nilai DBI yang lebih rendah dibandingkan K-medoids, maka dapat dikatakan K-Means menghasilkan klasterisasi yang lebih baik dalam klasterisasi data sebaran Covid-19 di Indonesia.

DAFTAR PUSTAKA

[1] M. Z. Rodriguez *et al.*, *Clustering algorithms: A comparative approach*, vol. 14, no. 1. 2019. doi: 10.1371/journal.pone.0210236.

[2] K. Deswiaqsa, E. Darmawan, and S. Sugiyarto, "Application of K-Means for Clustering Based on the Severity of COVID-19 in Indonesian Private Hospitals," *EKSAKTA J. Sci. Data Anal.*, vol. 3, no. 2, pp. 95–102, 2022, doi: 10.20885/eksakta.vol3.iss2.art5.

[3] P. M. A. Putra and I. G. A. G. A. Kadyanan, "Implementation of K-Means Clustering Algorithm in Determining Classification of the Spread of the COVID-19 Virus in Bali," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 10, no. 1, p. 11, 2021, doi: 10.24843/jlk.2021.v10.i01.p03.

[4] A. D. Andini and T. Arifin, "Implementasi Algoritma K-Medoids Untuk Klasterisasi Data Penyakit Pasien Di Rsud Kota Bandung," *J. RESPONSIF Ris. Sains ...*, vol. 2, no. 2, pp. 128–138, 2020.

[5] F. Virgantari and Y. E. Faridhan, "K-Means Clustering of COVID-19 Cases in Indonesia's Provinces," *ADRI Int. J. Eng. Nat. Sci.*, vol. 5, no. 2, pp. 34–39, 2020, doi: 10.29138/aijens.v5i2.15.

[6] S. Sindi, W. R. O. Ningse, I. A. Sihombing, F. I. R.H.Zer, and D. Hartama, "Analisis Algoritma K-Medoids Clustering Dalam Pengelompokan Penyebaran Covid-19 Di Indonesia," *J. Teknol. Inf.*, vol. 4, no. 1, pp. 166–173, 2020, doi: 10.36294/jurti.v4i1.1296.

[7] C. M. Annur, "Tingkat Kematian Akibat Covid-19 di Indonesia Capai 2,58%, Peringkat Berapa di ASEAN?," *katadata.com*, 2022.

[8] Qomariyah and M. U. Siregar, "Comparative Study of K-Means Clustering Algorithm and K-Medoids Clustering in Student Data Clustering," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 7, no. 2, pp. 91–99, 2022, doi: 10.14421/jiska.2022.7.2.91-99.

[9] N. Suarna, Y. A. Wijaya, Mulyawan, T. Hartati, and T. Suprapti, "Comparison K-Medoids Algorithm and K-Means Algorithm for Clustering Fish Cooking Menu from Fish Dataset," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1088, no. 1, p. 012034, 2021, doi: 10.1088/1757-899x/1088/1/012034.

[10] Nurhayati, N. S. Sinatrya, L. K. Wardhani, and Busman, "Analysis of K-Means and K-Medoids's Performance Using Big Data Technology," *2018 6th Int. Conf. Cyber IT Serv. Manag. CITSM 2018*, no. Citsm, pp. 1–5, 2019, doi: 10.1109/CITSM.2018.8674251.

[11] D. Abdullah, S. Susilo, A. S. Ahmar, R. Rusli, and R. Hidayat, "The application of K-means clustering for province clustering in Indonesia of the risk of the COVID-19 pandemic based on COVID-19 data," *Qual. Quant.*, vol. 56, no. 3, pp. 1283–1291, 2022, doi: 10.1007/s11135-021-01176-w.

[12] R. Silvi, "Analisis Cluster dengan Data Outlier Menggunakan Centroid Linkage dan K-Means Clustering untuk Pengelompokkan Indikator HIV/AIDS di Indonesia," *J. Mat. "MANTIK,"* vol. 4, no. 1, pp. 22–31, 2018, doi: 10.15642/mantik.2018.4.1.22-31.

[13] A. Supriyadi, A. Triayudi, and I. D. Sholihati, "Perbandingan Algoritma K-Means Dengan K-Medoids Pada Pengelompokan Armada Kendaraan Truk Berdasarkan Produktivitas," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 6, no. 2, pp. 229–240, 2021, doi: 10.29100/jipi.v6i2.2008.

[14] A. Fira, C. Rozikin, and G. Garno, "Komparasi Algoritma K-Means dan K-Medoids Untuk Pengelompokkan Penyebaran Covid-19 di Indonesia," *J. Appl. Informatics Comput.*, vol. 5, no. 2, pp. 133–138, 2021, doi: 10.30871/jaic.v5i2.3286.

[15] R. Adha, N. Nurhaliza, U. Sholeha, and M. Mustakim, "Perbandingan Algoritma DBSCAN dan K-Means Clustering untuk Pengelompokan Kasus Covid-19 di Dunia," *SITEKIN J. Sains, Teknol. dan Ind.*, vol. 18, no. 2, pp. 206–211, 2021.

[16] I. W. Septiani, A. C. Fauzan, and M. M. Huda, "Implementasi Algoritma K-Medoids Dengan Evaluasi Davies-Bouldin- Index Untuk Klasterisasi Harapan Hidup Pasca Operasi Pada Pasien Penderita Kanker Paru-Paru," *J. Sist. Komput. dan Inform.*, vol. 3, no. 4, pp. 556–566, 2022, doi: 10.30865/json.v3i4.4055.

[17] A. K. Singh, S. Mittal, P. Malhotra, and Y. V. Srivastava, "Clustering Evaluation by Davies-Bouldin Index(DBI) in Cereal data using K-Means," *Proc. 4th Int. Conf. Comput. Methodol. Commun. ICCMC 2020*, no. Iccmc, pp. 306–310, 2020, doi: 10.1109/ICCMC48092.2020.ICCMC-00057.