Implementasi Data Mining untuk Mengklasterisasi Tingkat Kemiskinan di Indonesia

Disusun Untuk Memenuhi Tugas Besar Mata Kuliah Data Mining

Oleh Kelompok 7 :

Najla Humaira Desni	2211521002
Rizka Kurnia Illahi	2211521012
Fadli Hidayat	2211522010
Daffa Agustian Saadi	2211523022
Muhammad Fariz	2211523034

Dosen Pengampu:

Aina Hubby Aziira. M.Eng

Dwi Welly Sukma Nirad. MT,

DEPARTEMEN SISTEM INFORMASI

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

TA 2023/2024

KATA PENGANTAR

Dengan nama Allah Yang Maha Pengasih lagi Maha Penyayang, kami panjatkan puji dan syukur kehadirat-Nya atas segala limpahan rahmat, hidayah, serta karunia-Nya yang telah melimpah kepada kami sehingga kami dapat menyelesaikan tugas besar ini. Laporan ini merupakan hasil kerja keras dan kolaborasi tim kami dalam memahami dan mengimplementasikan Data Mining untuk "Mengklasterisasi Tingkat Kemiskinan di Indonesia". Kami berharap laporan ini dapat memberikan manfaat dan wawasan bagi pembaca serta menjadi referensi yang bermanfaat dalam memahami implementasi data mining untuk mengklasterisasi tingkat kemiskinan di Indonesia.

Kami mengucapkan terima kasih kepada dosen pengampu, Ibuk Aina Hubby Aziira, M.Eng, dan Ibuk Dwi Welly Sukma Nirad, MT., yang telah memberikan bimbingan dan dukungan selama proses pembuatan laporan ini. Kami juga berterima kasih kepada teman-teman sekelas yang telah berkontribusi dalam diskusi dan sharing ilmu. Semoga laporan ini dapat memberikan manfaat dan menjadi bagian dari upaya kita dalam mengembangkan pemahaman tentang Data Mining dan penerapannya dalam kehidupan sehari-hari.

Padang, 27 Mei 2024

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Di tengah dinamika pertumbuhan ekonomi global, Indonesia masih menghadapi tantangan serius dalam mengatasi masalah kemiskinan. Masalah kemiskinan merupakan salah satu permasalahan mendesak di Indonesia yang sering kali ditandai dengan tingginya tingkat pengangguran dan keterbelakangan. Kemiskinan merupakan salah satu persoalan mendasar dan masalah yang sulit untuk diatasi di seluruh negara yang ada didunia termasuk di Indonesia (Purnomo, 2021). Hal ini karena, kemiskinan memiliki sifat yang kompleks artinya kemiskinan yang ada tidak muncul sendiri secara tiba-tiba akan tetapi dipengaruhi oleh berbagai latar belakang yang ada (Parwa & Yasa, 2019).

Menurut Badan Pusat Statistik (BPS) pada Maret 2023, angka kemiskinan nasional masih mencapai 9,36 persen. Menanggapi hal ini pemerintah melakukan program bantuan sosial agar dapat menurunkan tingkat kemiskinan di Indonesia. Pelaksanaan bansos dari tahun ke tahun menunjukkan perbaikan dengan adanya penurunan angka kemiskinan. Meskipun sudah menunjukkan perbaikan, dalam pelaksanaannya, program bansos masih menghadapi berbagai tantangan yang berpotensi menurunkan efektivitas program.

Tantangan utama pada program bansos adalah masih besarnya salah sasaran (targeting error), baik inclusion maupun exclusion error. Masalah pada targeting tersebut akan membuat komplementaritas antar program dengan masih sedikitnya kelompok desil terbawah yang menerima lebih dari satu program serta keluarga di kelompok 20 persen berpenghasilan terendah yang belum mendapatkan bansos, sebaliknya terdapat keluarga di kelompok menengah dan kaya yang menerima bansos (Kemenkeu 2021). Merujuk pada data dan laporan pemerintah tersebut, kami menggunakan data mining untuk pengelompokan status kemiskinan di Indonesia sehingga dapat menyelesaikan permasalahan salah sasaran (targeting error) yang menjadi tantangan terbesar pemerintah.

Data Mining merupakan proses yang menggunakan berbagai teknik dan analisis data untuk menemukan hubungan dan pola tersembunyi. (Sudirman, Windarto dan Wanto,2018) Penggunaan data mining telah terbukti menjadi alat yang efektif dalam menganalisis pola-pola kompleks dalam kumpulan data besar. Data mining dapat memberikan pengetahuan baru yang berasal dari pengelolaan data yang ada, dengan mengidentifikasi pola-pola yang berguna dalam memahami kondisi kemiskinan. Dalam penelitian ini menggunakan metode klasterisasi dengan menerapkan algoritma K-Means, Agglomerative Hierarchical Clustering (AHC), dan DBSCAN. dengan tujuan memberikan data yang lebih akurat tentang tingkat kemiskinan di Indonesia

Dengan demikian, implementasi data mining dapat menjadi alat yang efektif dalam mengklasterisasi tingkat kemiskinan penduduk di Indonesia sehingga penyaluran bantuan sosial dilakukan dengan tepat sasaran. Penggunaan data mining diharapkan dapat menghasilkan data yang lebih akurat tentang tingkat kemiskinan di Indonesia dan membantu pemerintah dalam menargetkan bansos kepada masyarakat yang benar-benar membutuhkan dalam upaya pengentasan kemiskinan yang dilakukan pemerintah di Indonesia.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah di uraikan di atas maka terdapat rumusan masalah untuk laporan ini adalah bagaimana penerapan teknik data mining untuk mengklasterisasi tingkat kemiskinan di Indonesia dan membandingkan algoritma klasterisasi yang paling sesuai.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Data yang digunakan dalam project ini dibatasi hanya data kemiskinan di Indonesia yang didapatkan dari kaggle

2. Data yang telah diolah akan diklasterisasi dengan algoritma K-Means, Agglomerative Hierarchical Clustering (AHC), dan DBSCAN

1.4 Tujuan

Tujuan dari laporan ini yaitu :

1. Untuk memenuhi Tugas Besar Mata Kuliah Data Mining
2. Menerapkan data mining untuk mengklasterisasi tingkat kemiskinan di Indonesia
3. Dapat mengetahui algoritma pada metode klasterisasi yang paling sesuai untuk mengelompokkan data kemiskinan penduduk di Indonesia.

1.5 Manfaat

Manfaat yang diharapkan dari penelitian ini yaitu sebagai berikut:

1. Dapat memberikan pemahaman yang lebih dalam tentang pola dan faktor-faktor yang mempengaruhi kemiskinan di Indonesia. Dengan demikian, akan lebih mudah bagi pemerintah dan lembaga terkait untuk merancang kebijakan yang lebih efektif dalam upaya mengurangi tingkat kemiskinan di Indonesia.
2. Dapat memberikan dasar yang kuat bagi pengambilan kebijakan yang terarah dan efektif dalam upaya pengentasan kemiskinan di Indonesia
3. Dapat menghasilkan data yang akurat agar pemberian bantuan sosial lebih tepat sasaran
4. Dapat memberikan kontribusi terhadap pengembangan ilmu pengetahuan dalam bidang data mining dan aplikasinya dalam konteks sosial ekonomi seperti penanganan kemiskinan.

BAB II
METODOLOGI PENELITIAN

Bab ini menjelaskan teori, pendekatan, dan metode yang digunakan dalam penelitian untuk mencapai tujuan yang telah ditetapkan.

2.1 Kemisikinan

Kemiskinan merupakan masalah utama bagi banyak negara di du nia, terutama di negara berkembang. Kemiskinan merupakan kondisi dimana seseorang yang tidak dapat memenuhi kebutuhan dasar seperti makanan, pakaian, obat-obatan dan tempat tinggal (Hardinandar, 2019). Salah satu ukuran kondisi sosial dan ekonomi dalam menilai keberhasilan pembangunan pemerintah di suatu daerah adalah adanya kemiskinan itu sendiri (Oktaviana et al., 2021).

2.2 *Data Mining*

Penelitian ini menggunakan penerapan *data mining* yang mempunyai pengertian, tujuan dan manfaat *data mining*. Hal tersebut menjadi alasan untuk menerapkan data mining pada penelitian. Data akan diolah dengan mengikuti tahap *data mining* dan mendapatkan hasil yang mengacu pada tujuan serta manfaat *data mining*.

2.2.1 Pengertian *Data Mining*

*Data minin*g adalah proses yang menggunakan statistik, matematika, kecerdasan buatan dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar (Turban ,2005).

2.2.2 Tujuan *Data Mining*

*Data minin*g dalam pengimplementasiannya memiliki beberapa tujuan(Hoffer et al.,2007):

1. *Explanatory*: Menjelaskan beberapa kondisi penelitian.
2. *Confirmatory*: Mempertegas hipotesis.

3. *Exploratory*: Menganalisis data yang memiliki hubungan yang baru.

2.2.3 Manfaat Data Mining

Metode data mining telah banyak diimplementasikan di dalam dunia nyata untuk menyelesaikan berbagai macam permasalahan manusia,berikut adalah beberapa manfaat dari data mining (Adinugroho & Sari, 2018):

1. Pengindareraan jarak jauh melalui citra satelit merupakan hasil implementasi metode klasifikasi.
2. Mengelompokkan konsumen berdasarkan karakteristik demografis, perilaku, atau preferensi.

Data mining telah lama digunakan di dalam dunia bisnis untuk meningkatkan keuntungan. Pola-pola belanja konsumen dapat dipetakan menggunakan beberapa metode seperti *clustering*, *association rule mining*, dan *classification*.

2.3 Clustering

Clustering adalah proses mengelompokkan atau penggolongan objek berdasarkan informasi yang diperoleh dari data yang menjelaskan hubungan antar objek dengan prinsip untuk memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas/*cluster*. Clustering membagi data ke dalam grup-grup yang mempunyai obyek yang karakteristiknya sama. Beberapa algoritma yang dapat digunakan untuk clustering yaitu *K-Means, Agglomerative Hirarchial*, dan DBSCAN.

2.3.1 *K-Means*

Algoritma *K-means* merupakan teknik *data mining* yaitu metode *clustering* atau pengelompokkan yang proses pemodelannya tanpa supervisi/pembelajaran serta metode pengelompokkan datanya dilakukan secara partisi. Pada metode yang digunakan dalam algoritma *K-means*, data akan dikelompokkan menjadi beberapa bagian kelompok dan setiap kelompok memiliki ciri-ciri yang mirip dengan satu sama lain, namun memiliki ciri-ciri yang

berbeda dengan kelompok lain. Hal itu bertujuan untuk meminimalisir perbedaan antara satu *cluste*r dan memaksimalkan perbedaan data dengan *cluster* lain (Rizki Muliono dan Zulfikar Sembiring, 2019).

Berikut ini istilah dalam algoritma k-means:

1. *Cluster*, merupakan suatu kelompok atau grup.
2. *Centroid*, merupakan titik pusat untuk menentukan *Euclidean* distance.
3. Iterasi, merupakan pengulangan suatu proses yang akan berhenti ketika hasil iterasi telah konvergen.

Dalam menggunakan algoritma *K-Means*, terdapat langkah-langkah yang harus dilakukan diantaranya yaitu:

1. Menentukan banyak k-*cluster* yang ingin dibentuk.
2. Membangkitkan nilai random untuk pusat *cluster* awal (*centroid*) sebanyak k-*cluster*.
3. Menghitung jarak setiap data input terhadap masingmasing centroid menggunakan rumus jarak *Eucledian* (*Eucledian Distance*) hingga ditemukan jarak yang paling dekat dari setiap data dengan *centroid*. Berikut adalah persamaan *Eucledian Distance*:

   $$d(x_i, \mu_i) = \sqrt{(x_i - \mu_i)^2}$$

   dengan d (xi , μi) adalah jarak antara *cluster* x dengan pusat *cluster* μ pada kata ke i , xi adalah bobot kata ke i pada *cluster* yang ingin dicari jaraknya, μi bobot kata ke i pada pusat *cluster*.
4. Mengklasifikasikan setiap data berdasarkan kedekatannya dengan centroid (jarak terkecil).
5. Memperbarui nilai *centroid*. Nilai *centroid* baru diperoleh dari rata-rata *cluster* yang bersangkutan dengan menggunakan rumus:

$$C_k = \frac{1}{n_k} \sum d_i$$

nk = jumlah data dalam *cluster*

di = jumlah dari nilai jarak yang masuk dalam masing-masing *cluster*

6. Melakukan perulangan dari langkah 2 hingga 5 hingga anggota tiap *cluster* tidak ada yang berubah.
7. Jika langkah 6 telah terpenuhi, maka nilai rata-rata pusat cluster (μj) pada iterasi terakhir akan digunakan sebagai parameter untuk menentukan klasifikasi data.

2.3.2 *Agglomerative Hirarchial*

Hierarchical Clustering adalah metode analisis kelompok yang berusaha untuk membangun sebuah hierarki kelompok. Hierarchical clustering dibagi menjadi dua yaitu *Agglomeratif Clustering* dan *Difisive Clustering*. *Agglomeratif Clustering* mengelompokkan data dengan pendekatan bawah atas (*bottom up*), sedangkan *Difisif Clustering* menggunakan pendekatan atas bawah (*top-bottom*). *Agglomerative Hierarchical Clustering* diklasifikasikan kedalam dua kategori yaitu *bottom-up* dan *top-down*. Pada *bottom-up* setiap pengamatan dimulai dari *cluster*nya sendiri dan pasangan *cluster* digabung saat satu data mendekati hierarki. Dan pada kategori *top-down* semua pengamatan dimulai dalam satu *cluster* dan perpecahan dilakukan secara rekursif saat satu data memisahkan diri dari hierarki (Sasirekha & Baby, 2013).

Proses pengelompokan *Agglomerative Hierarchical Clustering*. Jika diberi satu set item N yang akan dikelompokkan dan matriks N * N (atau kesamaan) maka proses dasar dari pengelompokan Agglomerative Hierarchical dapat dilakukan dengan iteratif mengikuti keempat langkah berikut (Marinove & Boncheva (2008):

1. Mulailah dengan menugaskan setiap item ke *cluster*. Biarkan jarak (kesamaan) antara kelompok adalah sama sebagai jarak (kesamaan) antara item yang dikandungnya.

2. Temukan pasangan cluster terdekat (paling mirip) dan gabungkan keduanya menjadi satu kelompok.
3. Hitung jarak (kesamaan) antara cluster baru dan masing-masing cluster lama.
4. Ulangi langkah 2 dan 3 sampai semua item dikelompokkan menjadi satu kelompok dengan ukuran N.

2.3.3 DBSCAN

Density-Based Spatial Clustering Of Applications With Noise (DBSCAN) adalah pengelompokkan yang dilakukan berdasarkan wilayah padat dalam ruang data, dipisahkan oleh wilayah dengan kepadatan titik yang lebih rendah. Algoritma DBSCAN didasarkan pada gagasan intuitif tentang "cluster" dan "noise". Ide utamanya adalah bahwa untuk setiap titik dalam sebuah *cluster*, kelompok dengan radius tertentu harus memiliki setidaknya sejumlah titik minimum.

2.4 Objek Penelitian

Objek kajian dari penelitian ini adalah data "Klasifikasi Tingkat Kemiskinan di Indonesia". Bagian dari data ini yang dijadikan sebagai objek penelitian adalah faktor-faktor yang mempengaruhi tingkat kemiskinan tersebut antara lain pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun), persentase rumah tangga yang memiliki akses terhadap sanitasi layak, persentase Rumah tangga yang memiliki akses terhadap air Minum layak dan faktor lainnya yang terdapat pada dataset tersebut. Objek ini diambil karena klasifikasi pada dataset tersebut kurang akurat sehingga diperlukan pengelompokkan kembali untuk meningkatkan akurasi dan relevansi hasil analisis.

2.5 Metode Penelitian

Metode penelitian ini menggunakan studi literatur pada dataset klasifikasi tingkat kemiskinan di Indonesia dengan pendekatan yang memanfaatkan sumber-sumber tertulis yang relevan dan terpercaya untuk

mengumpulkan informasi, menganalisis data, dan menarik kesimpulan menarik tentang fenomena kemiskinan di Indonesia.

2.6 Pengumpulan Data

Pengumpulan data merupakan tahap penting dalam proses penelitian yang melibatkan pengambilan informasi yang diperlukan untuk menjawab pertanyaan penelitian dan mencapai tujuan penelitian.

2.6.1 Sumber Data

Sumber data adalah asal atau tempat dari mana data dikumpulkan. Dalam konteks penelitian, sumber data merujuk pada basis atau asal informasi yang digunakan untuk melakukan analisis dan mencapai tujuan penelitian. Dalam penelitian ini, data diperoleh dari Kaggle, sebuah platform terkemuka yang menyediakan akses ke berbagai dataset untuk analisis data dan pengembangan model machine learning. Kaggle dikenal luas di kalangan data scientist dan peneliti karena menyediakan dataset yang terverifikasi dan dapat diandalkan.

Dataset yang digunakan dalam penelitian ini berjudul "Klasifikasi Tingkat Kemiskinan di Indonesia". Dataset ini dipilih karena mengandung informasi yang relevan dan komprehensif mengenai berbagai aspek yang berkaitan dengan kemiskinan di Indonesia. Pemilihan dataset ini didasarkan pada pertimbangan bahwa data tersebut harus mencakup informasi yang cukup detail dan beragam untuk mendukung analisis yang mendalam. Data yang komprehensif ini memungkinkan peneliti untuk melakukan eksplorasi data, pembersihan data, dan pemodelan dengan cara yang lebih efektif.

2.6.2 Atribut

Atribut dalam data merujuk pada kolom atau fitur yang mewakili karakteristik atau variabel tertentu dari data yang sedang dianalisis. Setiap atribut memberikan informasi spesifik yang digunakan untuk memahami dan menganalisis fenomena yang sedang diteliti. Pada dataset "Klasifikasi Tingkat Kemiskinan di Indonesia", atribut-atribut ini mencakup berbagai aspek sosial, ekonomi, dan

demografis yang relevan dengan tingkat kemiskinan di berbagai wilayah Indonesia. Beberapa atribut yang digunakan dalam penelitian ini yang didapat dari dataset "Klasifikasi Tingkat Kemiskinan di Indonesia" yaitu:

1. Provinsi

   Nama provinsi di Indonesia tempat data dikumpulkan. Atribut ini membantu mengidentifikasi lokasi geografis dari data yang dianalisis.

2. Kabupaten/Kota

   Nama kabupaten atau kota di dalam provinsi yang menjadi unit analisis. Atribut ini memberikan detail lebih lanjut tentang lokasi spesifik dari data.

3. Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota

   Persentase penduduk di setiap kabupaten atau kota yang dikategorikan sebagai miskin. Atribut ini penting untuk mengukur tingkat kemiskinan di setiap wilayah.

4. Rata-rata Lama Sekolah Penduduk 15+ (Tahun)

   Rata-rata jumlah tahun pendidikan formal yang telah diselesaikan oleh penduduk berusia 15 tahun ke atas. Atribut ini mencerminkan tingkat pendidikan penduduk.

5. Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)

   Jumlah pengeluaran rata-rata per individu dalam satu tahun, yang disesuaikan dengan faktor-faktor ekonomi tertentu. Atribut ini memberikan gambaran tentang kesejahteraan ekonomi individu.

6. Indeks Pembangunan Manusia

   Indeks yang mengukur capaian pembangunan manusia berdasarkan kesehatan, pendidikan, dan standar hidup. Atribut ini memberikan gambaran keseluruhan tentang kualitas hidup di wilayah tertentu.

7. Umur Harapan Hidup (Tahun)

   Rata-rata jumlah tahun seseorang diharapkan hidup sejak lahir. Atribut ini mencerminkan kondisi kesehatan dan kesejahteraan masyarakat.

8. Persentase Rumah Tangga yang Memiliki Akses terhadap Sanitasi Layak

   Persentase rumah tangga yang memiliki akses ke fasilitas sanitasi yang memadai. Atribut ini penting untuk menilai kualitas lingkungan hidup dan kesehatan masyarakat.

9. Persentase Rumah Tangga yang Memiliki Akses terhadap Air Minum Layak

   Persentase rumah tangga yang dapat mengakses air minum yang bersih dan aman. Atribut ini juga penting untuk kesehatan dan kesejahteraan masyarakat.

10. Tingkat Pengangguran Terbuka

    Persentase angkatan kerja yang tidak bekerja tetapi aktif mencari pekerjaan. Atribut ini mencerminkan situasi pasar tenaga kerja di wilayah tersebut.

11. Tingkat Partisipasi Angkatan Kerja

    Persentase penduduk usia kerja yang aktif bekerja atau mencari pekerjaan. Atribut ini mengukur tingkat keterlibatan penduduk dalam pasar tenaga kerja.

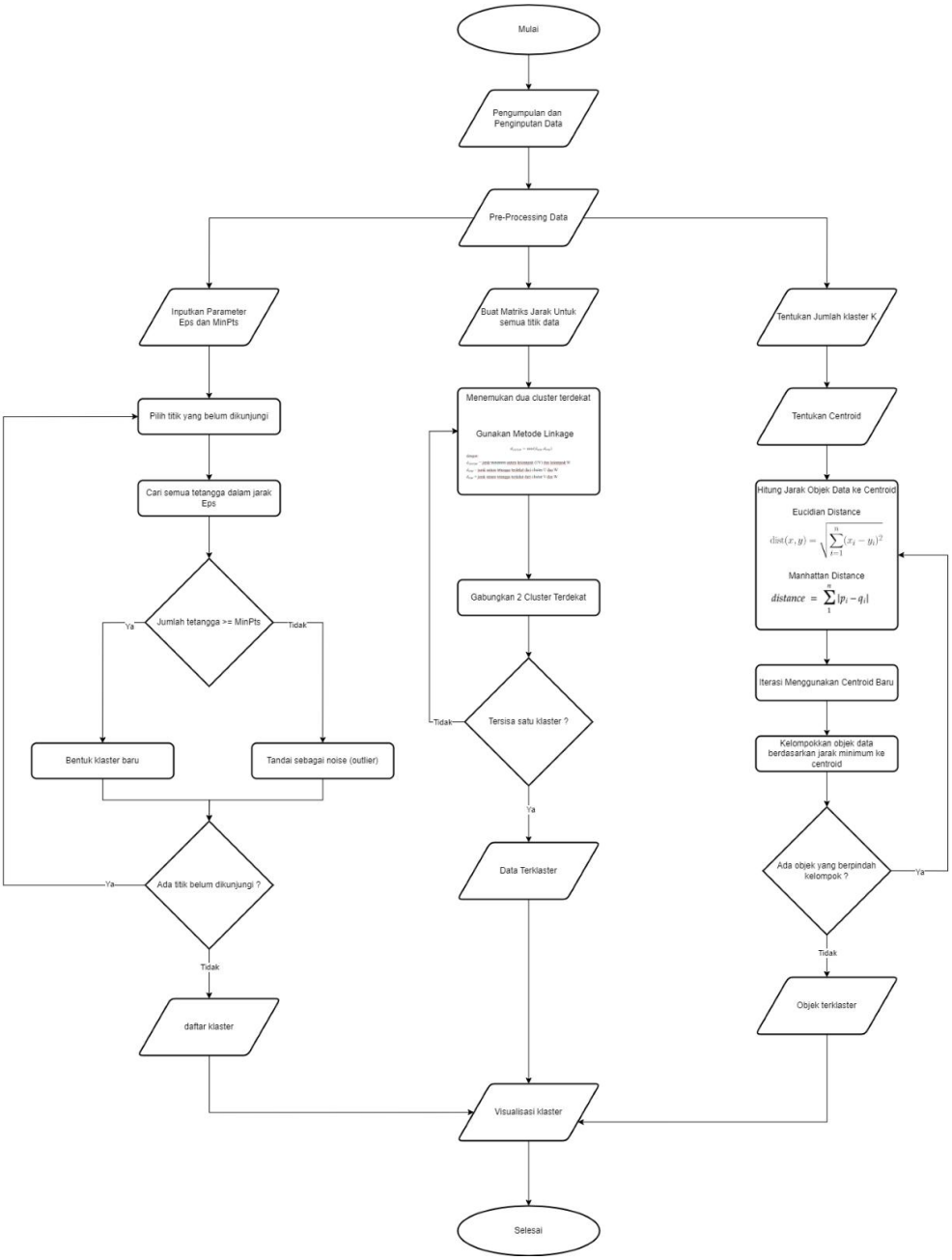
12. PDRB atas Dasar Harga Konstan Menurut Pengeluaran (Rupiah)

    Produk Domestik Regional Bruto (PDRB) yang dihitung berdasarkan harga konstan, menunjukkan nilai total produksi barang dan jasa di wilayah tersebut. Atribut ini digunakan untuk mengukur pertumbuhan ekonomi daerah.

Dengan memahami atribut-atribut ini, dapat dilakukan analisis yang lebih komprehensif untuk mengidentifikasi faktor-faktor yang mempengaruhi tingkat kemiskinan di Indonesia dan membangun algoritma klasterisasi yang dapat digunakan untuk memprediksi tingkat kemiskinan di berbagai wilayah di Indonesia.

2.4 Flowchart Penelitian

Berikut flowchart dari penilitian yang akan dilakukan sesuai pada Gambar 2.1



Gambar 2.1 Flowchart Penelitian

Penjelasan:

1. Mulai, merupakan tahap awal dari proses
2. Pengumpulan dan Pengunduhan Data Pada tahap ini, dataset "Klasifikasi Tingkat Kemiskinan di Indonesia" yang diperoleh dari Kaggle dikumpulkan dan diunduh untuk diproses

3. Pre-processing data untuk memeriksa apakah ada nilai yang hilang (missing value) atau tidak.
4. Pemrosesan data selanjutnya memiliki 3 algoritma seperti yang terlihat di flowchart ada 3 algoritma dari yang paling kiri yaitu DBSCAN, Agglomerative Hierarchical, dan K-Means
    a) DBSCAN
        i. Inisialisasi parameter, menginputkan parameter EPS dan MinPts
        ii. Memilih titik yang belum dikunjungi untuk dijadikan titik yang akan digunakan
        iii. Mencari titik lain yang berada dalam jarak EPS
        iv. Menghitung apakah jumlah titik yang ditemukan dalam jarak EPS >= Minpts, jika ya bentuk klaster baru namun jika tidak tandai sebagai noise(outlier)
        v. Mengecek apakah masih ada titik yang belum dikunjungi, jika ya kembali pada langkah ke-2 yaitu memilih titik belum dikunjungi, jika tidak maka iterasinya selesai dan dari situ didapatkan daftar klaster
    b) Agglomerative Hierarchical Clustering
        i. Membuat matriks jarak untuk semua titik data
        ii. Menemukan dua klaster terdekat, bisa menggunakan metode linkage
        iii. Menggabungkan 2 klaster terdekat
        iv. Memeriksa apakah tersisa 1 klaster, jika ya berarti data sudah terklaster, jika tidak kembali ke langkah kedua yaitu menemukan dua klaster terdekat
    c) K-Means
        i. Tentukan jumlah klaster K pada tahap ini, jumlah klaster yang ingin dibentuk (K) ditentukan terlebih dahulu sebelum proses klasterisasi dimulai.
        ii. Menentukan centroid setelah jumlah klaster ditentukan, centroid awal untuk setiap klaster harus ditentukan. Centroid

awal ini dapat dipilih secara acak atau menggunakan metode inisialisasi tertentu.

iii. Hitung Jarak Objek Data ke Centroid Untuk setiap objek data, jarak antara objek data tersebut dengan setiap centroid dihitung menggunakan metrik jarak tertentu. misalnya. Euclidean Distance: Jarak Euclidean antara objek data x dan centroid y dihitung dengan rumus: dist(x, y) = √(∑(x_i - y_i)^2). Manhattan Distance: Jarak Manhattan antara objek data x dan centroid y dihitung dengan rumus: distance = ∑|x_i - y_i|.

iv. Iterasi Menggunakan Centroid Baru Setelah jarak setiap objek data ke centroid dihitung, objek data akan dikelompokkan ke dalam klaster dengan centroid terdekat. Kemudian, centroid baru untuk setiap klaster dihitung kembali menggunakan nilai rata-rata dari anggota klaster tersebut. Proses ini diulang secara iteratif hingga centroid konvergen (tidak ada perubahan anggota klaster).

v. Kelompokkan objek data berdasarkan jarak minimum ke centroid Pada setiap iterasi, setiap objek data akan dikelompokkan ke dalam klaster dengan centroid terdekat berdasarkan jarak minimum yang dihitung sebelumnya.

vi. Ada objek yang berpindah kelompok? Jika masih ada objek data yang berpindah ke klaster lain setelah iterasi terakhir, maka proses iterasi menggunakan centroid baru dilanjutkan kembali. Namun, jika tidak ada objek data yang berpindah klaster, maka proses clustering dianggap konvergen dan selesai.

5. Visualisasi klaster sesuai dengan algoritma yang dilakukan penggunaan diagramnya juga berbedanamun tujuannya tetap sama yaitu membuat pelaporan dalam bentuk gambar yang mudah dipahami
6. Selesai, tahap akhir dari proses

BAB III
PERANCANGAN DAN IMPLEMENTASI

3.1 Perhitungan Manual

Pada dataset ini kami hanya mengambil beberapa sampel kolom dan baris saja untuk dilakukannya perhitungan seperti Tabel 3.1 dibawah

Tabel 3.1 Data Perhitungan Algoritma K-Means

Data	Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen) (A)	Indeks Pembangunan Manusia (B)	Umur Harapan Hidup (Tahun) (C)	Persentase rumah tangga yang memiliki akses terhadap air minum layak(D)
1	18.98	66.41	65.28	87.45
2	20.36	69.22	67.43	78.58
3	13.18	67.44	64.4	79.65
4	13.41	69.44	68.22	86.71
5	14.45	67.83	68.74	83.16
6	15.26	73.37	68.86	90.1
7	18.81	71.67	67.99	94.22
8	14.05	73.58	69.79	82.36
9	19.59	70.7	66.95	89.24
10	13.25	72.33	71.26	93.53

3.1.1 Perhitungan *K-Means*

Algoritma *KMeans* merupakan salah satu metode partisial clustering berbasis titik pusat (*centroid*) dimana proses *clustering* dilakukan dengan meminimalkan jarak jumlah kuadrat antara data dengan masing-masing pusat *cluster*. Algoritma *K-Means* dalam penerapannya memerlukan tiga parameter yang seluruhnya ditentukan pengguna yaitu jumlah *cluster* k, inisialisasi *cluster*, dan jarak.

Berikut langkah-langkah melakukan clustering dengan menggunakan algoritma k-means:

1. Menentukan nilai K sebanyak jumlah cluster atau kelompok yang diinginkan. Jumlah cluster yang diambil pada data ini adalah sebanyak 3 kelompok yang terdiri dari kelompok miskin, rentan miskin, dan tidak miskin.
   a) Miskin : Individu atau rumah tangga yang pendapatannya berada di bawah garis kemiskinan, sehingga mereka kesulitan memenuhi kebutuhan dasar seperti makanan, tempat tinggal, dan pakaian.
   b) Rentan miskin : Individu atau rumah tangga yang pendapatannya sedikit di atas garis kemiskinan, namun masih berisiko jatuh ke dalam kemiskinan jika terjadi guncangan ekonomi.
   c) Tidak miskin : Individu atau rumah tangga yang pendapatannya jauh di atas garis kemiskinan, memungkinkan mereka untuk memenuhi kebutuhan dasar serta memiliki cadangan keuangan untuk menghadapi situasi darurat dan meningkatkan kualitas hidup mereka.

   Pada perhitungan ini ditandai dengan angka yang akan menentukan individu/rumah tangga tersebut akan masuk ke cluster yang mana dengan rincian sebagai berikut:
   a) Angka 3 : Kategori miskin
   b) Angka 2 : Kategori rentan miskin
   c) Angka 1 : Kategoti tidak miskin
2. Setelah menentukan nilai K sebanyak jumlah cluster yang diinginkan, lalu pilih sebanyak K data dari set data sebagai pusat cluster (centroid) secara random. Disini data random yang diambil yaitu pada baris ke 1,5 dan 9 seperti Tabel 3.2 dibawah

Tabel 3.2 Data Centroid

Data	**Centroid**	**A**	**B**	**C**	**D**
1	1	18.98	66.41	65.28	87.45
5	2	14.45	67.83	68.74	83.16
9	3	19.59	70.7	66.95	89.24

3. Menghitung jarak antara objek dengan masing-masing *centroid* menggunakan rumus euclidean distance.

$$d(x_i, x_j) = \sqrt{\left(\left|x_{i1} - x_{j1}\right|^2 + \left|x_{i2} - x_{j2}\right|^2 + \cdots + \left|x_{ip} - x_{jp}\right|^2\right)}$$

Dimisalkan perhitungan untuk jarakn data 1 dengan data 1 sampai 5

$$D11 = \sqrt{(18.98 - 18.98)^2 + (66.41 - 66.41)^2 + (65.28 - 65.28)^2 + (87.45 - 87.45)^2} = 0$$

$$D12 = \sqrt{(20.36 - 18.98)^2 + (69.22 - 66.41)^2 + (67.43 - 65.28)^2 + (78.58 - 87.45)^2} = 9.6488$$

$$D13 = \sqrt{(13.18 - 18.98)^2 + (67.44 - 66.41)^2 + (64.4 - 65.28)^2 + (79.65 - 87.45)^2} = 9.8140$$

$$D14 = \sqrt{(13.41 - 18.98)^2 + (69.44 - 66.41)^2 + (68.22 - 65.28)^2 + (86.71 - 87.45)^2} = 7.0283$$

$$D15 = \sqrt{(14.45 - 18.98)^2 + (67.83 - 66.41)^2 + (68.74 - 65.28)^2 + (83.16 - 87.45)^2} = 7.2741$$

Perhitungan diatas diatas dilakukan pada *centroid* pertama, lanjutkan sampai data ke-10 begitupun dengan *centroid* 2 dan 3 sampai didapatkan hasil pada Tabel 3.3 dibawah ini.

Tabel 3.3 Hasil Perhitungan Euclidean Distance

Data	(A)	(B)	(C)	(D)	C1	C2	C3
1	18.98	66.41	65.28	87.45	0	7.203929	4.976866
2	20.36	69.22	67.43	78.58	9.648829	7.656409	10.80043
3	13.18	67.44	64.4	79.65	9.814036	5.811076	12.25505
4	13.41	69.44	68.22	86.71	7.0283	3.867105	6.913306
5	14.45	67.83	68.74	83.16	7.274132	0	8.65026
6	15.26	73.37	68.86	90.1	9.061948	7.059327	5.501409
7	18.81	71.67	67.99	94.22	8.992969	11.9539	5.23749
8	14.05	73.58	69.79	82.36	11.04364	1.703673	9.715246
9	19.59	70.7	66.95	89.24	4.976866	8.22132	0
10	13.25	72.33	71.26	93.53	11.85774	10.78551	8.934915

4. Mengelompokkan objek berdasarkan jarak terdekat dengan *centroid*. Setelah melakukan perhitungan untuk mencari jarak antar centroid, lalu dikelompokkan data tersebut berdsarkan jarak

terdekat dengan *centroid* nya dengan menggunakan nilai minimum antar tiga centroid tersebut seperti Tabel 3.4 dibawah

Tabel 3.4 Penentuan Cluster

C1	C2	C3	Min	Cluster
0	7.203929	4.976866	0	1
9.648829	7.656409	10.80043	7.656409	2
9.814036	5.811076	12.25505	5.811076	2
7.0283	3.867105	6.913306	3.867105	2
7.274132	0	8.65026	0	2
9.061948	7.059327	5.501409	5.501409	3
8.992969	11.9539	5.23749	5.23749	3
11.04364	1.703673	9.715246	1.703673	2
4.976866	8.22132	0	0	3
11.85774	10.78551	8.934915	8.934915	3

Pada Tabel 4.4 terlihat bahwa jika suatu data dekat dengan *centroid* pertama atau nilai minimum nya pada *centroid* pertama maka akan dikelompokkan menjadi *cluster* 1 begitu pun untuk data yang dekat pada *centroid* 2 dan 3 maka masing-masing nya akan masuk ke dalam *cluster* 2 dan *cluster* 3. Berikut hasil dari *cluster* pada Tabel 3.5 dibawah

Tabel 3.5 Pengelompokkan Cluster

Data	C1	C2	C3
1	1		
2		1	
3		1	
4		1	
5		1	
6			1
7			1
8		1	

9			1
10			1

5. Setelah mendapatkan kelompok pada iterasi pertama selanjutnya kita akan mencari pada iterasi kedua ataupun selanjutnya hingga tidak ada lagi objek yang berpindah *cluster*.
6. Pada iterasi kedua ini, tentukan terlebih dahulu centroid baru dengan menggunakan rumus sebagai berikut:

$$C_{m(q)} = \frac{1}{n_m} \sum_{i=1}^{n_m} x_{i(q)}$$

Cara melakukan perhitungan untuk menentukan centroid nya dengan rumus tersebut sebagai berikut:

a) Pada tabel 3.5 terlihat bahwa data yang masuk pada centroid 1 hanya satu buah pada baris pertama. Jadi data baris pertama pada tabel 4.1 akan dibagi dengan satu.
b) Pada tabel 3.5 untuk data yang masuk ke dalam centroid 2 sebanyak 5 buah terdapat pada data ke- 2,3,4,5, dan 8. Jadi untuk mencari centroid baru nya dengan membagi data ke-2,3,4,5, dan 8 yang pada tabel 4.1 dibagi dengan lima.
c) Pada tabel 3.5 untuk data yang masuk ke dalam centroid 3 sebanyak 4 buah terdapat pada data ke- 6,7,9,10. Jadi untuk mencari centroid baru nya dengan membagi data ke-6,7,9,10 yang pada tabel 4.1 dibagi dengan empat.
d) Setelah mencari perhitungan tersebut maka akan dapat hasil perhitungan yang ada pada Tabel 3.6 berikut\

Tabel 3.6 Hasil Perhitungan Centroid Baru

Penentuan cluster baru	**A**	**B**	**C**	**D**
Centroid baru ke-1	18.98	66.41	65.28	87.45
Centroid baru	15.09	69.502	67.716	82.092

ke-2				
Centroid baru ke-3	16.7275	72.0175	68.765	91.7725

7. Setelah mendapatkan *centroi*d baru, lakukan perhitungan kembali dengan rumus eucledian distance dan didapatkan hasil seperti Tabel 3.7 dibawah

Tabel 3.7 Hasil Perhitungan Iterasi 2

Data	**(A)**	**(B)**	**(C)**	**(D)**	**C1**	**C2**	**C3**
1	18.98	66.41	65.28	87.45	0	7.70291	8.206524
2	20.36	69.22	67.43	78.58	9.648829	6.345736	14.03016
3	13.18	67.44	64.4	79.65	9.814036	4.985897	14.12609
4	13.41	69.44	68.22	86.71	7.0283	4.940262	6.601155
5	14.45	67.83	68.74	83.16	7.274132	2.322581	9.843675
6	15.26	73.37	68.86	90.1	9.061948	8.968117	2.605589
7	18.81	71.67	67.99	94.22	8.992969	12.87253	3.32392
8	14.05	73.58	69.79	82.36	11.04364	4.699466	9.96274
9	19.59	70.7	66.95	89.24	4.976866	8.565388	4.431421
10	13.25	72.33	71.26	93.53	11.85774	12.4407	4.637294

8. Mengelompokkan objek berdasarkan jarak terdekat dengan *centroid* seperti Tabel 3.8 dibawah

Tabel 3.8 Penentuan Cluster Iterasi 2

C1	**C2**	**C3**	**Min**	**Cluster**	**Ket**
0	7.70291	8.206524	0	1	**tetap**
9.648829	6.345736	14.03016	6.345736	2	**tetap**
9.814036	4.985897	14.12609	4.985897	2	**tetap**
7.0283	4.940262	6.601155	4.940262	2	**tetap**
7.274132	2.322581	9.843675	2.322581	2	**tetap**
9.061948	8.968117	2.605589	2.605589	3	**tetap**
8.992969	12.87253	3.32392	3.32392	3	**tetap**

11.04364	4.699466	9.96274	4.699466	2	**tetap**
4.976866	8.565388	4.431421	4.431421	3	**tetap**
11.85774	12.4407	4.637294	4.637294	3	**tetap**

Pada Tabel 3.8 terlihat bahwa hasil *cluster* yang didapatkan sama dengan hasil *cluster* pada iterasi pertama. Jadi hasil tersebut berarti sudah benar untuk pengelompokkan nya yang terdapat pada Tabel 3.9 dibawah.

Tabel 3.9 Pengelompokkan Cluster Iiterasi 2

Data	C1	C2	C3
1	1		
2		1	
3		1	
4		1	
5		1	
6			1
7			1
8		1	
9			1
10			1

Kesimpulan:

Pada Tabel 3.5 dan Tabel 3.9 terlihat bahwa data ke-1 masuk ke dalam *cluster* 1 artinya tergolong tidak miskin, data ke-2,3,4,5, dan 8 masuk ke dalam *cluster* 2 artinya masuk ke kategori rentan miskin dan data ke- 6,7,9,10 masuk ke dalam *cluster* 3 artinya masuk ke kategori miskin.

3.1.2 Perhitungan *Agglomerative Hirarchial*

Agglomerative Clustering adalah salah satu teknik *clustering* hierarki yang populer dalam analisis data. Algoritma ini bekerja dengan cara menggabungkan pasangan-pasangan data terdekat ke dalam *cluster*,

kemudian menggabungkan *cluster-cluster* tersebut hingga semua data tergabung dalam satu *cluster* besar atau hingga kriteria tertentu terpenuhi.

Berikut langkah-langkah melakukan clustering dengan menggunakan algoritma *Agglomerative Hirarchial*:

1. Menghitung matriks jarak menggunakan rumus Euclidean

$$d_{ij} = \sqrt{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2}$$

rumus jarak *Euclidean*

keterangan.
d_{ij} : jarak antara objek *i* dengan *j*
x_{ij} : nilai objek *i* pada variabel ke-*k*
x_{jk} : nilai objek *j* pada variabel ke-*k*
p : banyaknya variabel yang diamati

Jadi, setiap data akan dihitung jarak tiap-tiap data dari data 1 ke data 2 hingga jarak data 9 ke data 10. Setiap pasangan data akan menjadi satu *cluster* sehingga pada tahapan pertama akan terbentuk 10 *cluster*. Sebagai contoh untuk cluster pertama yaitu jarak antara data 1 dan data 2 dengan nilai

$d_{1,2} = \sqrt{(20.36 - 18.98)^2 + (69.22 - 66.41)^2 + (67.43 - 65.28)^2 + (78.58 - 87.45)^2} = 9.6488$

Sehingga didapatkan matriks jaraknya seperti Tabel 3.10 dibawah

Tabel 3.10 Matrix Jarak

Data	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0									
2	9.6488	0								
3	9.8140	8.0651	0							
4	7.0283	10.7272	8.2758	0						
5	7.2741	7.7170	5.7377	4.0678	0					
6	9.0619	13.3412	12.9840	5.5470	8.9177	0				

7	8.9930	15.9163	16.5760	9.5177	12.5156	5.7640	0			
8	11.0436	8.8704	8.6517	6.2399	5.9131	7.8918	13.0463	0		
9	4.9769	10.8004	12.2551	6.9133	8.6503	5.5014	5.2375	9.7152	0	
10	11.8577	17.2741	16.2367	8.0082	11.6438	4.7588	6.5206	11.3636	8.9349	0

2. Menggabungkan dua *cluster* terdekat yaitu *cluster* 4 dengan 5 karena jaraknya yang terkecil yaitu 4.0678 dibandingakan dengan *cluster* yang lain
3. Memperbarui matriks jarak menggunakan teknik pengelompokan complete linkage

$$d_{(ab)c} = \max\{d_{a,c}; d_{b,c}\}$$

rumus *complete linkage*

Pembaruan tahap 1

Dengan menggunakan rumus *complete linkage* jarak antara *cluster* 4,5 dengan data 1 sampai 10 akan dihitung dan matriks jarak akan diperbarui. Sebagai contoh untuk jarak *cluster* 4,5 dengan data 1

d(4,5),1 = max{d4,1; d5,1} = max{7.0283; 7.2741} = 7.2741}

Maka didapatlah matriks pembaruan seperti Tabel 3.11

Tabel 3.11 Pembaruan Tahap 1

Data	**4,5**	**1**	**2**	**3**	**6**	**7**	**8**	**9**	**10**
4,5	0								
1	7.2741	0							
2	10.7272	9.6488	0						
3	8.2758	9.8140	8.0651	0					

6	8.9177	9.0619	13.3412	12.9840	0				
7	12.5156	8.9930	15.9163	16.5760	5.7640	0			
8	6.2399	11.0436	8.8704	8.6517	7.8918	13.0463	0		
9	8.6503	4.9769	10.8004	12.2551	5.5014	5.2375	9.7152	0	
10	11.6438	11.8577	17.2741	16.2367	4.7588	6.5206	11.3636	8.9349	0

Pembaruan tahap 2

Pembaruan tahap 2 dilakukan untuk *cluster* terdekat yaitu 6,10 sehingga didapatkanlah perhitungan matriks jarak seperti Tabel 3.12 dibawah

Tabel 3.12 Pembaruan Tahap 2

Data	**4,5**	**6,10**	**1**	**2**	**3**	**7**	**8**	**9**
4,5	0							
6,10	11.6438	0						
1	7.2741	11.8577	0					
2	10.7272	17.2741	9.6488	0				
3	8.2758	16.2367	9.8140	8.0651	0			
7	12.5156	6.5206	8.9930	15.9163	16.5760	0		
8	6.2399	11.3636	11.0436	8.8704	8.6517	13.0463	0	

9	8.6503	8.9349	4.9769	10.8004	12.2551	5.2375	9.7152	0

Pembaruan tahap 3

Pembaruan tahap 3 dilakukan untuk *cluster* terdekat yaitu 1,9 sehingga didapatkanlah perhitungan matriks jarak seperti Tabel 3.13 dibawah

Tabel 3.13 Pembaruan Tahap 3

Data	**4,5**	**6,10**	**1,9**	**2**	**3**	**7**	**8**
4,5	0						
6,10	11.6438	0					
1,9	8.6503	11.8577	0				
2	10.7272	17.2741	10.8004	0			
3	8.2758	16.2367	12.2551	8.0651	0		
7	12.5156	6.5206	8.9930	15.9163	16.5760	0	
8	6.2399	11.3636	11.0436	8.8704	8.6517	13.0463	0

Pembaruan tahap 4

Pembaruan tahap 4 dilakukan untuk *cluster* terdekat yaitu 4,5,8 sehingga didapatkanlah perhitungan matriks jarak seperti Tabel 3.14 dibawah

Tabel 3.14 Pembaruan Tahap 4

Data	**4,5,8**	**6,10**	**1,9**	**2**	**3**	**7**
4,5,8	0					
6,10	11.6438	0				
1,9	11.0436	11.8577	0			
2	10.7272	17.2741	10.8004	0		
3	8.6517	16.2367	12.2551	8.0651	0	
7	13.0463	6.5206	8.9930	15.9163	16.5760	0

Pembaruan tahap 5

Pembaruan tahap 5 dilakukan untuk *cluster* terdekat yaitu 6,10,7 sehingga didapatkanlah perhitungan matriks jarak seperti Tabel 3.15 dibawah

Tabel 3.15 Pembaruan Tahap 5

Data	4,5,8	6,10,7	1,9	2	3
4,5,8	0				
6,10,7	13.0463	0			
1,9	11.0436	11.8577	0		
2	10.7272	17.2741	10.8004	0	
3	8.6517	16.5760	12.2551	8.0651	0

Pembaruan tahap 6

Pembaruan tahap 6 dilakukan untuk *cluster* terdekat yaitu 2,3 sehingga didapatkanlah perhitungan matriks jarak seperti Tabel 3.16 dibawah

Tabel 3.16 Pembaruan Tahap 6

Data	4,5,8	6,10,7	1,9	2,3
4,5,8	0			
6,10,7	13.0463	0		
1,9	11.0436	11.8577	0	
2,3	10.7272	17.2741	12.2551	0

Pembaruan tahap 7

Pembaruan tahap 7 dilakukan untuk *cluster* terdekat yaitu 4,5,8,2,3 sehingga didapatkanlah perhitungan matriks jarak seperti Tabel 3.17 dibawah

Tabel 3.17 Pembaruan Tahap 7

Data	4,5,8,2,3	6,10,7	1,9
4,5,8,2,3	0		
6,10,7	17.2741	0	

1,9	12.2551	11.8577	0

Pembaruan tahap 8

Pembaruan tahap 8 dilakukan untuk *cluster* terdekat yaitu 6,10,7,1,9 sehingga didapatkanlah perhitungan matriks jarak seperti Tabel 3.18 dibawah

Tabel 3.18 Pembaruan Tahap 8

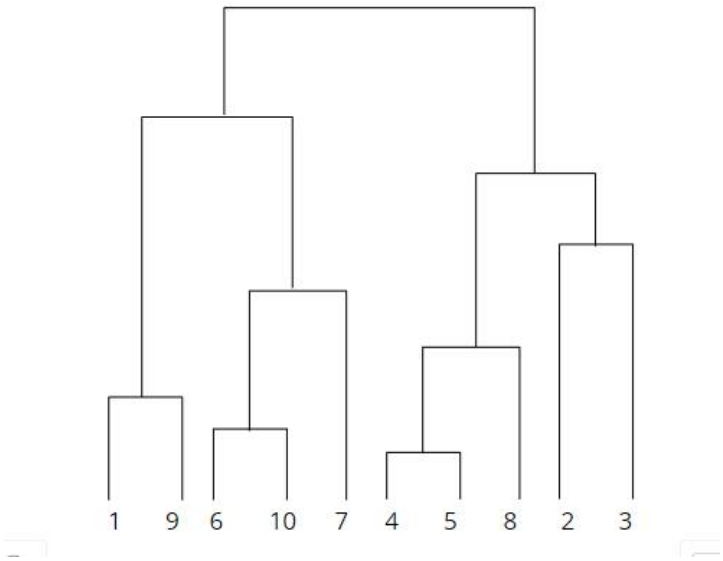
Data	**4,5,8,2,3**	**6,10,7,1,9**
4,5,8,2,3	0	
6,10,7,1,9	17.2741	0

4. Membuat list dari cluster yang terbentuk pada masing-masing tahap seperti Tabel 3.19 dibawah

Tabel 3.19 List Cluster Hirarki

Tahapan	**Jumlah Cluster**	**Anggota**	**Jarak Terdekat**
1	9	4,5	4.0678
2	8	6,10	4.7588
3	7	1,9	4.9769
4	6	4,5,8	6.2399
5	5	6,10,7	6.5206
6	4	2,3	8.0651
7	3	4,5,8,2,3	10.7272
8	2	6,10,7,1,9	11.8577
9	1	1,2,3,4,5,6,7,8,9,10	17.2741

5. Membuat dendogram dari *list cluster* yang telah dibuat seperti gambar di bawah ini



Gambar 3.1 Dendogram

6. Dari hasil perhitungan yang dilakukan ketika diambil clusternya 3 maka
   a) *Cluster* 1 berisi data 4,5,8,2,3
   b) *Cluster* 2 berisikan data 1,9
   c) *Cluster* 3 berisikan data 6,10,7

3.1.3 Perhitungan DBSCAN

Berikut langkah-langkah dalam melakukan perhitungan menggunakan algoritma DBSCAN

1. Pilih parameter epsilon (ϵ) dan MinPts:
   a) ϵ (radius maksimum untuk mencari titik tetangga).
   b) MinPts (jumlah minimum titik untuk membentuk sebuah cluster)

   Misalkan kita pilih ϵ=5 dan MinPts = 3.

   Kolom yang digunakan untuk membuat algoritma DBSCAN pada Tabel 3.20 dibawah

Tabel 3.20 Data Algoritma DBSCAN

Persentase penduduk miskin	**Rata rata lama sekolah Penduduk 15+**
18,98	9,48

20,36	8,68
13,18	8,88
13,41	9,67
14,45	8,21
36,26	2,16
28,81	4,94
41,66	0,39
40,59	3,25
11,39	11,57

2. Hitung jarak antar titik: Kita akan menghitung jarak Euclidean antara setiap pasangan titik. Rumus dasar untuk menghitung jarak euclidean sebagai berikut:

$$d_{ij} = \sqrt{\left[(x_i - x_j)^2 + (y_i - y_j)^2\right]}$$

x_i = koordinat x untuk fasilitas i

y_i = koordinat y untuk fasilitas i

d_{ij} = jarak antar fasilitas I dan j

Contoh perhitungan jarak antara titik pertama dan titik kedua :

$$\text{Jarak} = \sqrt{(20.36 - 18.98)^2 + (8.68 - 9.48)^2}$$
$$\text{Jarak} = \sqrt{(1.38)^2 + (-0.80)^2}$$
$$\text{Jarak} = \sqrt{1.9044 + 0.64}$$
$$\text{Jarak} = \sqrt{2.5444}$$
$$\text{Jarak} = 1.59$$

Lakukan perhitungan ini untuk semua pasangan titik dan buat matriks jaraknya sehingga didapatkan hasilnya seperti Tabel 3.21 dibawah

Tabel 3.21 Jarak Antar Titik

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0	1,59	6,26	1,91	2,45	24,38	15.25	27,59	22,27	3,27

2	1,59	0	7,29	3,11	1,99	21,73	12,09	25,02	19,63	4,79
3	6,26	7,29	0	0,86	0,72	23,75	15,97	27,95	23,68	3,51
4	1,91	3,11	0,86	0	1,69	23,65	15,08	27,99	22,69	2,19
5	2,45	1,99	0,72	1,69	0	22,79	13,51	26,96	21,52	3,47
6	24,38	21,73	23,75	23,65	22,79	0	9,23	5,78	6,11	27,32
7	15,25	12,09	15,97	15,08	13,51	9,23	0	13,56	8,36	19,95
8	27,59	25,02	27,95	27,99	26,96	5,78	13,56	0	2,87	31,37
9	22,27	19,63	23,68	22,69	21,52	6,11	8,36	2,87	0	26,42
10	3,27	4,79	3,51	2,19	3,47	27,32	19,95	31,37	26,42	0

3. Identifikasi titik inti, titik batas, dan *noise*
   a) Titik inti: Titik yang memiliki setidaknya MinPts dalam radius ϵ.
   b) Titik batas: Titik yang berada dalam radius ϵ dari titik inti, tetapi tidak memiliki cukup tetangga untuk menjadi titik inti.
   c) *Noise*: Titik yang tidak berada dalam radius ϵ dari titik inti manapun.

   Misalkan kita gunakan epsilon = 5 dan minPts = 3

   a) Titik 1: Jarak ke 4 titik dalam ϵ: 2 (1.59), 4 (1.91), 5 (2.45), 10 (3.27) - Titik inti
   b) Titik 2: Jarak ke 4 titik dalam ϵ: 1 (1.59), 4 (3.11), 5 (1.99), 10 (4.79) - Titik inti
   c) Titik 3: Jarak ke 3 titik dalam ϵ: 4 (0.86), 5 (0.72), 10 (3.51) - Titik inti
   d) Titik 4: Jarak ke 4 titik dalam ϵ: 1 (1.91), 2 (3.11), 3 (0.86), 5 (1.69) - Titik inti
   e) Titik 5: Jarak ke 4 titik dalam ϵ: 1 (2.45), 2 (1.99), 3 (0.72), 4 (1.69) - Titik inti
   f) Titik 6: Jarak ke 2 titik dalam ϵ: 7 (9.23), 9 (6.11) - Noise
   g) Titik 7: Jarak ke 2 titik dalam ϵ: 6 (9.23), 9 (8.36) - Noise
   h) Titik 8: Jarak ke 2 titik dalam ϵ: 6 (5.78), 9 (2.87) - Noise
   i) Titik 9: Jarak ke 2 titik dalam ϵ: 6 (6.11), 8 (2.87) - Noise

j) Titik 10: Jarak ke 4 titik dalam ϵ: 1 (3.27), 2 (4.79), 3 (3.51), 4 (2.19) - Titik inti

4. Hasil Clustering

Berdasarkan perhitungan di atas, kita dapat membentuk cluster sebagai berikut:

a) Cluster 1: Titik 1, Titik 2, Titik 3, Titik 4, Titik 5, Titik 10

b) Noise: Titik 6, Titik 7, Titik 8, Titik 9

3.2 Implementasi

Implementasi dilakukan dengan pengkodingan menggunakan bahasa python di google colab dengan tahapan:

3.2.1 Import Library

Melakukan import library yang akan digunakan selama implementasi seperti Gambar 3.2 di bawah

```
[2]  import pandas as pd
     import matplotlib.pyplot as plt
     from sklearn.preprocessing import StandardScaler
     from sklearn.cluster import DBSCAN
     from sklearn.datasets import make_circles
     import numpy as np
     from sklearn import metrics
     from sklearn.metrics import silhouette_score
     from sklearn.cluster import AgglomerativeClustering
     from sklearn.cluster import KMeans
```

Gambar 3.2 Import Libaray

3.2.2 Data Gathering

Menampilkan dataset yang digunakan yaitu dataset “Klasifikasi Tingkat Kemiskinan di Indonesia” menggunakan library pandas. Diketahui bahwa dataset terdiri dari 999 baris dan 13 kolom seperti Gambar 3.3 di bawah

```
df = pd.read_csv('Klasifikasi Tingkat Kemiskinan di Indonesia.csv', sep=";")
df
```

	Provinsi	Kab/Kota	Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)	Rata-rata Lama Sekolah Penduduk 15+ (Tahun)	Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)	Indeks Pembangunan Manusia	Umur Harapan Hidup (Tahun)	Persentase rumah tangga yang memiliki akses terhadap sanitasi layak	Persentase rumah tangga yang memiliki akses terhadap air minum layak	Tingkat Pengangguran Terbuka	Tingkat Partisipasi Angkatan Kerja	PDRB atas Dasar Harga Konstan menurut Pengeluaran (Rupiah)	Klasifikasi Kemiskinan
0	ACEH	Simeulue	18,98	9,48	7148.0	68,41	65,28	71,56	87,45	5,71	71,15	1648096.0	0.0
1	ACEH	Aceh Singkil	20,36	8,68	6776.0	69,22	67,43	69,56	78,68	8,36	62,85	1780419.0	1.0
2	ACEH	Aceh Selatan	13,18	8,88	6180.0	67,44	64,4	62,55	79,85	6,46	60,85	4345784.0	0.0
3	ACEH	Aceh Tenggara	13,41	9,67	8030.0	69,44	68,22	62,71	86,71	6,43	69,62	3487157.0	0.0
4	ACEH	Aceh Timur	14,45	8,21	6577.0	67,83	68,74	66,75	83,16	7,13	59,48	8433526.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
994	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
995	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
996	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
997	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
998	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

999 rows × 13 columns

Gambar 3.3 Data Gathering

3.2.3 Data Preparation

1. Menampilkan informasi dari setiap kolom yang ada, tipe data setiap kolom, dan informasi lainnya seperti Gambar 3.4 dibawah.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 0 to 998
Data columns (total 13 columns):
 #   Column                                                Non-Null Count  Dtype
---  ------                                                --------------  -----
 0   Provinsi                                              514 non-null    object
 1   Kab/Kota                                              514 non-null    object
 2   Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)  514 non-null  object
 3   Rata-rata Lama Sekolah Penduduk 15+ (Tahun)           514 non-null    object
 4   Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)  514 non-null  float64
 5   Indeks Pembangunan Manusia                            514 non-null    object
 6   Umur Harapan Hidup (Tahun)                            514 non-null    object
 7   Persentase rumah tangga yang memiliki akses terhadap sanitasi layak  514 non-null  object
 8   Persentase rumah tangga yang memiliki akses terhadap air minum layak  514 non-null  object
 9   Tingkat Pengangguran Terbuka                          514 non-null    object
 10  Tingkat Partisipasi Angkatan Kerja                    514 non-null    object
 11  PDRB atas Dasar Harga Konstan menurut Pengeluaran (Rupiah)  514 non-null  float64
 12  Klasifikasi Kemiskinan                                514 non-null    float64
dtypes: float64(3), object(10)
memory usage: 101.6+ KB
```

Gambar 3.4 Info Dataset

2. Mengecek nilai null yang ada pada dataset dari setiap kolom dan menjumlahkannya. Terdapat sekitar 485 baris dari setiap kolom yang artinya jumlah nilai null nya sama untuk seluruh kolom seperti Gambar 3.5 dibawah

```
df.isnull().sum()
```

```
Provinsi                                                485
Kab/Kota                                                485
Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)  485
Rata-rata Lama Sekolah Penduduk 15+ (Tahun)             485
Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)  485
Indeks Pembangunan Manusia                              485
Umur Harapan Hidup (Tahun)                              485
Persentase rumah tangga yang memiliki akses terhadap sanitasi layak  485
Persentase rumah tangga yang memiliki akses terhadap air minum layak  485
Tingkat Pengangguran Terbuka                            485
Tingkat Partisipasi Angkatan Kerja                      485
PDRB atas Dasar Harga Konstan menurut Pengeluaran (Rupiah)  485
Klasifikasi Kemiskinan                                  485
dtype: int64
```

Gambar 3.5 Jumlah Data NULL

3. Mengitung jumlah nilai null dari setiap kolom dan menambahkannya sehingga didapatlah total nilai null yang ada pada dataset, serta menghapusnya. Jika berhasil maka akan ada pesan berapa banyaknya jumlah nilai null yang telah dihapus tersebut seperti Gambar 3.6 dibawah

```
# Menghitung jumlah nilai null dalam DataFrame
jumlah_null = df.isnull().sum().sum()

if jumlah_null > 0:
    # Menghapus baris yang mengandung nilai null dari DataFrame
    df.dropna(inplace=True)
    print(f"{jumlah_null} nilai null berhasil dihapus dari DataFrame.")
else:
    print("Tidak ada nilai null yang ditemukan dalam DataFrame.")
```

```
6305 nilai null berhasil dihapus dari DataFrame.
```

Gambar 3.6 Menghapus Data NULL

4. Menghapus kolom yang tidak digunakan pada dataset, kolom yang tidak memfasilitasi untuk pembuatan model data mining, diantaranya kolom Provinsi, Kab/Kota, dan Klasifikasi Kemiskinan, serta menampilkan kolom yang tersisa yang akan digunakan untuk analisis lebih lanjut seperti Gambar 3.7 dibawah

```
# Menghapus kolom yang tidak digunakan
kolom_yang_tidak_digunakan = ['Provinsi','Kab/Kota','Klasifikasi Kemiskinan']
df2 = df.drop(columns=kolom_yang_tidak_digunakan)
df2
```

	Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)	Rata-rata Lama Sekolah Penduduk 15+ (Tahun)	Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)	Indeks Pembangunan Manusia	Umur Harapan Hidup (Tahun)	Persentase rumah tangga yang memiliki akses terhadap sanitasi layak	Persentase rumah tangga yang memiliki akses terhadap air minum layak	Tingkat Pengangguran Terbuka	Tingkat Partisipasi Angkatan Kerja	PDRB atas Dasar Harga Konstan menurut Pengeluaran (Rupiah)
0	18,98	9,48	7148.0	66,41	65,28	71,56	87,45	5,71	71,15	1648096.0
1	20,36	8,68	8776.0	69,22	67,43	69,56	78,58	8,36	62,85	1780419.0
2	13,18	8,88	8180.0	67,44	64,4	62,55	79,65	6,46	60,85	4345784.0
3	13,41	9,67	8030.0	69,44	68,22	62,71	86,71	6,43	69,62	3487157.0
4	14,45	8,21	8577.0	67,83	68,74	66,75	83,16	7,13	59,48	8433526.0
...	...	...	...	...	...	...	...	...	...	...
509	36,26	2,16	5412.0	43,17	65,86	11,43	85,03	0,94	89,43	831070.0
510	28,81	4,94	5415.0	55	65,85	12,11	71,24	5,68	78,20	906904.0
511	41,66	3,09	5328.0	48,34	65,69	0,36	35,01	1,43	75,75	767101.0
512	40,59	3,25	4673.0	49,96	65,36	0,00	85,23	0,79	85,01	841296.0
513	11,39	11,57	14937.0	80,11	70,52	85,31	97,10	11,67	63,75	22852202.0

514 rows × 10 columns

Gambar 3.7 Drop Kolom

3.2.4 Data Wrangling

Memastikan bahwa data dalam kolom-kolom tersebut (yang mungkin semula berupa string dengan pemisah ribuan menggunakan koma) diubah menjadi tipe data float64, sehingga data dapat

dioperasikan matematis secara benar. Misalnya, data angka persentase, rata-rata lama sekolah, indeks pembangunan manusia, umur harapan hidup, dll., akan lebih mudah diolah dan dianalisis setelah diubah menjadi tipe data numerik seperti Gambar 3.8 dibawah

```
df2['Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)'] = df2['Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)'].str.repla
df2['Rata-rata Lama Sekolah Penduduk 15+ (Tahun)'] = df2['Rata-rata Lama Sekolah Penduduk 15+ (Tahun)'].str.replace(',', '.').astype('float64')
df2['Indeks Pembangunan Manusia'] = df2['Indeks Pembangunan Manusia'].str.replace(',', '.').astype('float64')
df2['Umur Harapan Hidup (Tahun)'] = df2['Umur Harapan Hidup (Tahun)'].str.replace(',', '.').astype('float64')
df2['Persentase rumah tangga yang memiliki akses terhadap sanitasi layak'] = df2['Persentase rumah tangga yang memiliki akses terhadap sanitasi layak'].s
df2['Persentase rumah tangga yang memiliki akses terhadap air minum layak'] = df2['Persentase rumah tangga yang memiliki akses terhadap air minum layak']
df2['Tingkat Pengangguran Terbuka'] = df2['Tingkat Pengangguran Terbuka'].str.replace(',', '.').astype('float64')
df2['Tingkat Partisipasi Angkatan Kerja'] = df2['Tingkat Partisipasi Angkatan Kerja'].str.replace(',', '.').astype('float64')

df2
```

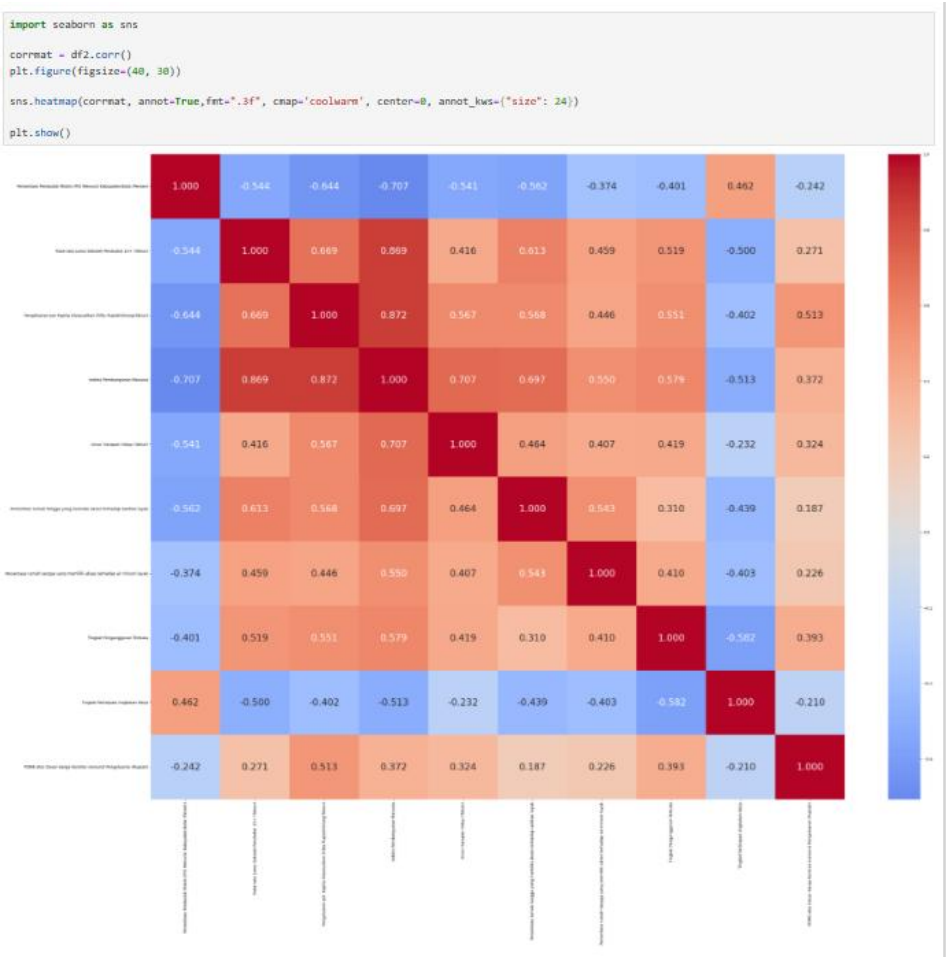
	Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)	Rata-rata Lama Sekolah Penduduk 15+ (Tahun)	Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)	Indeks Pembangunan Manusia	Umur Harapan Hidup (Tahun)	Persentase rumah tangga yang memiliki akses terhadap sanitasi layak	Persentase rumah tangga yang memiliki akses terhadap air minum layak	Tingkat Pengangguran Terbuka	Tingkat Partisipasi Angkatan Kerja	PDRB atas Dasar Harga Konstan menurut Pengeluaran (Rupiah)
0	18.98	9.48	7148.0	66.41	65.28	71.56	87.45	5.71	71.15	1648096.0
1	20.36	8.68	8776.0	69.22	67.43	69.56	78.58	8.36	62.85	1780419.0
2	13.18	8.88	8180.0	67.44	64.40	62.55	79.65	6.46	60.85	4345784.0
3	13.41	9.67	8030.0	69.44	68.22	62.71	86.71	6.43	69.62	3487157.0
4	14.45	8.21	8577.0	67.83	68.74	66.75	83.16	7.13	59.48	8433526.0
...	...	...	...	...	...	...	...	...	...	...
509	36.26	2.16	5412.0	43.17	65.86	11.43	85.03	0.94	89.43	831070.0
510	28.81	4.94	5415.0	55.00	65.85	12.11	71.24	5.68	78.20	906904.0
511	41.66	3.09	5328.0	48.34	65.69	0.36	35.01	1.43	75.75	767101.0
512	40.59	3.25	4673.0	49.96	65.36	0.00	85.23	0.79	85.01	841296.0
513	11.39	11.57	14937.0	80.11	70.52	85.31	97.10	11.67	63.75	22852202.0

514 rows × 10 columns

Gambar 3.8 Tipe Data

3.2.5 Data Analysis

Menampilkan heatmap untuk melihat hubungan antar kolom dalam dataset dengan parameternya dalam bentuk warna semakin pekat maka semakin kuat hubungan ataupun korelasinya seperti Gambar 3.9 dibawah



Gambar 3.9 Heatmap

3.2.6 Modeling

1. K-Means

Membuat sebuah variabel untuk menyimpan kolom yang akan digunakan untuk pembuatan model, disini variabel X1 akan digunakan untuk menyimpan semua kolom yang ada pada dataset setelah diolah. Selajutnya melakukan standardScaler untuk menyamakan skala dari nilai data agar memudahkan untuk analisis seperti Gambar 3.10 dibawah.

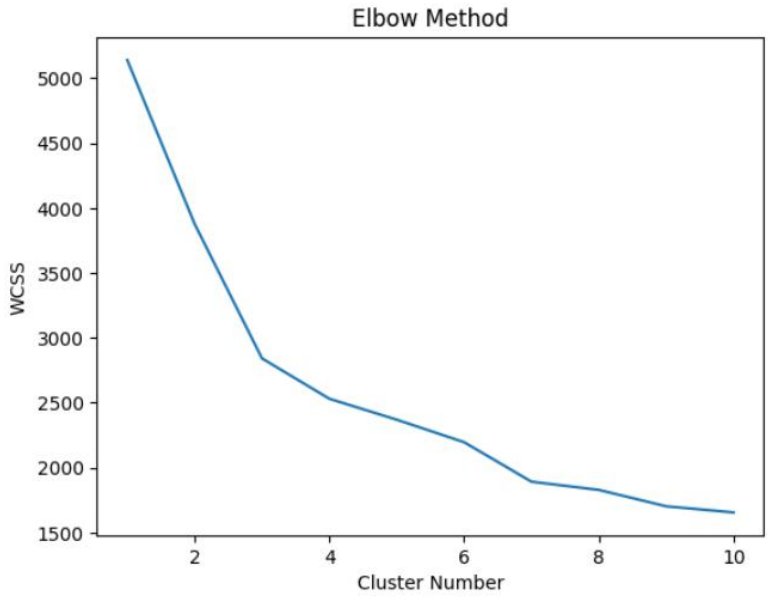
```
X1 = df2.iloc[:,:]
```

```
from sklearn.preprocessing import StandardScaler

X = StandardScaler().fit_transform(X1)
```

Gambar 3.10 Standarisasi 1

Membuat Elbow Method untuk melihat jumlah klaster yang optimal untuk mengelompokkan data, diliat berdasarkan kelengkungan garis pada gambar dibawah yang lebih membentuk sudut siku-siku yaitu pada nilai 3 dan akan kita tetapkan menjadi 3 klaster untuk pengelompokkan datanya seperti Gambar 3.11 dibawah

```
from sklearn.cluster import KMeans

wcss = []
for i in range(1, 11):
    model_kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state=42)
    model_kmeans.fit(X)
    wcss.append(model_kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Cluster Number')
plt.ylabel('WCSS')
plt.show()
```



Gambar 3.11 Elbow Method

Menerapkan algoritma *K-Means* untuk membagi data X menjadi 3 kluster berdasarkan karakteristiknya. Parameter init='k-means++' digunakan untuk inisialisasi *centroid* yang lebih baik, dan random_state=42 memastikan hasil yang konsisten pada setiap run. Hasil klustering disimpan dalam variabel y_kmeans, yang berisi

label kluster untuk setiap data point dalam X seperti Gambar 3.12 dibawah

```
model_kmeans = KMeans(n_clusters = 3, init = 'k-means++', random_state=42)
y_kmeans = model_kmeans.fit_predict(X)
```

Gambar 3.12 Model K-Means

Membuat kolom baru yang bernama pada dataset dengan nama Cluster K-Means untuk menyimpan hasil dari prediksi k-means dan serta menampilkan hasilnya seperti Gambar 3.13 dibawah

```
df2['Cluster K-Means'] = y_kmeans
df2
```

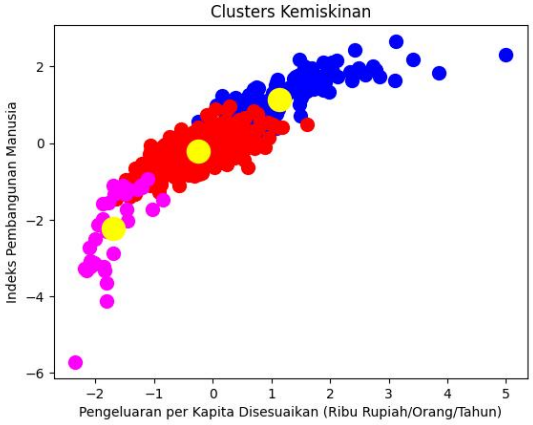
	Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)	Rata-rata Lama Sekolah Penduduk 15+ (Tahun)	Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)	Indeks Pembangunan Manusia	Umur Harapan Hidup (Tahun)	Persentase rumah tangga yang memiliki akses terhadap sanitasi layak	Persentase rumah tangga yang memiliki akses terhadap air minum layak	Tingkat Pengangguran Terbuka	Tingkat Partisipasi Angkatan Kerja	PDRB atas Dasar Harga Konstan menurut Pengeluaran (Rupiah)	Cluster K-Means
0	18.98	9.48	7148.0	66.41	65.28	71.56	87.45	5.71	71.15	1648096.0	1
1	20.36	8.68	8776.0	69.22	67.43	69.56	78.58	8.36	62.85	1780419.0	1
2	13.18	8.88	8180.0	67.44	64.40	62.55	79.65	6.46	60.85	4345784.0	1
3	13.41	9.67	8030.0	69.44	68.22	62.71	86.71	6.43	69.62	3487157.0	1
4	14.45	8.21	8577.0	67.83	68.74	66.75	83.16	7.13	59.48	8433526.0	1
...	...	...	...	...	...	...	...	...	...	...	...
509	36.26	2.16	5412.0	43.17	65.86	11.43	85.03	0.94	89.43	831070.0	2
510	28.81	4.94	5415.0	55.00	65.85	12.11	71.24	5.68	78.20	906904.0	2
511	41.66	3.09	5328.0	48.34	65.69	0.36	35.01	1.43	75.75	767101.0	2
512	40.59	3.25	4673.0	49.96	65.36	0.00	85.23	0.79	85.01	841296.0	2
513	11.39	11.57	14937.0	80.11	70.52	85.31	97.10	11.67	63.75	22852202.0	0

514 rows × 11 columns

Gambar 3.13 Hasil K-Means

Membuat scatter plot untuk melihat sebaran data yang kita kelompokkan menjadi beberapa klaster, memberi warna untuk klaster yang berbeda seperti Gambar 3.14 dibawah.

```
plt.scatter(X[y_kmeans == 0, 2], X[y_kmeans == 0, 3], s=100, c='blue', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 2], X[y_kmeans == 1, 3], s=100, c='red', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 2], X[y_kmeans == 2, 3], s=100, c='magenta', label = 'Cluster 3')
plt.scatter(model_kmeans.cluster_centers_[:,2], model_kmeans.cluster_centers_[:,3], s=300, c='yellow', label = 'Centroids')

plt.title('Clusters Kemiskinan')
plt.xlabel('Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)')
plt.ylabel('Indeks Pembangunan Manusia')
plt.show()
```



Gambar 3.14 Scatter K-Means

Mencetak banyaknya jumlah dari masing masing klaster yang telah dibentuk sebelumnya dengan menggunakan metode *K-Means* seperti Gambar 3.15 dibawah

```
df2['Cluster K-Means'].value_counts()
```

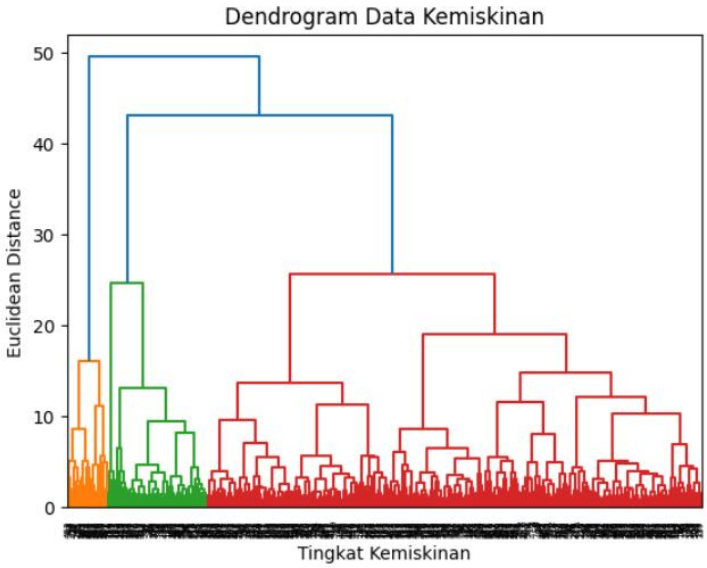
```
Cluster K-Means
1    349
0    130
2     35
Name: count, dtype: int64
```

Gambar 3.15 Jumlah Cluster K-Means

2. *Agglomerative Hierarchical Clustering*

Memvisualisasikan bagaimana data di *cluster* secara hierarkis dengan menggunakan Dendogram, dengan menggambarkan bagaimana kluster digabungkan satu sama lain sepanjang proses clustering. Ini membantu dalam memahami struktur dan hubungan antar data dalam dataset. Gambaran dendogram terdapada pada Gambar 3.16 dibawah

```
import scipy.cluster.hierarchy as sch

dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title('Dendrogram Data Kemiskinan')
plt.xlabel('Tingkat Kemiskinan')
plt.ylabel('Euclidean Distance')
plt.show()
```



Gambar 3.16 Dendogram Agglomerative

Membuat sebuah variabel untuk menyimpan kolom yang akan digunakan untuk pembuatan model, disini variabel X5 akan digunakan untuk menyimpan semua kolom yang ada pada dataset setelah diolah. Selajutnya melakukan standardScaler untuk menyamakan skala dari nilai data agar memudahkan untuk analisis dan disimpan dalam variabel X6 seperti Gambar 3.17 dibawah

```
X5 = df2.iloc[:,:]
```

```
X6 = StandardScaler().fit_transform(X5)
```

Gambar 3.17 Standarisasi 2

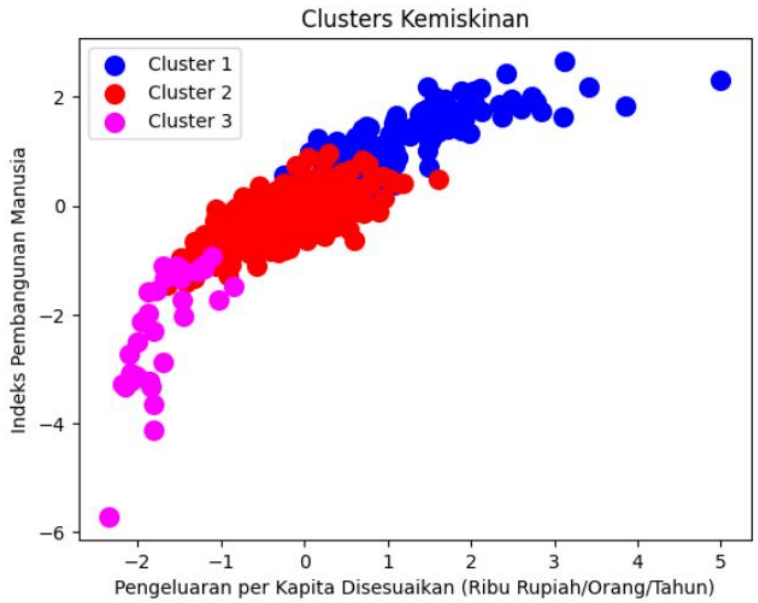
Menggunakan algoritma *Agglomerative Clustering* untuk mengelompokkan data dalam X6 menjadi 3 klaster, dengan menggunakan metrik jarak Euclidean dan metode penggabungan ward. Hasilnya, label klaster untuk setiap sampel disimpan dalam variabel y_ag seperti Gambar 3.18 dibawah

```
from sklearn.cluster import AgglomerativeClustering

ag = AgglomerativeClustering(n_clusters=3, metric='euclidean', linkage='ward')
y_ag = ag.fit_predict(X6)
```

Gambar 3.18 Model Agglomerative

Memvisualisasikan hasil clustering pada data X6, di mana setiap klaster direpresentasikan dengan scatter plot dan setiap klaster diberi warna yang berbeda seperti Gambar 3.19 dibawah

```
if isinstance(X6, pd.DataFrame):
    X6 = X6.values
plt.scatter(X6[y_ag == 0, 2], X6[y_ag == 0, 3], s=100, c='blue', label='Cluster 1')
plt.scatter(X6[y_ag == 1, 2], X6[y_ag == 1, 3], s=100, c='red', label='Cluster 2')
plt.scatter(X6[y_ag == 2, 2], X6[y_ag == 2, 3], s=100, c='magenta', label='Cluster 3')

plt.title('Clusters Kemiskinan')
plt.xlabel('Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)')
plt.ylabel('Indeks Pembangunan Manusia')
plt.legend()
plt.show()
```



Gambar 3.19 Scatter Agglomerative

Menambahkan kolom 'Cluster Agglomerative' ini ke DataFrame seperti Gambar 3.20, kita dapat dengan mudah melihat hasil clustering yang telah dilakukan dalam konteks data lain yang mungkin ada dalam DataFrame. Hal ini membantu dalam analisis dan pemahaman lebih lanjut tentang pola-pola dalam data yang muncul setelah dilakukan proses clustering.

```
df2['Cluster Agglomerative'] = y_ag
df2
```

	Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)	Rata-rata Lama Sekolah Penduduk 15+ (Tahun)	Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)	Indeks Pembangunan Manusia	Umur Harapan Hidup (Tahun)	Persentase rumah tangga yang memiliki akses terhadap sanitasi layak	Persentase rumah tangga yang memiliki akses terhadap air minum layak	Tingkat Pengangguran Terbuka	Tingkat Partisipasi Angkatan Kerja	PDRB atas Dasar Harga Konstan menurut Pengeluaran (Rupiah)	Cluster K-Means	Cluster Agglomerative
0	18.98	9.48	7148.0	66.41	65.28	71.56	87.45	5.71	71.15	1648096.0	2	2
1	20.36	8.68	8776.0	69.22	67.43	69.56	78.58	8.36	62.85	1780419.0	2	2
2	13.18	8.88	8180.0	67.44	64.40	62.55	79.65	6.46	60.85	4345784.0	2	2
3	13.41	9.67	8030.0	69.44	68.22	62.71	86.71	6.43	69.62	3487157.0	2	2
4	14.45	8.21	8577.0	67.83	68.74	66.75	83.16	7.13	59.48	8433526.0	2	2
...	...	...	...	...	...	...	...	...	...	...	...	...
509	36.26	2.16	5412.0	43.17	65.86	11.43	85.03	0.94	89.43	831070.0	0	1
510	28.81	4.94	5415.0	55.00	65.85	12.11	71.24	5.68	78.20	906904.0	0	1
511	41.66	3.09	5328.0	48.34	65.69	0.36	35.01	1.43	75.75	767101.0	0	1
512	40.59	3.25	4673.0	49.96	65.36	0.00	85.23	0.79	85.01	841296.0	0	1
513	11.39	11.57	14937.0	80.11	70.52	85.31	97.10	11.67	63.75	22852202.0	1	0

514 rows × 12 columns

Gambar 3.20 Hasil Agglomerative

Mencetak banyaknya jumlah dari masing masing klaster yang telah dibentuk sebelumnya dengan menggunakan metode A*gglomerative* seperti Gambar 3.21 dibawah

```
df2['Cluster Agglomerative'].value_counts()
```

```
Cluster Agglomerative
1    349
0    130
2     35
Name: count, dtype: int64
```

Gambar 3.21 Jumlah Cluster Agglomerative

3. DBSCAN

Membuat sebuah variabel untuk menyimpan kolom yang akan digunakan untuk pembuatan model, disini variabel X2 akan digunakan untuk menyimpan semua kolom yang ada pada dataset setelah diolah seperti Gambar 3.22 dibawah

```
X2 = df2.iloc[:,:]
X2
```

	Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)	Rata-rata Lama Sekolah Penduduk 15+ (Tahun)	Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)	Indeks Pembangunan Manusia	Umur Harapan Hidup (Tahun)	Persentase rumah tangga yang memiliki akses terhadap sanitasi layak	Persentase rumah tangga yang memiliki akses terhadap air minum layak	Tingkat Pengangguran Terbuka	Tingkat Partisipasi Angkatan Kerja	PDRB atas Dasar Harga Konstan menurut Pengeluaran (Rupiah)	Cluster K-Means	Clust Agglomerati
0	18.98	9.48	7148.0	66.41	65.28	71.56	87.45	5.71	71.15	1648096.0	1	
1	20.36	8.68	8776.0	69.22	67.43	69.56	78.58	8.36	62.85	1780419.0	1	
2	13.18	8.88	8180.0	67.44	64.40	62.55	79.65	6.46	60.85	4345784.0	1	
3	13.41	9.67	8030.0	69.44	68.22	62.71	86.71	6.43	69.62	3487157.0	1	
4	14.45	8.21	8577.0	67.83	68.74	66.75	83.16	7.13	59.48	8433526.0	1	
...	...	...	...	...	...	...	...	...	...	...	...	
509	36.26	2.16	5412.0	43.17	65.86	11.43	85.03	0.94	89.43	831070.0	2	
510	28.81	4.94	5415.0	55.00	65.85	12.11	71.24	5.68	78.20	906904.0	2	
511	41.66	3.09	5328.0	48.34	65.69	0.36	35.01	1.43	75.75	767101.0	2	
512	40.59	3.25	4673.0	49.96	65.36	0.00	85.23	0.79	85.01	841296.0	2	
513	11.39	11.57	14937.0	80.11	70.52	85.31	97.10	11.67	63.75	22852202.0	0	

514 rows × 12 columns

Gambar 3.22 Data DBSCAN

Kode ini menghasilkan dataset berbentuk lingkaran dengan 514 sampel, menggunakan faktor skala 0.3 dan level kebisingan sebesar 0.1. Selanjutnya, dataset tersebut dinormalisasi menggunakan StandardScaler. Kemudian, algoritma DBSCAN diterapkan dengan parameter eps=0.3 dan min_samples=5 untuk melakukan klastering pada dataset yang telah dinormalisasi, dan hasil prediksi disimpan dalam variabel y_pred. Hal ini dapat dilihat pada Gambar 3.23 dibawah

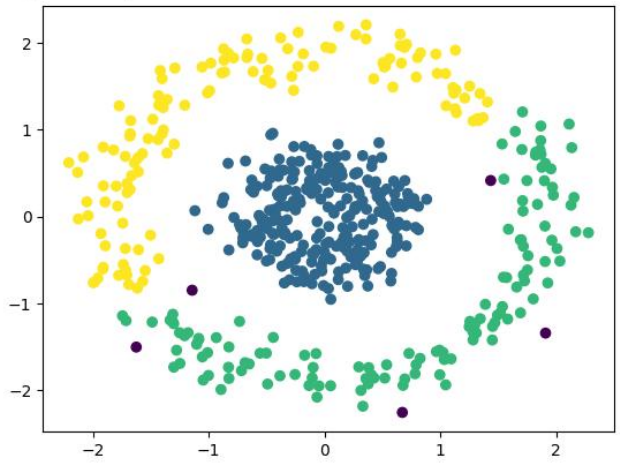
```
X2, Z = make_circles(n_samples = 514, factor = 0.3, noise = 0.1)

X2 = StandardScaler().fit_transform(X2)

y_pred = DBSCAN(eps=0.3, min_samples = 5). fit_predict(X2)
```

Gambar 3.23 Model DBSCAN

Membuat visualisasi scatter plot dari dataset yang telah dinormalisasi (X2), diwarnai berdasarkan hasil prediksi klaster dari DBSCAN (y_pred). Selanjutnya, mencetak jumlah klaster yang dihasilkan dan menghitung serta mencetak skor homogenitas dan kelengkapan dari hasil klaster (y_pred) terhadap label sejati (Z) menggunakan metrik homogenitas dan kelengkapan. Tampilan scatter plot seperti Gambar 3.24 dibawah

```
plt.scatter(X2[:,0], X2[:,1], c = y_pred)
print("Jumlah cluster: {}".format(len(set(y_pred[np.where(y_pred != -1)]))))
print("Homogeneity: {}", format(metrics.homogeneity_score(Z, y_pred)))
print("Completeness: {}", format(metrics.completeness_score(Z, y_pred)))
```

```
Jumlah cluster: 3
Homogeneity: {} 1.0000000000000002
Completeness: {} 0.6412576611810155
```



Gambar 3.24 Scatter DBSCAN

Memasukkan hasil dari cluster ke dataset awal seperti Gambar 3.25 dibawah

```
df2['Cluster DBSCAN'] = y_pred
df2
```

	Persentase Penduduk Miskin (P0) Menurut Kabupaten/Kota (Persen)	Rata-rata Lama Sekolah Penduduk 15+ (Tahun)	Pengeluaran per Kapita Disesuaikan (Ribu Rupiah/Orang/Tahun)	Indeks Pembangunan Manusia	Umur Harapan Hidup (Tahun)	Persentase rumah tangga yang memiliki akses terhadap sanitasi layak	Persentase rumah tangga yang memiliki akses terhadap air minum layak	Tingkat Pengangguran Terbuka	Tingkat Partisipasi Angkatan Kerja	PDRB atas Dasar Harga Konstan menurut Pengeluaran (Rupiah)	Cluster K-Means	Clust Agglomerati
0	18.98	9.48	7148.0	66.41	65.28	71.56	87.45	5.71	71.15	1648096.0	1	
1	20.36	8.68	8776.0	69.22	67.43	69.56	78.58	8.36	62.85	1780419.0	1	
2	13.18	8.88	8180.0	67.44	64.40	62.55	79.65	6.46	60.85	4345784.0	1	
3	13.41	9.67	8030.0	69.44	68.22	62.71	86.71	6.43	69.62	3487157.0	1	
4	14.45	8.21	8577.0	67.83	68.74	66.75	83.16	7.13	59.48	8433526.0	1	
...	...	...	...	...	...	...	...	...	...	...	...	
509	36.26	2.16	5412.0	43.17	65.86	11.43	85.03	0.94	89.43	831070.0	2	
510	28.81	4.94	5415.0	55.00	65.85	12.11	71.24	5.68	78.20	906904.0	2	
511	41.66	3.09	5328.0	48.34	65.69	0.36	35.01	1.43	75.75	767101.0	2	
512	40.59	3.25	4673.0	49.96	65.36	0.00	85.23	0.79	85.01	841296.0	2	
513	11.39	11.57	14937.0	80.11	70.52	85.31	97.10	11.67	63.75	22852202.0	0	

514 rows × 13 columns

Gambar 3.25 Hasil DBSCAN

3.3 Hasil Penelitian

Dengan menampilkan skor *silhoutte* untuk masing masing metode yang digunakan untuk melihat performa dari model data mining yang telah dibuat dengan menggunakan metode *K-Mean*s, *Agglomerative,* dan juga DBSCAN. Dari hasil tersebut algoritma *Agglomerative* memiliki hasil *silhouette* tertinggi yaitu sebesar 0.37 yang rinciannya terdapat pada Gambar 3.26 dibawah

```
from sklearn.metrics import silhouette_score
```

```
silhouette_avg = silhouette_score(X, y_kmeans)
print(f'Silhouette Score K-Means: {silhouette_avg}')
```

```
Silhouette Score K-Means: 0.30550888254216685
```

```
silhouette_avg = silhouette_score(X6, y_ag)
print(f'Silhouette Score Agglomerative: {silhouette_avg}')
```

```
Silhouette Score Agglomerative: 0.3700109544775474
```

```
silhouette_avg = silhouette_score(X2, Z)
print(f'Silhouette Score DBSCAN: {silhouette_avg}')
```

```
Silhouette Score DBSCAN: 0.19974871141603262
```

Gambar 3.26 Hasil Silhouette

BAB IV
PENUTUP

4.1 Kesimpulan

Penelitian ini menunjukkan bahwasannya terdapat perbedaan yang cukup signifikan dibandingkan dari hasil klasifikasi yang ada pada dataset sebelumnya. Penggunaan algoritma *Agglomerative Hirarchial* memiliki tingkat kecocokan yang tinggi dibandingkan dengan algoritma lainnya. Kecocokan ini disebabkan pada algoritma *Agglomerative* yang melakukan penhklasteran dengan membuat hirarki dari data-data yang ada. Sehingga data tingkat kemiskinan diklasterkan berdasarkan hirarki dari tingkat kemiskinannya

Dengan penggunaan data mining dalam mengklasterisasi tingkat kemiskinan di Indonesia diharapkan dapat membantu dalam pengelompokkan tingkat kemiskinan kabupaten dan kota di setiap provinsi di Indonesia, serta membantu agar penyaluran bantuan sosial menjadi lebih tepat sasaran. Melalui penelitian ini, diharapkan dapat memberikan sumbangsih yang bermanfaat bagi pemerintah, akademisi, dan masyarakat luas dalam upaya bersama untuk mengatasi masalah kemiskinan di Indonesia.

4.2 Saran

Penelitian ini masih menggunakan data secara umum, disarankan kedepannya dapat melakukan pengelompokkan tingkat kemiskinan menggunakan data yang lebih spesifik di wilayah kabupaten/kota.

DAFTAR PUSTAKA

Parwa, I. G. N. J. L.A., Yasa, I. G. W. M. 2019. *Pengaruh Pendidikan dan Investasi Terhadap Pertumbuhan Ekonomi Dan Kemiskinan Di Provinsi Bali*. E-Jurnal EP Unud, 8(5), 945-973.

Purnomo, S. D. 2021. *Analysis of Labor Absorption in Central Java Province. Ekonomis*: Journal of Economics and Business, 5(1), 240-244.

Hardinandar, F. (2019). *Determinan Kemiskinan (Studi Kasus 29 Kota/Kabupaten Di Provinsi Papua)*. Jurnal REP (Riset Ekonomi Pembangunan), 4(1), 1–12. https://doi.org/10.31002/rep.v4i1.1337

Oktaviana, D., Primandhana, W. P., & Wahed, M. (2021). *Analisis Pengaruh Pertumbuhan Ekonomi, Upah Minimum Kabupaten, dan Pengangguran terhadap Kemiskinan di Kabupaten Madiun.* Jurnal Syntax Idea, 4(1), 6.

Khomarudin, A. N. (2016). Teknik Data Mining : *Algoritma K-Means Clustering*. 1–12.

Warih, Eggy Inaidi Andana; Rahayu, Y. (2014). *Penerapan Data Mining untuk Menentukan Estimasi Produktivitas Tanaman Tebu dengan Menggunakan Algoritma Linear Regresi Berganda di Kabupaten Rembang*. Informatika, 1– 5.

R. Muliono and Z. Sembiring, *Data Mining Clustering Menggunakan Algoritma K-Means Untuk Klasterisasi Tingkat Tridarma Pengajaran Dosen.* CESS (Journal Comput. Eng. Syst. Sci., vol. 4, no. 2, pp. 2502–714, 2019.

Andayani, S. 2007. P*embentukan Cluster dalam Knowledge Discovery in Database dengan Algoritma KMeans*. Seminar Nasional Matematika dan Pendidikan Matematika 2007. Universitas Negeri Yogyakarta. Yogyakarta

Sasirekha, K., & Baby, P. (2013). *Agglomerative hierarchical clustering algorithm-a.*International Journal of Scientific and Research Publications, 83, 83.

Marinova–Boncheva, V. (2008). *Using the agglomerative method of hierarchical clustering as a data mining tool in capital market*.